Control Engineering Practice 18 (2010) 433–447

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Minimization of Wet End disturbances during web breaks using online LAV estimation

Amin Nobakhti [a,*], Hong Wang [b]

[a] Sharif University of Technology, Electrical Engineering, Iran
[b] The University of Manchester, Control Systems Centre, UK

ARTICLE INFO

Article history:
Received 5 December 2008
Accepted 9 January 2010
Available online 10 February 2010

Keywords:
Estimation algorithms
Paper industry
Linear estimation
Genetic algorithms

ABSTRACT

During a Wet End break, the loss of paper feed through the paper machine causes loss of sensory information and the remaining parts of the process are operated in open-loop. This causes the stock composition in the Headbox to deviate substantially from the nominal specifications, causing paper quality (after start up) and paper machine runability issues. In this work, the Base Sheet Ash measurement of the scanner is estimated using a least absolute value (LAV) model which can then be used for control of the chalk valve during the breaks to keep the Headbox Ash within specified limits. The model is computed using a very fast optimization algorithm which is able to compute the LAV solution using only basic elementary operations. The proposed approach has been developed for a UK paper mill.

1. Introduction to the paper making process

Paper making combines energy demanding complex systems and it is vitally important to the UK economy. These processes use large amount of resources (pulp, water) and demand huge amounts of energy. Even a small scale paper mill can use in excess of 150 GW h of energy, and 100 ktons of water per year. A reduction in energy usage of just 1% in such a mill is equivalent to the total energy used by 1200 households in one year. Part of this huge energy and water demand are intrinsic to the paper making process. In order to make paper, several stages of processing are applied to solid wood to convert it into a $<1\%$ consistency slurry which is free-running enough to make a web of material. This web then needs to be pressed and dried back to $<1\%$ moisture. This cycle is what makes paper making inherently energy and water consuming. Furthermore there are also procedural factors contributing to the inefficient operation. Unlike other manufacturing processes, paper making is subject to frequent production interruptions. When these interruptions cause a large disturbance to the machine, it results in marked increase of waste or unsalable paper production. One of the key challenges in paper making is therefore to minimize the effects of these disturbances on the machine and to minimize the machine downtime. An important and frequent disturbance which affects the paper making process is the phenomena of 'paper breaks'. When paper tears or breaks in the machine, production has to be stopped until the problem is rectified. In a typical paper machine with well trained personnel, a machine can be ready to resume paper making in under an hour (if there is a mechanical issue this may grow indefinitely longer). This papers deals with the effects of these disturbances on the paper machine and outlines the results of a study which was carried out at a UK paper mill.

A paper machine is generally divided into three main sections, 'Stock Preparation', the 'Wet End' and the 'Dry End'. An average sized paper machine is around 200 m long and 10 m wide. Paper can be produced at a rate of over 20 tons/h and at speeds of around 850 m/min. The major elements of the stock preparation are shown in Fig. 1. 'Stock' is the term used to refer to the slurry which is dried to make paper. The stock preparation area comprises of four 'lines'. The first line is the 'Virgin Fiber' line containing fresh fibers. The Virgin Fiber line starts by pulping; the process of separating the individual fibers in the Virgin pulp using water. Subsequently the fibers are further modified to help them bond. This mechanical process is known as Refinement. Different pulp types are usually added in varying proportions to form a 'Furnish' to make a particular grade of paper. In the virgin fiber line, stock is gradually reduced in consistency from almost solid, to 7.5%, then to 5.0% and finally the pulp which enters the refiners are diluted to 4.3% consistency.

In addition to the Virgin Fiber line, there are also three recycle lines. A recycle line can carry either recycled paper or material which did not make it on the web. Since fibers in these lines have already been refined, they do not need to be refined again.

[*] Corresponding author at: Sharif University of Technology, Electrical Engineering, Iran.
E-mail address: nobakhti@iee.org (A. Nobakhti).

0967-0661/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2010.01.003

Nomenclature

n_p	population size
$\mathbf{x}$	a vector of real valued parameters
$\mathbf{x}^U$	a vector upper bounding $\mathbf{x}$
$\mathbf{x}^L$	a vector lower bounding $\mathbf{x}$
$f(\mathbf{x})$	objective function value of $\mathbf{x}$
$\lfloor x \rfloor$	floor of x
$\lceil x \rceil$	ceiling of x
$\rho \mid_a^b$	a uniformly distributed random number between a and b
$\boldsymbol{\rho}^n \mid_a^b$	an n dimensional vector of uniformly distributed random numbers between a and b
$\rho \vdash_a^b$	a uniformly distributed random integer between a and b

Depending on where in the machine the material has come from, they will arrive in one of three lines. Any material which is collected on or around the 'Wire' section will be collected into the 'Couch' line. These will include edge spray cutter trims, solids drained off the Wire, and machine broke at the Wet End. Material collected from these sections can be used back in the process as they are. The second recycle feed line is the 'Recovered Stock' line. Recovered Stock refers to the fibers which are recovered from the water drained on the Wire by using 'Disk filters'. The final line is the 'Coated Broke' line. 'Broke' refers to base or coated paper which cannot be sold and is recycled. The paper may not be saleable due to poor quality parameters, or if a paper break has occurred blocking the path of paper in the machine.

The material from the Virgin Fiber, Coated Broke, Recovered Stock and Couch line enter the 'Blend Pipe' where mixing occurs, and arrive at the 'Thick Stock' blend chest. Chemicals, water and fillers are then added to form the Thick Stock, a 5% consistency slurry which is subsequently thinned down during several stages of cleaning and dilution to form the Thin Stock, a $<1\%$ consistency mix. Chemicals which aid the formation of the paper, improved filler/fiber bonding, improved flocculation and improved retention of fillers and fibers are then mixed into the Thin Stock. The Thin Stock then enters the Wet End (Fig. 2) through the 'Headbox' where it is sprayed onto the paper forming mesh called the 'Wire'. The purpose of the Wire section is to retain the solids while the water is drained away. Stock leaves the Headbox at a consistency of about 0.65% and it leaves the Wire at a consistency of about 30%. The rate of water drainage on the Wire can reach as high as 12,000 l/min. The next stage is the press section where the paper is flattened, consolidated and further

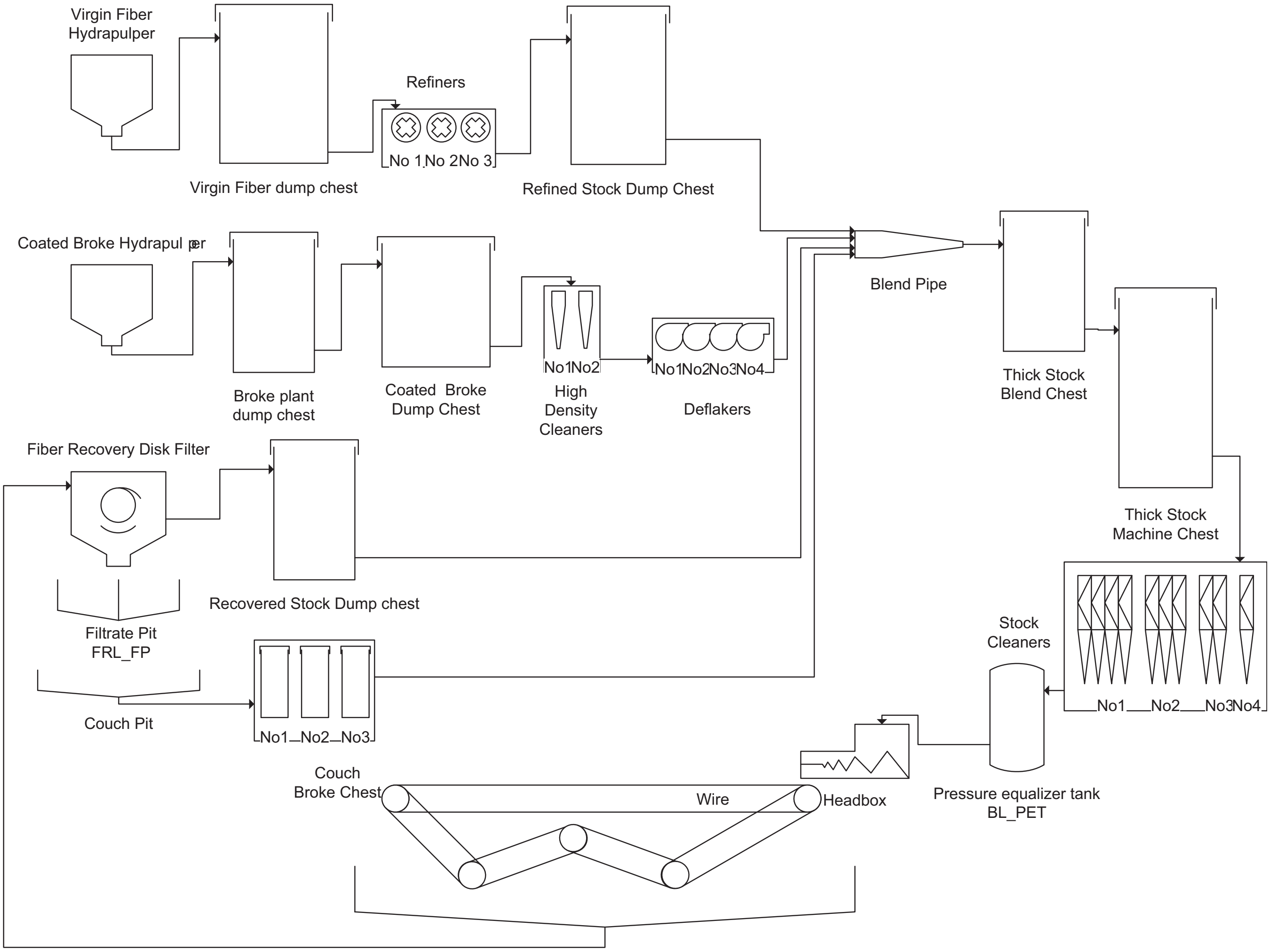

Fig. 1. Main elements of the Stock preparation system.

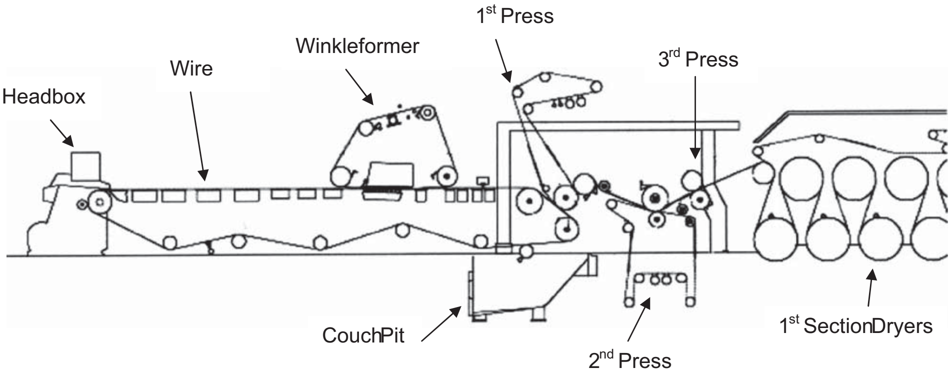

Fig. 2. The Wet End of the paper machine.

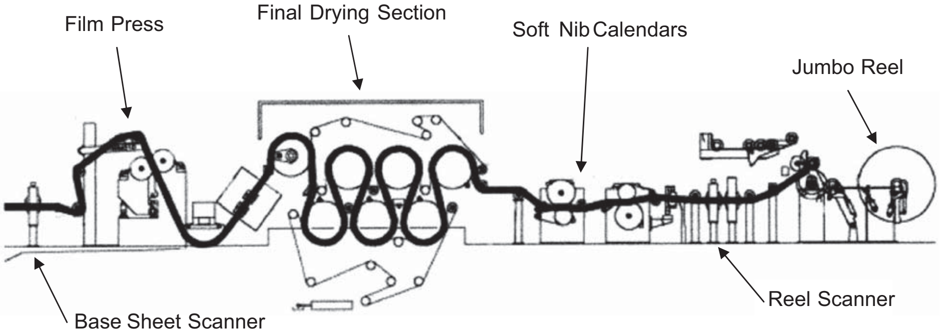

Fig. 3. The Dry End of the paper machine.

dried. Following the press section is the first drying stage where the paper is dried using steam heated rollers. At the end of the Wet End, the product is a paper web commonly referred to as the 'Base Sheet'. The Base Sheet then enters the Dry End of the machine (Fig. 3). The main functions of the Dry End are to add the coating to the Base Sheet, and a final drying section. This is then succeeded by the calendering stage, where irregularities in the sheet formation are corrected and its surface smoothness and gloss is improved. This final paper is referred to as the Reel End paper.

1.1. The role of Ash in paper making

Fillers are minerals which are added to the Thick Stock to fill the gap which is formed between wood fibers. Using fillers instead of fibers help make paper making economical since fillers are substantially cheaper than wood. They also aid in improving paper's printing properties (Smook, 1992). Nevertheless adding a high proportion of filler leads to paper discoloration, brittleness and poor opacity. Similarly, the paper machine runability will be adversely affected: The interlocking between wood fibers helps to build 'Tensile Strength' in the paper web (for more information on mechanical properties of the paper please see Brodeur, Gerhardstein, Lafond, Jong, & Habeger, 1999). Filler are particle shaped and do not bond well to either themselves or wood fibers. A high amount of filler will lead to reduced paper Tensile Strength in both the machine direction (MD) and cross direction (CD) leading to an increased occurrence of breaks.

Flocculation or aggregation mechanisms are required because fillers are too small to be retained mechanically on the Wire. Ideally one wants to limit the flocculation between the fibers (homoflocculation) and maximize the flocculation of fines and additives on the fibers (heteroflocculation). Early retention aid systems were single-component, most often based on acryl-amide chemistry, alum, starch, polyamines or polyethylene-imines (PEI). A further development of these systems was a dual-component system, which is usually based on interactions between a long-chain charged polyelectrolyte and a second polymer with the opposite charge. Two popular chemicals used for these purposes are Percol and Hydrocol (Bentonite). Percol is a cationic polymer added first and flocculates the anionic fibers and fillers. After the shear stage where the flocs are redispersed, the anionic polymer is added and forms new flocs by bridging between the cationic floc fragments. Hydrocol is an anionic nano or microparticle added to reflocculate the furnish components after the fan pump. The chemical reactions are almost instantaneous which can lead to large disturbances should the flow of chemicals be perturbed.

1.2. Control of Ash during breaks

At any one time there is a 600 m long web of paper of paper through the machine which can tear for a variety of reasons such as poor paper formation, low tensile strength, incorrect cylinder loads, incorrect chemical additions, or poor web position control (Daly, 1965; Wang, Logghe, & Miskin, 2005; Claveau, Chevrel, & Knittel, 2008; Ekvall & Hagglund, 2008). Although there have been some attempts to predict the occurrence of paper breaks (see Bonissone & Goebel, 2002, and references therein), these are generally unreliable. A well run paper machine will experience two or three breaks a day. The average recovery time from a break is 1 h. This down time is extremely costly, therefore it is desirable to not have such breaks, and if they occur, to recover as quickly as possible. The cost of breaks have been estimated to be around 2–7% of the total production loss. For a medium sized modern machine with a production of 150 ktons per annum, this translates to around $1 m (Ahola, 2005). Along the length of the machine, there are several diversion points where the web can be fed away from the machine. This allows the sections of the machine which are located before the break point to keep operating. Keeping the machine ticking over allows paper making to resume faster than from a cold start which means it is ultimately more economical. However, running the machine during a break poses difficult control challenges, one of which is the subject of this work.

There are two points of measurement which are vital for the control of the paper machine; the Base Sheet Scanner and the Reel End Scanner (See Fig. 3). These are large scanners which sweep across the paper web in a zig-zag fashion and measure several paper properties such as Sheet Weight, Sheet Ash (filler), floc size, floc intensity, caliper, etc (Wang, Dumont, & Davies, 1993). These measurements are then fed through to the plants DCS systems to adjust various chemical and product additions. The Base Sheet is largely used for control of the paper machine (Baki, Wang, Soylemez, & Munro, 2001). The Reel End scanner is primarily used to control the paper coating process (Walter, 1993). During a paper break, depending on where the break has occurred one or both of these will no longer be providing reliable information. For example, when a paper break occurs in the Dry End of the paper machine, paper will still pass through the Base Sheet Scanner, but not the Reel End scanner. Common practice is to ‘freeze’ all scanner measurements to their last value measured before the break occurred. This will stop the DCS system shutting down the plant.

One of the key measurements obtained from the Base sheet Scanner is the Base Sheet Ash, which is the percentage of fillers in the paper. Fig. 4 shows a typical Wet End Break. Notice that when the break occurs, the Base Sheet Ash reading freezes, which forces the Chalk Flow to also freeze. The recycle streams also bring back fillers to the blend pipe. The lowest ratio ($<5\%$) is found in the Recovered Stock line (due to the operation of the Disk Filter) and the highest ratio (around 40%) is in the Coated Broke line. During the break the fillers which were supposed to end up in the paper, find their way back to the Stock Preparation system. As the break prolongs, this build up of filler increases. At the same time, the controller cannot respond to this because of the frozen Base Sheet Ash signal. The effect is that the Ash content in the Machine Chest and consequently Headbox will climb (e.g. Fig. 4). The most prominent consequence is that the increased filler levels will lead to poor formation and low tensile strength when the paper is formed possibly causing a ‘train’ of successive breaks.

2. Estimation of base sheet ash

The proposal to the concerned mill to deal with this problem was to use data driven modeling and develop a soft-sensor which can be used for control of the Chalk Valve during the breaks. This is a novel approach in the paper industry. A major obstacle is that it is notoriously difficult to obtain ‘good’ data or models in this industry. Some of the most pertinent challenges are the wide range of products being made (over a 100 Basis Weight/Grade of finish/coating combinations), complex Wet End physical and chemical interactions which are not modelable, the frequency at which the parts, components, or systems on the machine are upgraded or changed, and finally the fact that despite advances in sensing technology, several important paper properties still cannot be measured online and require manual lab testing.

By far the largest obstacle in the way of successful modeling and estimation are operational disturbances. Measurement noise, interferences, and signal transients are typical disturbances which corrupt real time measurements. There are also ‘bad data’ associated with known faults. The primary source of such disturbances are poor operational compliance by the technical staff. For example during the routine maintenance of duplex-filters the operator does not prime the system correctly which will temporarily seize the flow, causing possible large disturbances (Fig. 5). There are also none-operator induced disturbances such as those encountered during scanner startup and warmup

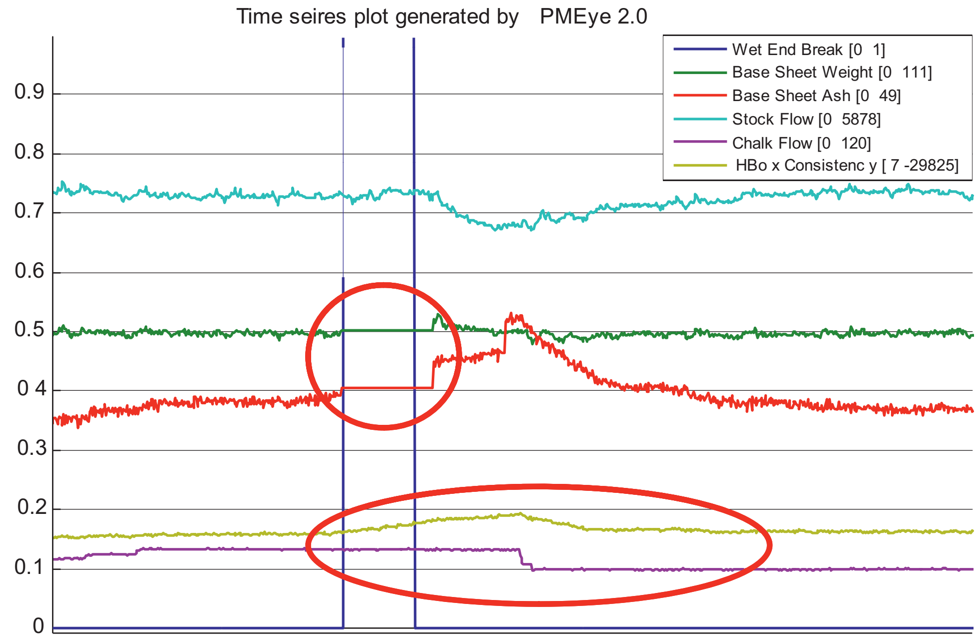

Fig. 4. The Wet End Break is shown by the thick blue line. When the break occurs, the Base Sheet Ash reading freezes. The DCS system is blind to the rising value of the Headbox Ash which is reflected in the bump in the Headbox consistency value after the break.

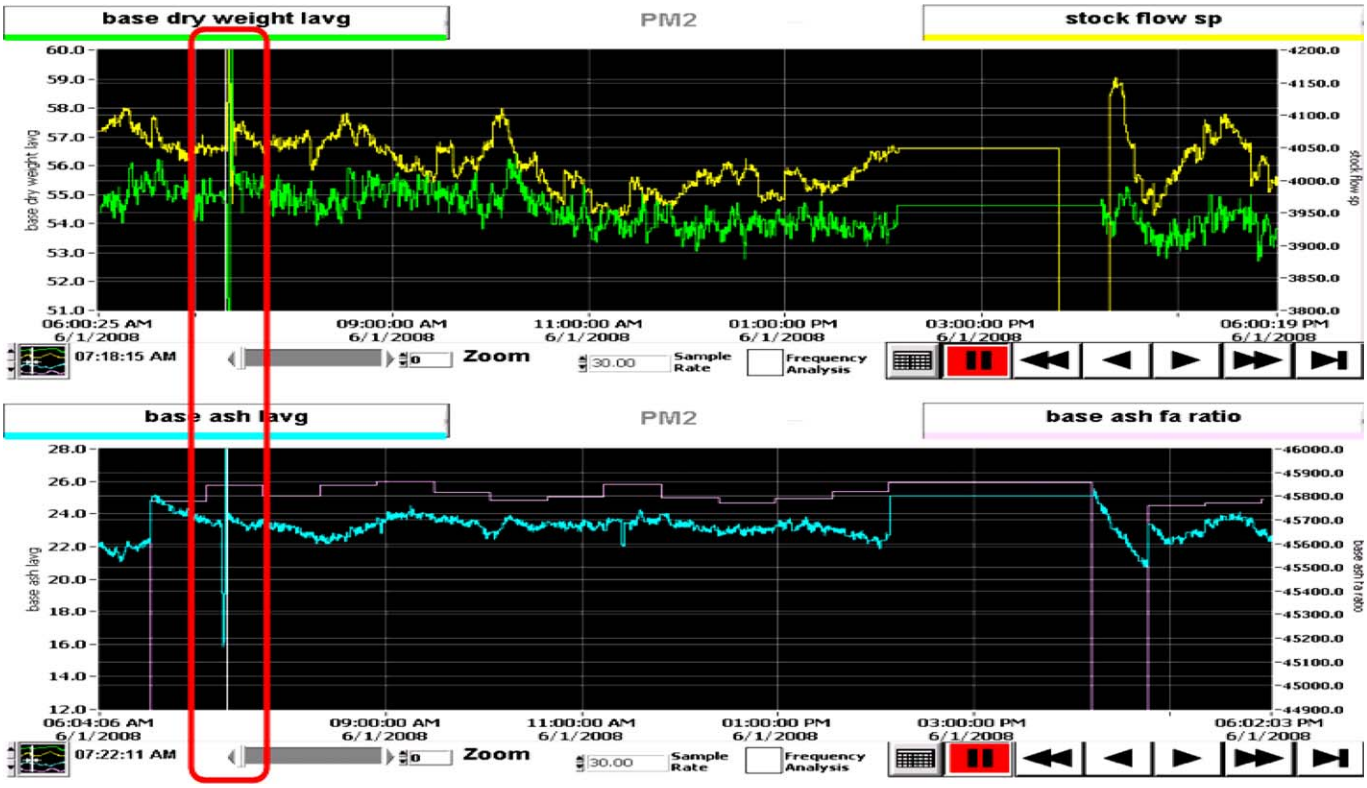

Fig. 5. A temporary drop in Percol flow leads to a 30% disturbance on Base Sheet Ash.

(See Fig. 4). The first problem is that when the break has ended, the scanner takes 5 min to warm up and there are no measurements during this period even though there is no break. There is also a 'jump' in the Ash reading which occurs around 30 min after scanner's first live reading. This is caused by the scanners 'initializing' every 30 min. Part of this process is to calculate an 'air gap' temperature correction factor which is used by the scanner to adjust its reading. When the scanner has been operating for a while the temperature of the air in the gap between the sheet and the scanner is relatively steady. However, when paper is first fed through the scanner, the temperature of the air will rise rapidly from room temperature to around 70°. Consequently at the first recalibration point after the break, there is a large jump (correction) in the Base Sheet Ash reading.

It is clear that an analytical approach will not be practical when faced with these obstacles, making the estimation a suitable alternative. The dominance and popularity of the least squares regression can be ascribed, at least partially, to the fact that the theory is simple, well developed and documented. The least squares regression is optimal and results in the maximum likelihood estimators of the unknown parameters of the model if the errors are independent and follow a normal distribution with mean zero and a common variance (Narula & Wellington, 1982). Nevertheless, the least squares regression is distinctively sub-optimal in many non-Gaussian situations, especially when the errors follow distributions with longer tails (Rice & White, 1964). For the regression problems, Huber (1973) stated that 'just a single grossly outlying observation may spoil the least squares estimate, and, moreover, outliers are much harder to spot in the regression than in the simple location case'. The outliers occurring with extreme values of the regressor variables can be especially disruptive. Least absolute value (LAV) regression on the other hand overcomes these drawbacks and provides an attractive alternative. It is less sensitive than least squares regression to extreme errors, has implicit mechanics to reject bad data and does not require a normal distribution of data which is unrealistic in practical situations (Claerbout & Muir, 1973).

2.1. Affect of bad data on the LS and LAV solutions

To see why the LAV approach offers an implicit bad data rejection property, let x_i be an arbitrary number and define γ_1 as the value of γ which minimizes the sum of the squared differences (i.e. the least squares solution) between γ and x:

$$\gamma_1 := \arg\min_{\gamma} \sum_{i=1}^{N} (\gamma - x_i)^2, \tag{1}$$

where N is the total number of data points. It is straightforward to find the minimum by setting the partial derivative of the sum with respect to γ equal to zero:

$$0 = \sum_{i=1}^{N} 2(\gamma_1 - x_i), \quad \Rightarrow \quad \gamma_1 = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{2}$$

Notice that the solution is the mean of the data observation values. In this case if any of the x_i are very large (say due to a measurement error) it will directly effect the solution. To see the effect of the same change in the LAV case, define now γ_2 to be the solution obtained by minimizing the least absolute value as

$$\gamma_2 := \arg\min_{\gamma} \sum_{i=1}^{N} |\gamma - x_i|. \tag{3}$$

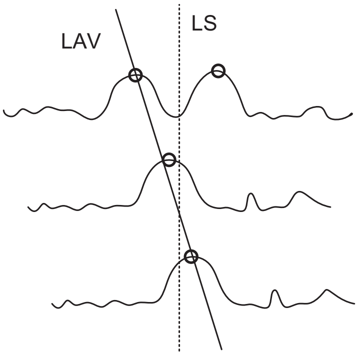

Fig. 6. Picking the maximum of a trace when the choice is ambiguous.

To find the minimum, the partial derivative with respect to γ is set equal to zero:

$$0=\sum_{i=1}^{N} sgn(\gamma_2-x_i), \tag{4}$$

where

$$sgn(x)=\begin{cases} +1, & x>0, \\ -1, & x<0, \\ 0, & x=0, \end{cases} \tag{5}$$

The solution defines γ_2 as the median. Relating the mean and the median to the LS and LAV problems helps to easily see why the LAV approach is inherently much more robust. For example suppose that there are three observations from an experiment $\{1, 1.1, 0.9\}$. The LAV solution in this case is 1 and the LS solution is also 1. However, if due to experimental error, or a measurement anomaly, the observations became $\{1, 100, 0.9\}$, then the LAV solution is *still* 1, but the LS solution will change to ≈ 30. This is because in the LAV case, as long as the *bad* data remains on the same side of the median, it has zero effect on the solution. In the general case, the solution of the LAV problem will 'exactly' satisfy as many equations out of the number proposed, as there are unknown quantities, and the remaining equations simply help to 'determine' the choice. The set of equations used to determine the LAV solution are referred to as the 'optimal-basis'. This difference in treating inconsistent data may have important physical implications. Suppose that one is trying to find the maximum of a cross-correlation function. As illustrated in Fig. 6, it is quite common to discover that the maximum of the function cannot be unambiguously picked. When neither maximum is consistent with the rest of the data, the LAV solution will choose a line which picks either one or the other. The LS solution on the other hand regards the two data values as one of double weight placed at the midpoint between the two. For the example, at hand this is inappropriate because the midpoint is clearly not a maximum. The median also has the property of stretch invariance, meaning any monotonic transformation preserves order and leaves the median pointer unchanged. This property can be useful when it is not clear what should be the correct scaling or averaging performed on the data.

2.2. The LAV regression problem

Consider the value of an experiment which, when taking into account the error may be represented as follows:

$$y_i=\sum_{j=1}^{m} x_{ij}\beta_j+\varepsilon_i, \quad i=1,2,\ldots,n, \; j=1,2,\ldots,m. \tag{6}$$

Let $(x_{i1}, x_{i2}, \ldots, x_{im}, y_i)$ be the i th observation (n observations in total) and let $b_0, b_1, \ldots, b_m$ be the estimates of $\beta_0, \beta_1, \ldots, \beta_m$ estimated by minimizing the sum of absolute values of the differences between the values of $\hat{y}$ and y. So the LAV problem is therefore,

$$\min\left(\sum_{i=1}^{n} |y_i-\hat{y}_i|\right). \tag{7}$$

The LAV regression problem actually predates the least squares approach. Its use, however, was not popular for many years because unlike least squares, the solution is difficult and not straightforward. It was not until the implementation of LP algorithms on the digital computer that LAV estimates could be obtained for problems of reasonable size (Armstrong & Kung, 1982). Charnes, Henderson, and Cooper (1953) appear to be the first to point out that (7) could be rewritten as a LP problem. Subsequently several other LP versions of the LAV problem were developed by Charnes, Ferguson, and Cooper (1955), Gonin and Money (1989), Li (1998), and Gentle, Narula, and Sposito (1987). The best known model for the linear regression model is the primary linear regression model developed by Charnes et al. (1955) and Ignizo (1995). In this model, the aim is to minimize the overall absolute difference between observations and estimation values, in other words, minimizing the overall error terms. In this respect, goal programming is used to develop the LP model which will be used to minimize the overall total positive and negative deviations (Cerezci & Gokpinar, 2005).

Evidently, the computation of the LAV solution is non-trivial. In particular it is often the case that modeling and estimation needs to be implemented on older legacy systems or within a limited programming environment. Under these circumstances it is difficult to solve the LP either using the simplex method or using some sort of interior point algorithm. There are also speed of computation concerns which become important in online estimation applications. Indeed, in case of the mill concerned, the modeling and estimation programme need to be implemented on a legacy Honeywell DCS system from the 1980s using the Measurex Programming Language (MPL). It was therefore necessary to find an alternative and efficient means of solving the LAV problem.

In light of this, Differential Evolution represents a suitable algorithm for the computation of the LAV solutions. Being a direct search method means that only elementary operations (addition and multiplication) are required to find the solutions, and at the same time since they are variants of Evolutionary algorithms, the implicit parallel processing makes them much faster than single point direct search methods such as bisection or Golden Section search.

2.3. Differential evolution

Differential Evolution was recently proposed by Storn and Price (1997) as a powerful direct search evolutionary algorithm. Amongst the DE's advantages are its simple structure, ease of use and speed of convergence. DE has consistently been ranked as the best search algorithm in several case studies. In Storn and Price (1997), DE is compared with the annealed version of the Nelder and Mead Simplex algorithm and the Adaptive Simulated Annealing. The DE outperformed both of these algorithms and it was the only one to converge for all of the test problems. In Babu and Angira (2002), the DE was compared with the simulated annealing M-SIMPSA algorithm (Cardoso, Salcedo, Azevedo, & Barbosa, 1997) and Genetic Algorithms, and was again found to be the only one converging on all problems and providing a better solution when other converged as well. Lin, Hwang, and Wang

(2001) applied several MINLP case problems and compared the DE with powerful MINLP solvers. They found that the DE was the only one able to simultaneously solve all the problems to optimality and satisfy the constraints. In Lampinen and Zelinka (1999) report that the DE outperformed several algorithms, including the Branch and Bound, Sequential Programming, Sequential Linearization Algorithm, Simulated Annealing, GAs, Evolution Programming and ESs, when tested on Sandgren's problem set.

The population of a DE is subject to three operators of mutation, crossover and selection. Price and Storn proposed several variants of the basic DE which are denoted using the notation *DE/vec/num/mode*. *vec* is the vector to be mutated, which is either a randomly chosen vector,'rand', or ' best' $=\mathbf{x}^{*k}$; the best vector of the current generation. *num* is the number of difference vectors used in the mutation, which is either 1 or 2 and *mode* is the method of crossover used. For independent binomial experiments of the genes, this is set to 'bin'. The initial population of the DE is generated uniformly to span the initial domain of $\mathbf{x}$,

$$P_i^{(0)} \leftarrow \mathbf{x}_i^{(0)} = \mathbf{x}^L + \boldsymbol{\rho}^n \,|_0^1\, (\mathbf{x}^U - \mathbf{x}^L), \quad i = \{1, \ldots, n_p\}. \tag{8}$$

Subject to,

$$\mathbf{x}^L \prec \mathbf{x}^U \in \mathbf{R}^n, \tag{9}$$

where $\mathbf{x} \prec \mathbf{y}$ iff $\forall x_i, y_i : x_i < y_i$.

Mutation: Unlike EAs, in DE the mutation amount is derived from a difference vector which is calculated using members of the current populations. For the standard $DE/rand/1/\ldots$ strategy, this is as follows:

$$\mathbf{tr}_i^{k+1} = \mathbf{x}_{best}^k + F(\mathbf{x}_s^k - \mathbf{x}_t^k), \tag{10}$$

where F is the weighting factor and $s = \rho \vdash_0^{n_p}$ and $t = \rho \vdash_0^{n_p}$ are randomly generated integers such that $s \neq t \neq i$ (i.e. the i th trial vector cannot self-mutate). Since the DE operates an elitist selection scheme, $\mathbf{x}_{best}^k$ also equals the best ever solution from the start of the search to the current iteration. The inherent 'self-adaptation' of the DE is seen. As the population converges to an optimum, any randomly chosen difference vector will become smaller in magnitude. Eventually when all members converge to a single solution, the difference vector will be zero and the mutation operator will be disabled all together. Therefore, the actual amount of mutation at iteration k is not only determined by F but also by the population diversity.

Crossover: For each gene of the trial vector a random number is generated, if this is lower than the crossover rate C_r set by the user, then the gene of the new trial vector is used, if not the gene of the original trial vector $\mathbf{x}_i^k = (x_{i1}^k, \ldots, x_{in}^k)$ is kept,

$$tr_{ij}^{k+1} = \begin{cases} x_{ij}^k & \text{if } \rho \,|_{0_{ij}}^1 < C_r, \\ tr_{ij}^{k+1} & \text{else}, \end{cases} \qquad j = \{1, \ldots, n\}. \tag{11}$$

Selection: The selection in DE is deterministic and simple: The resulting trial vector will only replace the original parent if it has a lower objective function value,

$$\mathbf{x}_i^{k+1} = \begin{cases} \mathbf{x}_i^k & \text{if } f(\mathbf{tr}_i^{k+1}) > f(\mathbf{x}_i^k), \\ \mathbf{tr}_i^{k+1} & \text{else}. \end{cases} \tag{12}$$

Although the selection pressure is only one, the best individual of the next generation will be at least as fit as the best individual of the current generation. Similar to a $(\beta,\beta){-}ES$, the spread is one, but the best individual can get better whilst the worst remain the same. Therefore, there is less loss of diversity than in truncation selection used in ES and this insures a relatively large selection variance.

2.4. Adaptive DE algorithms

DE has three control parameters; n_p, the population number, F the mutation weighting factor and C_r the crossover rate. The difficulty with the use of DE is that their choice is mainly based on empirical evidence. Although the standard DE is inherently adaptive, its sensitivity to the control parameters is well known (see Gamperle, Muller, & Koumoutsakos, 2002 for a general treatment and Lampinen & Zelinka, 2000 for the stagnation problem).

There have been a handful of generic adaptive DEs developed in the past. Some algorithms such as Lampinen (2002) or Zaharie (2003) rely heavily on human intervention. For example, Zaharie (2003) proposes a feedback update rule for F that is designed to maintain the diversity of the population at a given level (and thus stop the search converging prematurely). However, the method requires the user to tune a γ parameter which determines the update law for F. Therefore, as the author points out himself, although the algorithm seems to perform better than the self-adaptive DE proposed in Abbass (2002), in actual fact the problem merely changes from choosing F to choosing γ. Recently a self-adaptive algorithm was developed (Nobakhti & Wang, 2007) which takes into account these inadequacies and involves no feedback law. This removes the need to determine a generic rule, or replace tuning of F with tuning some other parameters (for example the mean and variance of a distribution). Unlike the standard DE, in the proposed algorithm each $\mathbf{x}_i$ has its own unique value of F_i^k which is subject to change during the evolution. The algorithm referred to as Random Adaptive Differential Evolution (RADE) is a generic optimization algorithm which is applicable for the generic class of nonlinear and discontinuous problems. However, in this work, some of the special features of the LAV problem are taken into consideration to develop a faster algorithm based on RADE which will be referred to as *f*ADE.

2.5. A fast adaptive DE for LAV computation: fADE

Although the LAV problem is not a smooth function, one of its key geometric features is that when a linear regression model is used, it is always convex. The convexity of the cost function greatly reduces the effectiveness of the crossover operator. In this case, the crossover operator will only yield a better result if the two parent vectors form a difference vector which is not orthogonal with the main function axis (See Fig. 7). Clearly this

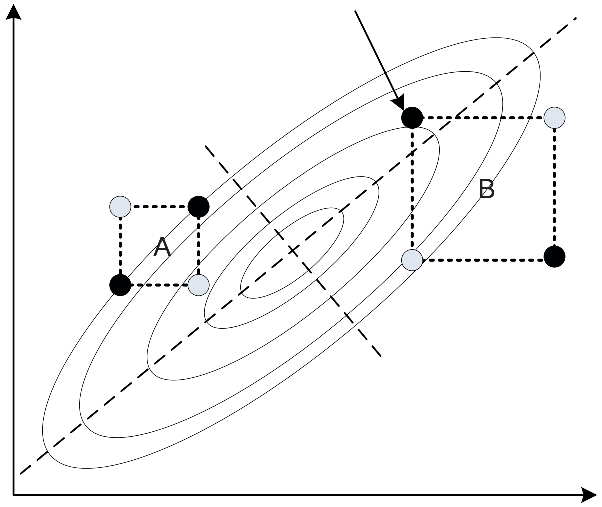

Fig. 7. The crossover operation in case of convex cost function. The parent vectors are shown in gray and the trial vectors are shown in black. Note that among the four crossover operations, only one leads to an improvement in this case.

is a special case and the computational cost of performing the crossover operation outweighs the potential benefits arising from the above geometry. Therefore, in the *f*ADE algorithm, the crossover operator is removed completely.

Recall that in the general case of multi-modal function, an EA needs to maintain a large population to avoid entrapment in the function's local minima. This emanates from the assumption that each member of the population might possess a set of 'good' genes which in later generations and through recombination with other members will lead to an overall better solution. Since the LAV problem is convex, the need to maintain a large population is evidently reduced. Yet, if a small population is used at the start of the search, there is a chance that there will not be sufficient distribution of initial choices around the global minima, causing the search to stagnate. In light of this, the second modification is to dynamically reduce the size of the population. This is quite safe to do since in the LAV case the worst member has no useful genetic information and can be safely discarded at each iteration. For an m dimensional problem, a lower bound of m members has been defined so that the size of the population will never drop below the dimension of the problem. The benefits of this dynamic reduction are that it ensures good initial coverage of the surface, but then reduces the size of the population to avoid unnecessary computation at the later stages. A complete pseudo-code for *f*ADe is given in the appendix.

3. Benchmark and results

3.1. Comparison of *f ADE* and interior point algorithms

First, the regression coefficients of the LP and *f*ADE algorithms are estimated for benchmarking purposes. Each experiment is repeated 10 times and the results given here are the average of those 10 runs. The experiments were performed on an Intel Pentium Core 2 Due 1.83 GHz system with 2 Gb of RAM. The CPU times (in seconds) and total memory allocation in kb (in MATLAB) are recorded. In case of the LP problem, MATLAB standard function `linprog` was used (fixed to `LargeScale` option to always use the interior point algorithm) which uses Mehrotra's (1992) predictor–corrector infeasible interior point method. In all cases, n is the number of observation data points, and m is the dimension of the problem. In case of the LP problem, an interior point (IP) algorithm was used to solve the problem. For the scale of problems considered here, the IP algorithm was faster than the Simplex method which is the other common approach to solve the LP. For *f*ADE, the initial population was always fixed at 10 m and the algorithm had a 100% convergence rate for all the experiments; then again this is a convex problem. For both algorithms the same function tolerance was used. The results are given in Tables 1 and 2 and are shown graphically in Fig. 8.

Note that *f*ADE has strikingly different properties from the LP algorithm. The LP is much less sensitive to the dimension of the problem, than *f*ADE. However, the situation is completely reversed when the number of observations are increased. In this case, *f*ADE is much less sensitive and the complexity rises notably slower. It is also noteworthy to underline the huge difference in memory requirements which is approximately three orders of magnitude. This stems mainly from the fact that to solve the LP, large data matrices need to be created, where as with DE direct evaluation is performed.

For example, in case of five regression coefficients and 2000 observations, the proposed algorithm uses 96 kb of memory, versus 64 Mb required for the data matrices of the LP formulation (This is simply the allocation space required for the input data, and does not include the temporary memory required during computation).

Table 1
Benchmark results for varying problem dimension.

$n=1000$	*f*ADE	LP (Interior point)
$m=2$	CPU = 0.0344, Mem = 24	CPU = 0.2641, Mem = 16, 040
$m=3$	CPU = 0.0688, Mem = 32	CPU = 0.2672, Mem = 16, 048
$m=4$	CPU = 0.1266, Mem = 40	CPU = 0.2813, Mem = 16, 056
$m=5$	CPU = 0.1812, Mem = 48	CPU = 0.2875, Mem = 16, 064
$m=6$	CPU = 0.2953, Mem = 56	CPU = 0.3016, Mem = 16, 072
$m=7$	CPU = 0.5078, Mem = 64	CPU = 0.3172, Mem = 16, 080
$m=8$	CPU = 0.6656, Mem = 72	CPU = 0.3234, Mem = 16, 088
$m=9$	CPU = 0.8250, Mem = 80	CPU = 0.3359, Mem = 16, 096
$m=10$	CPU = 1.1516, Mem = 88	CPU = 0.3516, Mem = 16, 104

CPU refers to the CPU time in seconds (s) and Mem refers to the allocated memory size in kilobytes (kb).

Table 2
Benchmark results for varying number of observations.

$m=5$	Proposed algorithm	LP (Interior point)
$n=500$	CPU = 0.0859, Mem = 24	CPU = 0.1922, Mem = 7884
$n=800$	CPU = 0.1266, Mem = 38.4	CPU = 0.2266, Mem = 10, 291
$n=1100$	CPU = 0.1047, Mem = 52.8	CPU = 0.3172, Mem = 19, 430
$n=1400$	CPU = 0.2047, Mem = 67.2	CPU = 0.4359, Mem = 31, 450
$n=1700$	CPU = 0.2266, Mem = 81.6	CPU = 0.5703, Mem = 46, 349
$n=2000$	CPU = 0.2594, Mem = 96	CPU = 0.7328, Mem = 64, 128
$n=2300$	CPU = 0.2375, Mem = 110.4	CPU = 0.9125, Mem = 84, 787
$n=2600$	CPU = 0.2484, Mem = 124.8	CPU = 1.1000, Mem = 108, 330
$n=2900$	CPU = 0.3406, Mem = 139.2	CPU = 1.3250, Mem = 134, 750
$n=3200$	CPU = 0.3681, Mem = 153.6	CPU = 1.5703, Mem = 164, 040

CPU refers to the CPU time in seconds (s) and Mem refers to the allocated memory size in kilobytes (kb).

Clearly *f*ADE has an overall advantage in terms of memory usage. It can also be argued that for the application it was developed, it is also superior in terms of computational efficiency. Namely, when estimating in real life industrial applications, one is generally dealing with a few of coefficients, but many observations. For example, the paper machine's DCS system will log plantwide data every 5 s. This results in over 17,280 observations per 24 h. For a four dimensional problem, this will take the LP approach an estimated 2 min to compute the LAV model, but only 1.3 s using *f*ADE. This is a very significant speed advantage, especially when noting that the Interior Point solver used for the LP problem is already one order of magnitude faster than the Simplex method which itself is reputed to be a fast algorithm. The speed of the proposed algorithm will make it possible to implement LAV problems online, in the same way that been traditionally done for many years with LS models. Moreover, the algorithm is very light on memory usage and is able to compute the LAV solution using only elementary operations of addition and multiplication.

3.2. Base sheet estimation

In the case of the paper machine considered there are around 200 variables in the Wet End process line that continue to be logged during the break as long as Stock still flows onto the Wire. Although it might seem that using as much of the data as possible will lead to an improved model, this is not the case for several reasons. First, not all data is logged by the same logging system and not all will be in real time. Consequently data corresponding to the same event, will be scattered in several databases with a different time stamp in each database. Considering too many variables will therefore necessitate a tedious synchronization process.

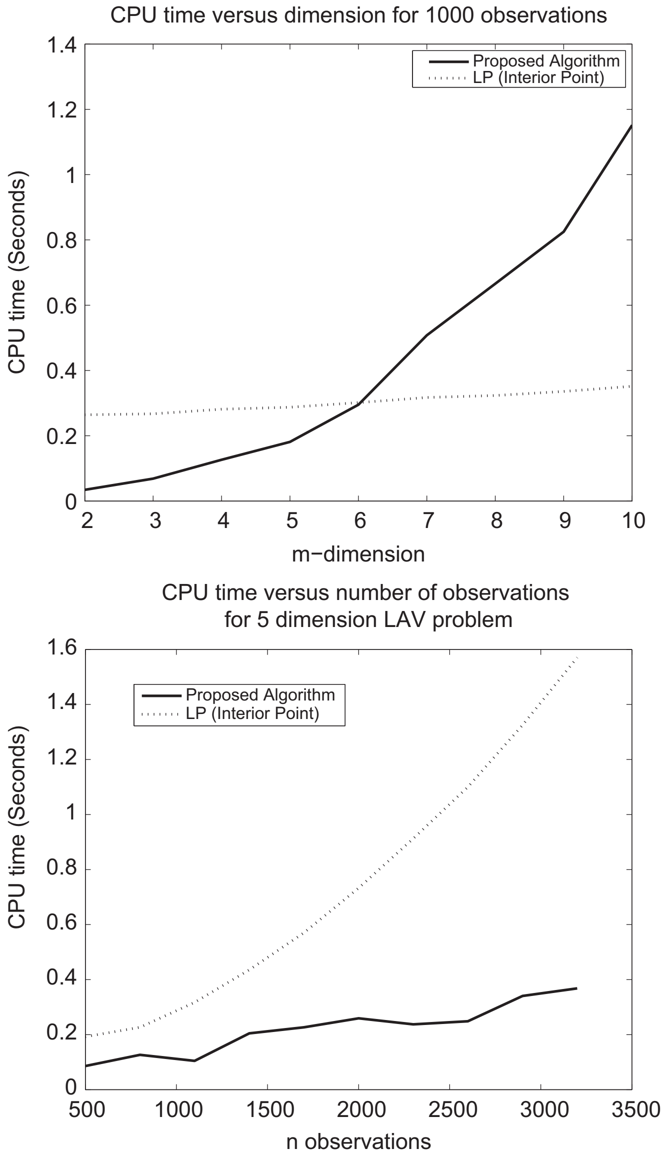

Fig. 8. CPU time versus LAV dimension ($n=1000$) shown on the left, and CPU time versus the number of observations ($m=5$) shown on the right.

Secondly, each of these variables will have many perturbations and data anomalies associated with them. Using too may of these will lead to an unnecessary amount of large quantity of bad data. Moreover, and most importantly machine operators will not be able to effectively use a complex model which requires 100 s of variables to estimate the Base Sheet Ash.

Consequently the aim was to develop the model using as few variables as possible. The results of the variable selection process are omitted here. However, following a process of correlation, PCA, Frequency Power spectrum analysis and physical system considerations, four variables were selected for the modeling purposes, Headbox Total Consistency, Headbox Ash, Whitewater Total Consistency and Whitewater Ash. Fig. 9 shows the relevance of these variables at the Wet End and particularly the Wire. The aim is to arrive at an estimate of the Base Sheet Ash, which is measured using the Base Sheet scanner situated at the end of the Wet End after the first set of dryers. A key consideration is that the Wire is the only section of the Wet End at which solids (fines or fibers) can be removed from the web. Thereafter, the only substance which is removed from the web is water. Thus, the Ash content of the web which is leaving the Wire, is directly correlated to the Base Sheet Ash measured at the scanner. Any Ash which leaves the Headbox, will either stay on the Wire, or be drained away with whitewater. This suggests a possible easy route to determine the Base Sheet Ash: perform a simple mass balance at the Wire. Unfortunately the mass balance is not possible since although the total flow of stock onto the Wire is accurately measured, the amount of drained whitewater is not measured and therefore knowing its consistency will not help to determine the total amount of fillers which have drained away from the Wire.

The two Headbox measurements are made using spectroscopy techniques and whilst the Total consistency measurement is generally accurate, the filler reading is not very accurate as it is not measured directly. Instead, a particle size threshold is defined, smaller than which all solid particles are considered to be minerals. Whitewater is the water which is drained on the Wire and is collected to be used as ‘sweetener’ back in the pulping process mainly for dilution. The Whitewater is so-called as the water that drains is cloudy due to the fine mineral particles which are not stopped on the wire mesh and drain away. Any filler particle which has not flocculated or bounded with filler particles will be drained away because the Wire mesh size is not sufficiently small to retain these fine particles.

The LAV model is updated online, therefore it is able to track the operating point changes of the paper machine. This means no change to the model is made when a change of grade occurs. During normal operation, data from the four variables are

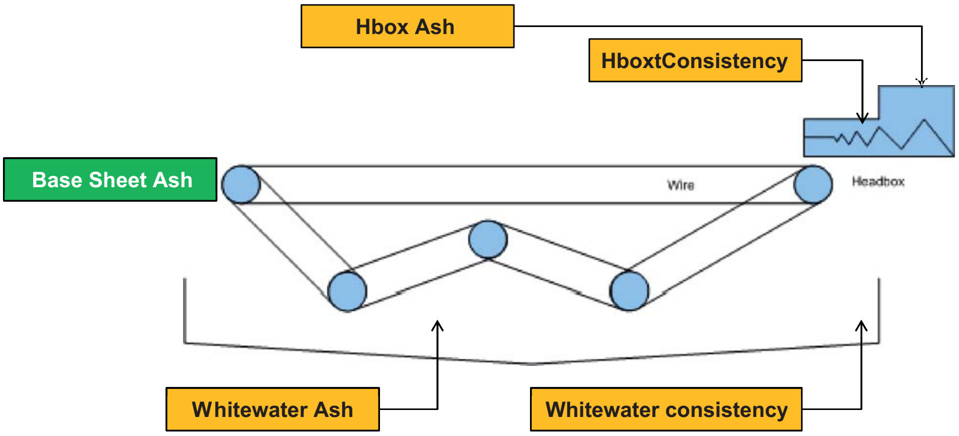

Fig. 9. The four variables used for Base Sheet Ash estimation.

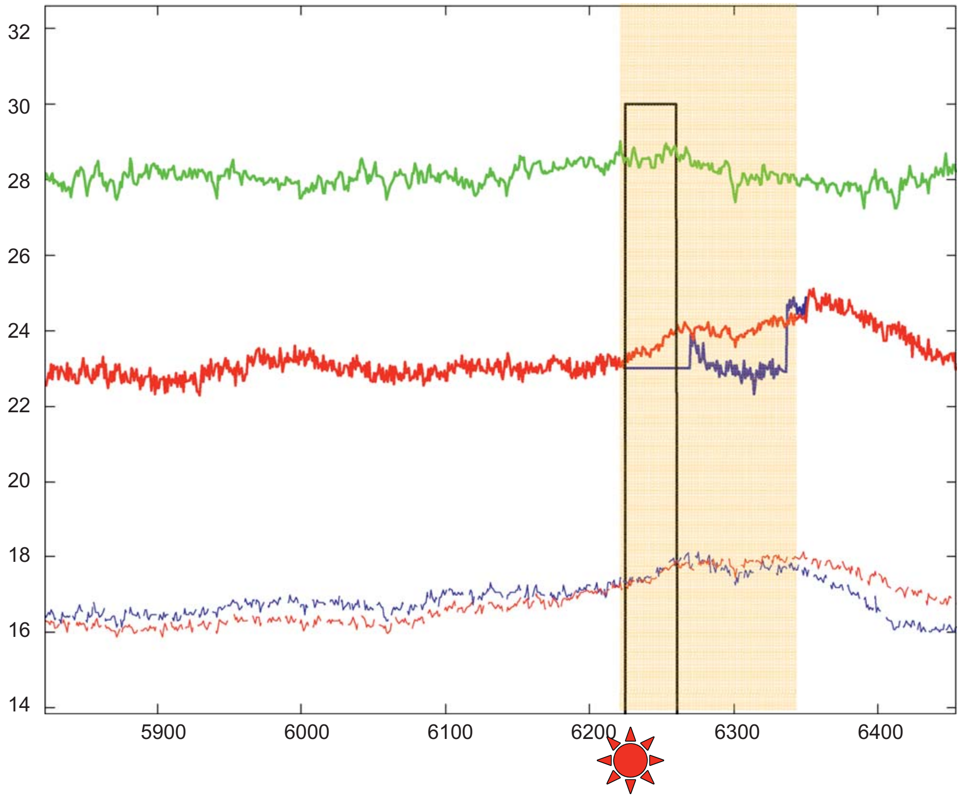

Fig. 10. Black: Wet End break, green: stock flow, solid-blue: base sheet scanner ash, dashed-blue: headbox ash, dashed-red: headbox consistency, solid-red: base sheet ash model estimate, red-sun: new model generated, shaded region: prediction interval. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

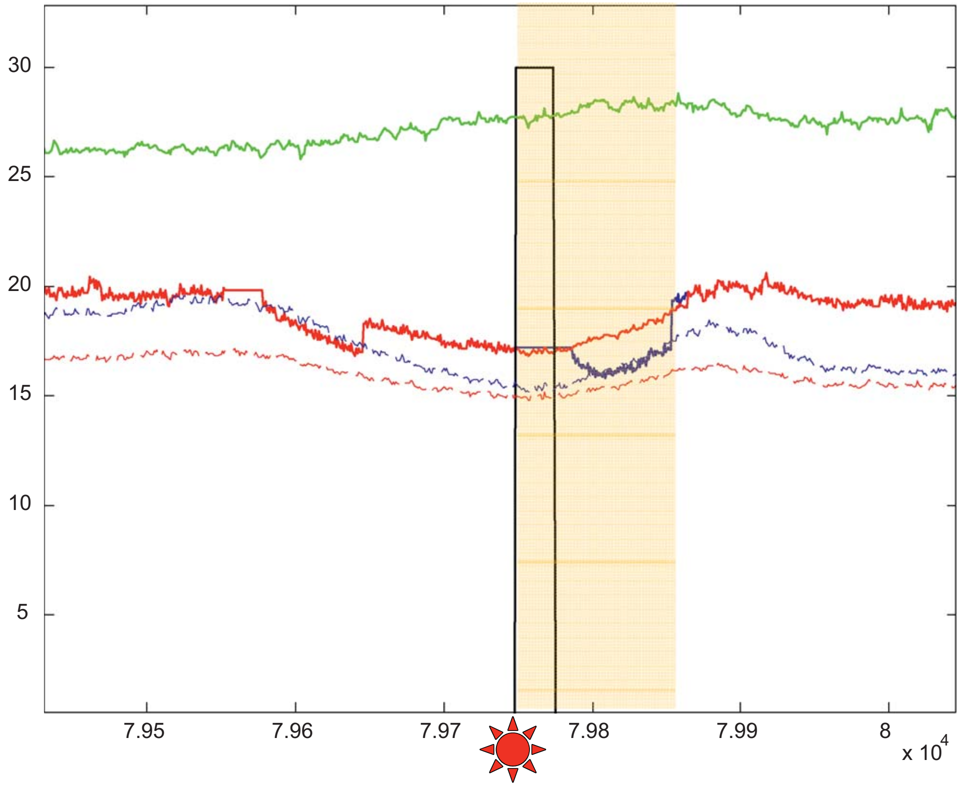

Fig. 11. See Fig. 10 for legend.

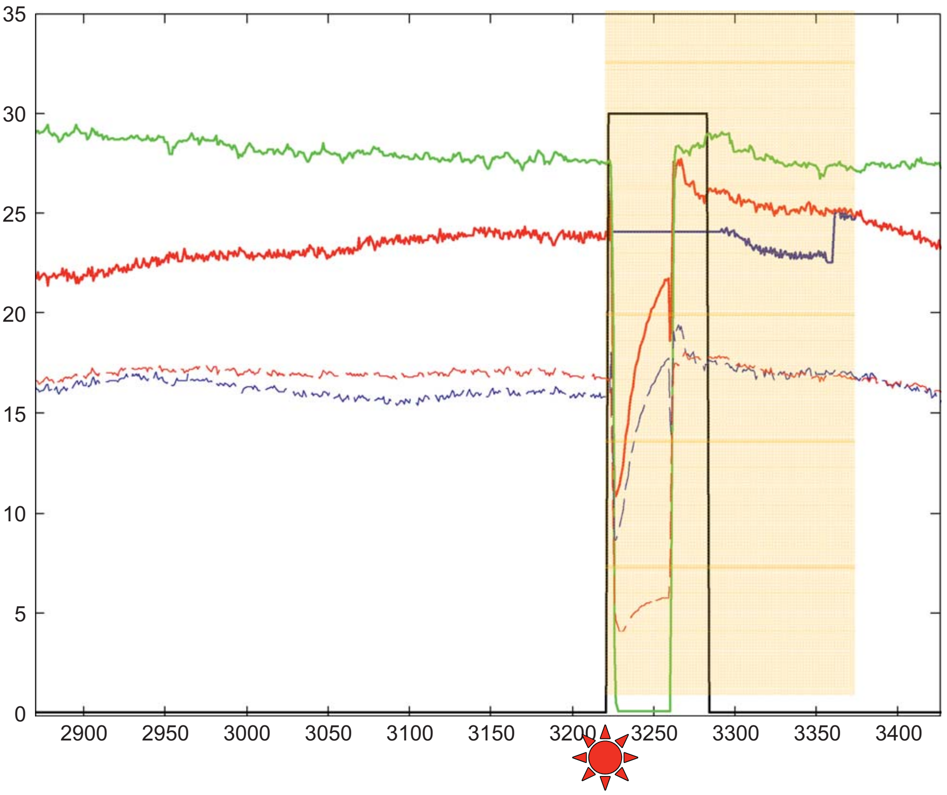

Fig. 12. See Fig. 10 for legend.

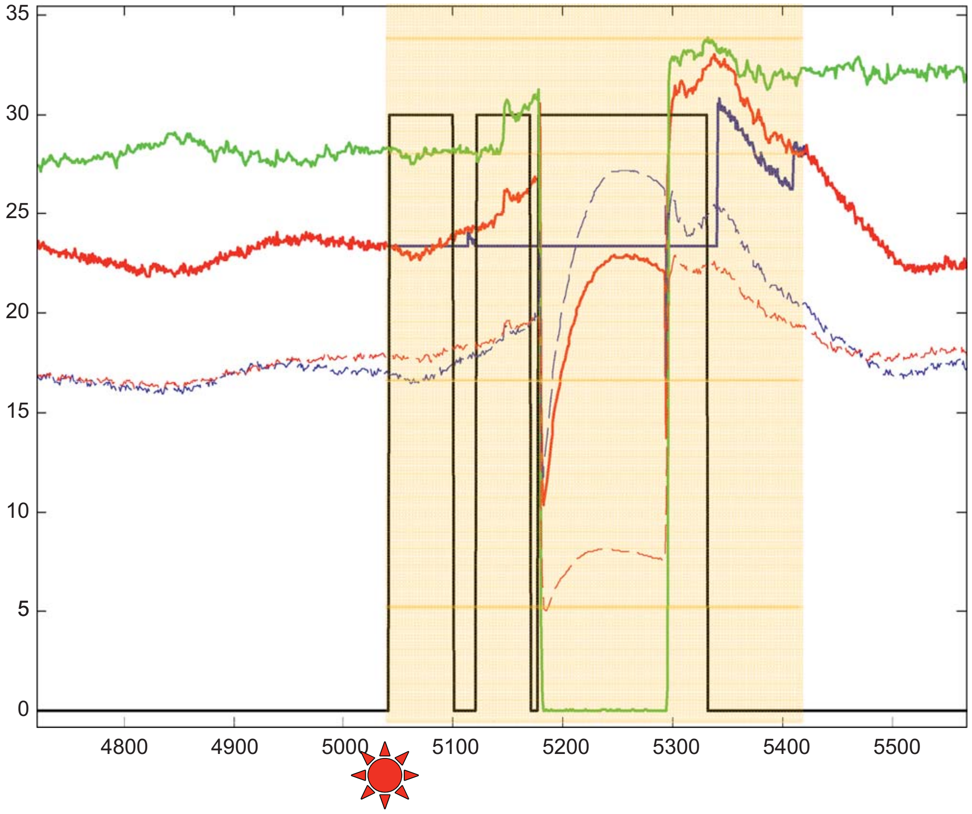

Fig. 13. See Fig. 10 for legend.

collected and logged and the output of the LAV model is forced to follow the actual sensor measurement. Once a break occurs, a new LAV model is generated. Using the proposed *f*ADE algorithm, the time is takes to generate a new model is less than the sampling time of the DCS controller (5 s). Therefore, the Base Sheet Ash estimate is available to the controller in the next sample time

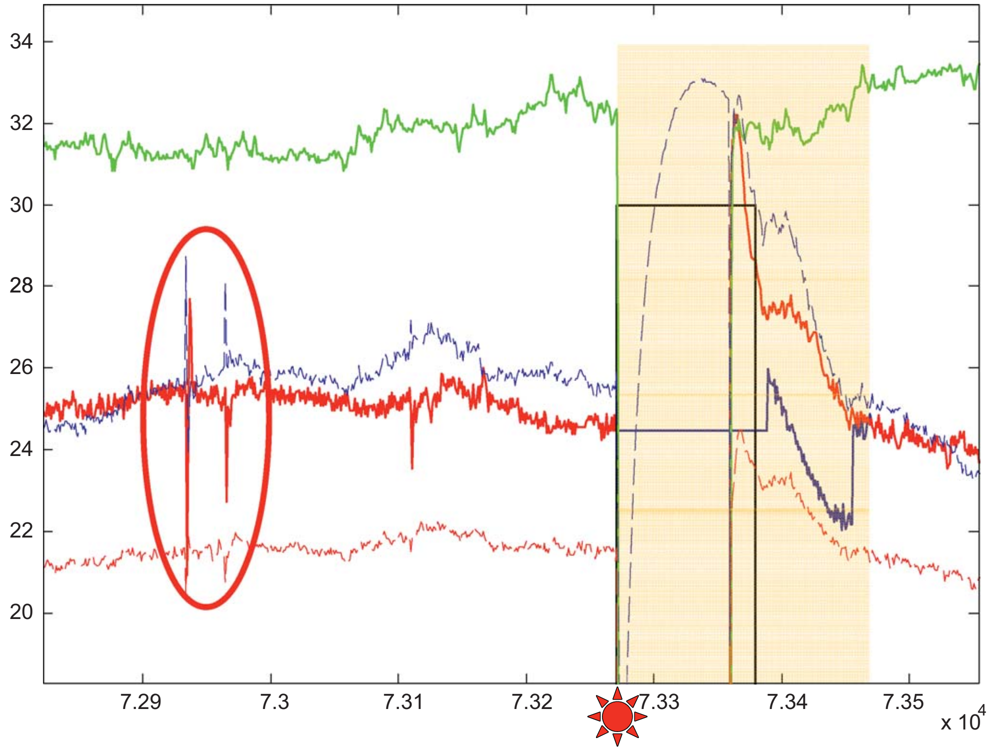

Fig. 14. See Fig. 10 for legend.

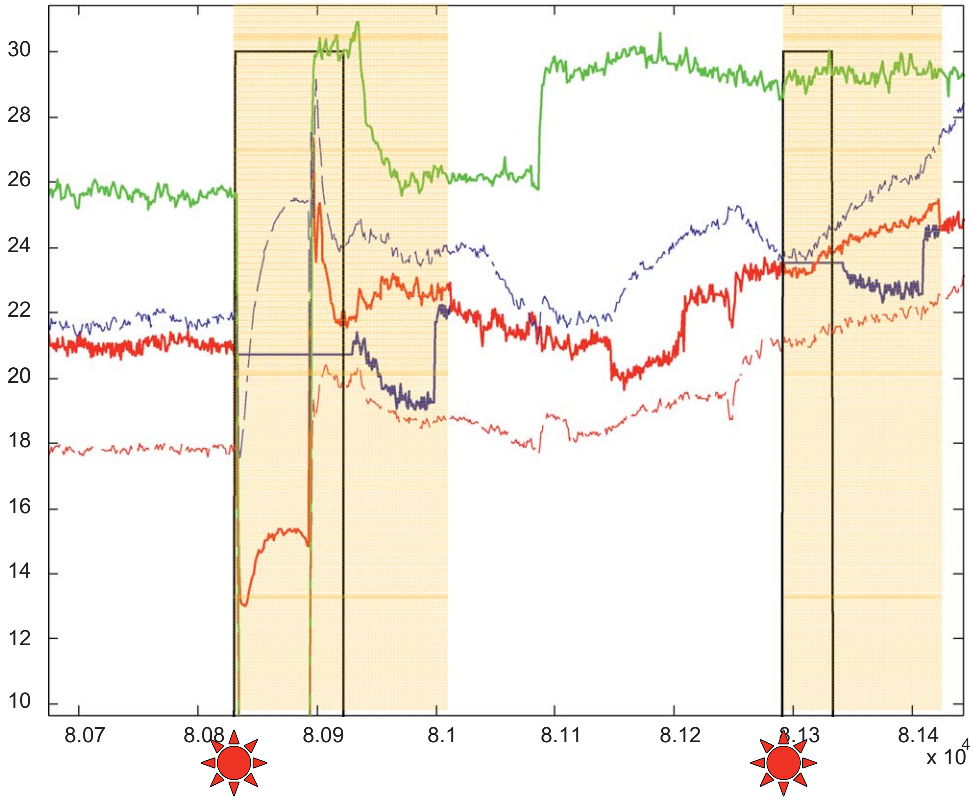

Fig. 15. See Fig. 10 for legend.

(in the break) and the controller will see no interruptions in its signal feed. Once the break has ended, the model estimate continues to be used until the sensor has completed its first standardization after the break, at which point control is handed back to the scanner. This procedure will be repeated at the time necessary. The results of the application of the proposed methodology are illustrated in Figs. 10–15. Numerous different studies of the model have been made relating to different operating points and product grades, to assess the stability and robustness of the model. The Ash estimates values were also validated with manual lab readings.

Fig. 10 is taken during a run of 90 gsm coated fine paper. The machine was running steadily for around $3\frac{1}{2}$ h when a Wet End break occurs. The break itself lasts for just over 20 min and stock flow is maintained during the break. It is never possible to state precisely why a break has occurred but in this case it might be related to the rise in Headbox Ash levels just prior the break. As described previously, during the break, the Base Sheet Ash measurement is frozen. Five minutes after the machine has recovered from the break, the Base Sheet scanner is online again, but due to the cold air gap temperature, the reading is false. This is corrected when the scanner has warmed up with the resulting jump in the Base Sheet Ash reading. Meanwhile when the break occurred, a new model was generated at the time indicated by the sun symbol. The model takes < 1 s to compute which means that the next estimate of the Base Sheet Ash is available before the next controller sample time. The model is used to estimate the Base Sheet for around 1 h. Once the scanner has recalibrated, control is handed back to the scanner. Notice that when the scanner has recalibrated, the reading jumps to almost exactly the same value as the estimate which shows the high precision of the estimate. Fig. 11 shows a similar situation from a run of 85 gsm fine paper.

Fig. 12 shows another run of 90 gsm fine coated paper. In this figure, the machine was running fine for around 3 h when the break occurs. The difference between this instance and the previous two examples is that the break is quite severe and the operators have stopped the stock flow onto the Wire. Consequently, even though a new Ash model is generated, there are no estimates during the first 20 min of the break. In this instance it takes 20 min for the main fault to be repaired, and reestablish stock flow, although the machine is still in a break (i.e. paper is not being made). As soon as stock flow has reestablished, the model is able to provide Base Sheet Ash estimates which as before continues to be made for around 1 h before the scanner has come back online and has recalibrated. Once the scanner has recalibrated, its reading is almost identical to the estimate given from the model. Fig. 13 is a similar example which also illustrates the typical 'train' of breaks behavior referred to earlier, where a first minor break can cause subsequent major breaks due to the disturbances propagating through the Wet End. When there is not at least 1 h of data since the last break a new model is not generated as in this case. Instead, the model which is used to estimate the Base Sheet Ash after the third break, is the model generated before the first break.

The main reason given for the use of the LAV model as opposed to the LS model was the inherent robustness and bad data rejection properties of the LAV model. Fig. 14 gives an explicit example of the performance of the LAV model in the presence of bad data. This Figure captures a run of 100 gsm fine coated paper. Here, around 2 h prior to the break, the machine suffers two major disturbances lasting for a few minutes each. A lesser (but still large) disturbance is also seen an hour later. This particular case is caused by retention aid chemical disturbances which is obvious from the reverse spikes in the Headbox and Base Sheet Ash trends. The disturbances eventually cause a major break to occur during which stock flow is stopped and the break itself lasts for around an hour. In spite of this, it is clearly evident that the Base Sheet estimate which is generated using the model computed from the disturbed data is still very accurate. A consequence of the inherent robustness of the LAV approach is that the model does not need a lot of data to be relatively accurate. Fig. 15 shows a run of two breaks, which are both grade change induced. A grade change break is common and it occurs when the paper machine is changing from one grade of paper to another grade. In this instance, the machine was making 90 gsm paper and it was scheduled to shift production to 100 gsm paper. The paper breaks at the first attempt to change grade which causes a large disturbance to the machine. Upon resumption of paper making, the operators first switch back to the previous grade and less than an hour later make a second attempt at changing grade. The machine is not able to recover fully and a secondary break occurs an hour and half later. Note that in this case, the second model is generated with little data, which itself was split between two different operating points. Nonetheless it is still relatively accurate although by the end of the second break the model was slightly over estimating.

4. Conclusions

As described previously, paper making is a complex process. This is further complicated by the occurrence of paper breaks. A paper break will lead to parts of the paper machine being controlled in open loop. Open loop control of the paper machine, especially the Wet End will lead to large disturbances to the nominal machine operating conditions. These deviations will cause poor paper to be produced once paper making is resumed after the break and may lead to problems in starting the machine due to a significant drop of the paper formation quality and tensile strength.

There is therefore a pressing need to reduce the disturbing effects of the breaks on the paper machine as a whole and the Wet End specifically. The majority of paper machines employ no measures to tackle this and in some cases there are manual changes made by the operator during the breaks, only if they observe that the furnish is greatly deviating from its nominal value. The approach proposed in this paper, namely, estimation of the Base Sheet Ash during breaks is novel in the paper industry and represents a large step towards reducing the effects of the disturbance. To make the models robust against bad data and disturbances in the measurements, the LAV minimization was used as opposed the commonly used Least Squares. A very fast optimization algorithm was proposed to overcome the problem of solving the LAV problem online since no efficient solution exists as in the case of LS.

Acknowledgments

This work was partially supported by the UK Leverhulme Trust (F/00 120/BC) and the China National Science Foundation under Grants (60534010, 60828007 and 60974029), 973 National Basic Research Program of China under Grant (2009CB320601) and the 111 project under (B08015).

Appendix A. *f*ADE pseudo code

```
begin
$k, \Gamma^{adapt} \leftarrow 0$
$\alpha, \alpha_u \leftarrow 5$
$\gamma^* \leftarrow \infty$
for $(i = 1, \ldots, n_p)$
   % Initialize the initial population
   $P^{(0)} \leftarrow \mathbf{x}_i^{(0)} = \mathbf{x}^L + \boldsymbol{\rho} \,|_0^1\, (\mathbf{x}^U - \mathbf{x}^L)$
   % Evaluate initial members
   $\gamma_i^{(0)} \leftarrow f(\mathbf{x}_i^{(0)})$
```

```
    if ($\gamma_i^0 < \gamma^*$) do begin
      $\gamma^* \leftarrow \gamma_i^0$
      $\mathbf{x}_{best} \leftarrow \mathbf{x}_i^0$
    end
    % Assign random F to each member
    $F_i^{(0)} \leftarrow \rho \,|_0^1$
  end
  % repeat until reached$k_max$ or$f_tol$
  while (not termination condition) do begin
    for ($i = 1, \ldots, n_p$) do begin
      $\boldsymbol{u}_i^{k+1} \leftarrow \mathbf{x}_{best} + F_i(\mathbf{x}_s^k - \mathbf{x}_t^k + \mathbf{x}_u^k - \mathbf{x}_v^k)$
        ($s = \rho \vdash_0^{n_p}, t = \rho \vdash_0^{n_p}, v = \rho \vdash_0^{n_p}, u = \rho \vdash_0^{n_p}$
        and $\forall k, l, d_k \neq d_l$, where $\boldsymbol{d} = \{s, t, u, v\}$)
      % Select if fitter than parent
      if ($f(\boldsymbol{u}_i^{k+1}) < \gamma_i^k$) do begin
        $\mathbf{x}_i^{k+1} \leftarrow \boldsymbol{u}_i^{k+1}$
        % Update cost vector
        $\gamma_i^{k+1} \leftarrow f(\boldsymbol{u}_i^{k+1})$
      else do begin
        $\mathbf{x}_i^{k+1} \leftarrow \mathbf{x}_i^k$
        $\gamma_i^{k+1} \leftarrow \gamma_i^k$
      end
      % Update best vector and least cost
      if ($\gamma_i^{k+1} < \gamma^*$) do begin
        $\gamma^* \leftarrow \gamma_i^{k+1}$
        $\mathbf{x}_{best} \leftarrow \mathbf{x}_i^{k+1}$
      end
    end
    % Update accumulative adaption vector
    $\Gamma^{adapt} \leftarrow (\Gamma^{adapt} + \Gamma^k - \Gamma^{k+1})$
    % Adaptation loop
    if $k = \alpha_u + \alpha$ do begin
    % Sort vector in decreasing order
    $\Gamma^{str} = \mathrm{sort}(\Gamma^{adapt})$
    % Calculate new performance threshold
    $T_r = \gamma^{str}_{\lfloor n_p/2 \rfloor}$
      % new F for below threshold performance
      for ($i = 1, \ldots, n_p$) do begin
        if $\gamma_i^{adapt} < T_r$ do begin
          $F_i = \rho \,|_{F^L}^{F^U}$
        end
      end
    $\Gamma^{adapt} \leftarrow 0$
    $\alpha_u \leftarrow \alpha_u + \alpha$
    end
  % Remove worst performing member, reduce pop
  if $(\gamma_i^k - \gamma^*)/\gamma_i^k \geq \kappa \&\& \mathrm{n_p} > \mathrm{m}$do begin
    for ($i = 1, \ldots, n_p$) do begin
      if $\gamma_i = max(\Gamma) \&\& \mathrm{n_p} > \mathrm{m}$ do begin
        $\mathbf{x}_i^{k+1} = empty$
        $n_p = n_p - 1$
      end
    end
  end
  $k \leftarrow k + 1$
  end
```

Remark 1. α is the adaptation update interval for the mutation weighting factor. In Nobakhti and Wang (2007) a recommended value of $\alpha = 5$ was determined. F^L and F^U define the range within which F varies. Although this may be changed for each problem, the values of $F^L = 0.1$ and $F^U = 0.9$ has been found to be suitable for most problems. These settings are fixed and are not required to change for different experiments.

References

Abbass, H. A. (2002). The self-adaptive pareto differential evolution algorithm. In *Proceedings of 2002 Congress on evolutionary computation* (pp. 831–836).

Ahola, T. (2005). *Intelligent estimation of web break sensitivity in paper machines.* Technical report C 232, Acta Universitatis Ouluensis.

Armstrong, R. D., & Kung, M. T. (1982). A dual algorithm to solve linear least absolute value problems. *Journal Operational Research Society*, *33*, 931–936.

Babu, B. V., & Angira, R. (2002). A differential evolution approach for global optimization of MINLP problems. In *Proceedings of fourth Asia Pacific conference on simulated evolution and learning*.

Baki, H., Wang, H., Soylemez, M. T., & Munro, N. (2001). Implementing machine-directional basis weight control for a pilot paper machine. *Control Engineering Practice*, *9*, 621–630.

Bonissone, P. P., & Goebel, K. (2002). *When will it break? A hybrid soft computing model to predict time-to-break margins in paper machines.* Technical report 2002GRC233, GE Global Research, Information and Decisions Technologies, Niskayuna.

Brodeur, P. H., Gerhardstein, J. P., Lafond, E. F., Jong, J. H., & Habeger, C. C. (1999). *Monitoring the mechanical behavior of paper during papermaking.* Technical report IPST technical paper series number 807, Institute of Paper Science and Technology, Atlanta, Georgia.

Cardoso, M. F., Salcedo, R. L., Azevedo, S. F., & Barbosa, D. (1997). A simulated annealing approach to the solution of MINLP problems. *Computers and Chemical Engineering*, *21*(12), 1349–1364.

Cerezci, E. T., & Gokpinar, F. (2005). A comparison of three linear programming models for computing least absolute value estimates. *Hacettepe Journal of Mathematics and Statistics*, *34*, 95–102.

Charnes, A., Ferguson, R., & Cooper, W. W. (1955). Optimal estimation of executive compensation by linear programming. *Management Science*, *1*, 138–151.

Charnes, A., Henderson, A., & Cooper, W. W. (1953). *Introduction to linear programming*. New York: Wiley.

Claerbout, J. F., & Muir, F. (1973). Robust modeling with erratic data. *Geophysics*, *38*(5), 826–844.

Claveau, F., Chevrel, K., & Knittel, K. (2008). A 2dof gain-scheduled controller design methodology for a multi-motor web transport system. *Control Engineering Practice*, *16*(5), 609–622.

Daly, D. A. (1965). Factors controlling traction between webs and their carrying rolls. *Tappi Journal*, *48*(9), 88–90.

Ekvall, J., & Hagglund, T. (2008). Improved web break strategy using a new approach for steam pressure control in paper machines. *Control Engineering Practice*, *16*(10), 1151–1160.

Gamperle, R., Muller, S., & Koumoutsakos, P. (2002). A parameter study for differential evolution. In A. Grmela & N. E. Mastorakis (Eds.), Advances in intelligent systems, fuzzy systems, evolutionary computation (pp. 293–298). WSEAS Press.

Gentle, J. E., Narula, S. C., & Sposito, V. A. (1987). Algorithms for unconstrained L_1 linear regression. In Statistical data analysis based on the L_1-norm and related methods (pp. 83–94). Amsterdam: North-Holland.

Gonin, R., & Money, A. H. (1989). *Nonlinear LP-norm estimation*. New York: Marcel Dekker.

Huber, P. J. (1973). Robust regression: Asymptotics, conjectures and Monte Carlo. Ann. Statist.

Ignizo, J. (1995). *Introduction to goal programming*. Newbury Park, CA: Sage.

Lampinen, J., & Zelinka, I. (1999). Mixed variable non-linear optimization by differential evolution. In *Proceedings of Nostradamus*.

Lampinen, J., & Zelinka, I. (2000). On stagnation of the differential evolution algorithm. In *Proceedings of MENDEL.*

Lampinen, J. L. (2002). Adaptive parameter control of differential evolution. In R. Matoušek & P. Ošmera (Eds.), Proceedings of Mendel 2002, eighth international conference on soft computing (pp. 19–26).

Li, H. L. (1998). Solve least absolute value regression problems using modified goal programming techniques. *Computers & Operations Research*, *25*(12), 1137–1143.

Lin, Y. C., Hwang, K. S., & Wang, F. S. (2001). Co-evolutionary hybrid differential evolution for mixed-integer optimization problems. *Engineering Optimization*, 1–20.

Mehrotra, S. (1992). On the implementation of a primal–dual interior point method. *SIAM Journal on Optimization*, *2*, 575–601.

Narula, S. C., & Wellington, J. F. (1982). The minimum sum of absolute errors regression: A state of the art survey. *International Statistical Review*, *50*, 317–326.

Nobakhti, A., & Wang, H. (2007). A simple self-adaptive differential evolution algorithm with application on the ALSTOM gasifier. *Applied Soft Computing*, *8*(1), 350–370.

Rice, J. R., & White, J. S. (1964). Norms for smoothing and estimation. *SIAM Review*, *6*, 243–256.

Smook, G. A. (1992). *Handbook for pulp and paper technologists*. Toronto: Angus Wilde Publications.

Storn, R., & Price, K. (1997). A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*, 341–359.

Walter, J. C. (1993). *The coating process*. Vancouver: Tappi Press.

Wang, H., Logghe, D., & Miskin, D. (2005). Physical modelling and control of lateral web position for wallpaper making processes. *Control Engineering Practice*, *13*, 401–412.

Wang, X. G., Dumont, G. A., & Davies, M. S. (1993). Estimation in paper machine control. IEEE Control Systems Magazine, 34–43.

Zaharie, D. (2003). Control of population diversity and adaptation in differential evolution algorithms. In R. Matoušek & P. Ošmera (Eds.), Proceedings of Mendel 2003, ninth international conference on soft computing, Brno (pp. 41–46).