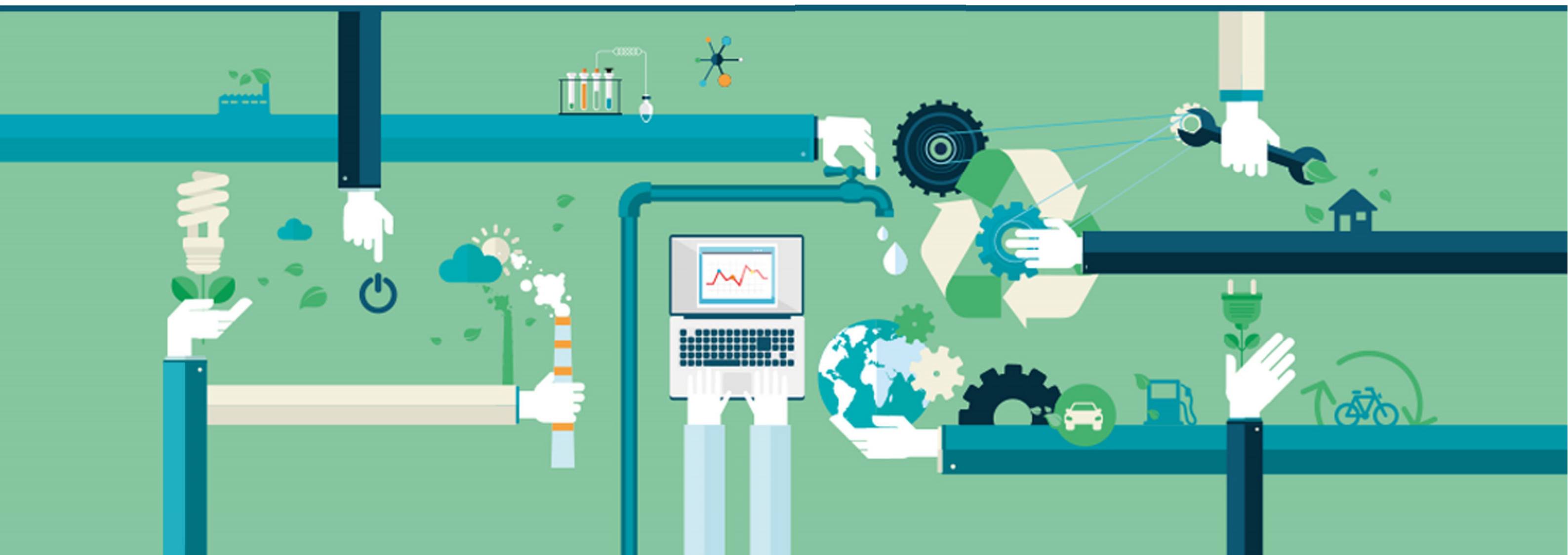
COORDINATED PRODUCTION
FOR BETTER RESOURCE EFFICIENCY

D2.1 – REPORT ON DYNAMIC DATA RECONCILIATION OF LARGE-SCALE PROCESSES

José Luis Pitarch[a]
César de Prada[b]
[a] *Research associate (UVA) – Spain*
[b] *Professor (UVA) – Spain*

October 2018
www.spire2030.eu/copro



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723575

Project Details

PROJECT TITLE	Improved energy and resource efficiency by better coordination of production in the process industries
PROJECT ACRONYM	**COPRO**
GRANT AGREEMENT NO	**723575**
INSTRUMENT	**RESEARCH AND INNOVATION ACTION**
CALL	**H2020-SPIRE-02-2016**
STARTING DATE OF PROJECT	**NOVEMBER, 1ST 2016**
PROJECT DURATION	**42 MONTHS**
PROJECT COORDINATOR (ORGANIZATION)	**PROF. SEBASTIAN ENGELL (TUDO)**

THE COPRO PROJECT

The goal of CoPro is to develop and to demonstrate methods and tools for process monitoring and optimal dynamic planning, scheduling and control of plants, industrial sites and clusters under dynamic market conditions. CoPro pays special attention to the role of operators and managers in plant-wide control solutions and to the deployment of advanced solutions in industrial sites with a heterogeneous IT environment. As the effort required for the development and maintenance of accurate plant models is the bottleneck for the development and long-term operation of advanced control and scheduling solutions, CoPro will develop methods for efficient modelling and for model quality monitoring and model adaption.

The CoPro Consortium

Participant No	Participant organisation name	Country	Organisation
1 (Coordinator)	Technische Universität Dortmund (TUDO)	DE	HES
2	INEOS Köln GmbH (INEOS)	DE	IND
3	Covestro Deutschland AG (COV)	DE	IND
4	Procter & Gamble Services Company NV (P&G)	BE	IND
5	Lenzing Aktiengesellschaft (LENZING)	AU	IND
6	Frinsa del Noroeste S.A. (Frinsa)	ES	IND
7	Universidad de Valladolid (UVA)	ES	HES
8	École Polytechnique Féderale de Lausanne (EPFL)	CH	HES
9	Ethniko Kentro Erevnas Kai Technologikis Anaptyxis (CERTH)	GR	RES
10	IIM-CSIC (CSIC)	ES	RES
11	LeiKon GmbH (LEIKON)	DE	SME
12	Process Systems Enterprise LTD (PSE)	UK	SME
13	Divis Intelligent Solutions GmbH (divis)	DE	SME
14	Argent & Waugh Ltd. (Sabisu)	UK	SME
15	ASM Soft S.L (ASM)	ES	SME
16	ORSOFT GmbH (ORS)	DE	SME
17	Inno TSD (inno)	FR	SME

Document details

Deliverable Type	**Report**
Deliverable No	**2.1**
Deliverable Title	Report on dynamic data reconciliation of large scale processes
Name of Lead Partner for this Deliverable	**Universidad de Valladolid**
Version	**2**
Contractual Delivery Date	**31 October 2018**
Actual Delivery Date	**31 October 2018**

Dissemination level		
PU	Public	**X**
CO	Confidential, only for members of the consortium (including the Commission)	

Abstract

Availability of reliable process information in real time is key in any decision-making procedure. Thus, good industrial decision-support implementations require dealing with gross errors and consideration of process transients in order to get a set of measurements which will be coherent with the basic underlying process dynamics. This report presents dynamic data reconciliation methods and tools adapted to the requirements of industrial environments (large-scale systems and noisy/faulty data). Moreover, basic concepts in literature are extended to artificially increase system redundancy as well as to cope with time-varying parameter estimation. The procedure summarized in this report has been tested in the Lenzing case study.

Revision History

The following table describes the main changes done in the document since it was created.

Revision	Date	Description	Author (Organisation)
V0	14/09/2018	Document creation	J.L. Pitarch (UVA)
V1	24/09/2018	Internal review	C. de Prada (UVA)
V2	04/10/2018	External review	A. Santecchia (EPFL)
V2	31/10/2018	Final approval	S. Engell (TUDO)

Disclaimer

Table of contents

1 Executive summary

The CoPro partner UVA is responsible of Task 2.1 – Data reconciliation techniques – with the participation of partners LENZING, EPFL, CSIC, TUDO, INEOS and P&G. The objectives of this task can be summarized in two:

- Development of indicators to elucidate measurement errors and to suggest corrective actions for those that are systematic.
- Development of robust data reconciliation algorithms which are suitable for large-scale continuous plants, with particular demonstration in the project case studies.

In particular, this deliverable focuses on data reconciliation in dynamic situations, and we have considered an evaporation plant in LENZING as proof of concept. Although these plants attempt to operate normally around some desired points, the variability provided by the external factors such as the product income (variable temperatures, flows and concentrations) or the cooling system performance (affected by the weather), makes the control system to change setpoints in order to adapt the plant to each situation while fulfilling the desired evaporation demands. This translates in a non-negligible time percentage where the plant is not in steady state, a common fact in many large-scale systems in the process industry.

Task 2.1 extends from month 7 to 24 and, during this period, the partners UVA, EPFL and LENZING mainly were the ones conducting the work on dynamic data reconciliation. The chronogram of the task execution is as follows:

1. Literature review on dynamic data reconciliation methods and proposal of adaptations/extensions to make them suitable for the applicability to large-scale processes.
2. In parallel with the previous, selection of the more suitable case study to serve as proof of concept. Development of a first-principles model to be the backbone for further reconciliation algorithms.
3. Scheduling and execution of experimental tests onsite in order to collect data from the plant in different transient states.
4. Testing the more promising data reconciliation methods with the provided plant dataset.

The main criteria used to select and to extend the dynamic data reconciliation methods were: a) the computational efficiency, i.e., the ability to process hundreds of variables in acceptable time for online implementations; b) the robustness against sensor noise and gross errors and; c) the ability to perform consistent estimation of time-varying variables and parameters, especially for slow varying dynamics that are either not considered or difficult to model. Indeed, this builds the bridge between Task 2.1 with tasks 2.3 – Model-based soft sensing – and 2.4 – Monitoring of equipment degradation–.

2 Introduction

Decision-support systems require information about process performance, preferable in real time, normally in the form of some *efficiency indicators* [1] to be computed from process measurements. However, all measurements are subject to errors (sensor calibration, noise, out of range situations, etc.). Therefore, in large-scale systems, redundant sensoring are normally implemented either via hardware (duplicated measurements) or software (soft sensors [1]). Based on this last concept of redundancy, data-reconciliation algorithms aim to provide a set of process-variables estimates close to the sensor values, but coherent with the process dynamics : fulfilling basic first-principle laws such as mass and energy balances [1].

This report focuses on dynamic data reconciliation (DDR), which is solving an optimization problem where the process (dynamic) equations act as constraints to be satisfied within a certain time interval, and all process variables (input, output or parameter) are actually decision variables for the optimization algorithm. As a result, the optimization setup is generally large, with many nonlinear constraints from the process model. Therefore, the inclusion of dynamic models requires a careful balance between the added complexity and the required computation times for solving the associated (dynamic) optimization problems. A common trade-off solution is making use of models that combine a detailed stationary process constraints with additional simplified dynamics.

In case that the measurement errors are normally distributed around their true values, the DDR approach is able to provide the best set of estimations coherent with the model. Nevertheless, due to several reasons such as serious defects in instruments or in the communication network, the solution provided by the data reconciliation is distorted. As a consequence, the error is spread throughout the rest of the variables, creating a smearing effect. These problems are called gross errors and their detection and treatment is crucial for obtaining good estimations. The propagation of gross errors in measurements to the efficiency indicators must be avoided because, otherwise, the decision-support systems will recommend wrong actions. Hence, previous data treatment introducing gross-error detection plus the use of robust estimators in the data reconciliation are also mandatory. In this way, this step avoids the inclusion of corrupted data (outliers) in further decision support phases and serves as a detector of systematic errors in sensors/process.

This report presents an enhanced DDR methodology which consists on a first step of data treatment to exclude outliers and detection of transient measurements. After this step, reliable data is assumed to be available to perform DDR itself. So, the second step is using reconciliation for experimental identification of the model's grey part : time-varying parameters and experimental patterns. Once the model has been identified, the third step is validation with new transient data. Last, the proposed methodology is tested in a particular evaporation plant in Lenzing AG, where a grey-box nonlinear model is validated through eight months of operation.

3 Detection of transient and steady-state data

Identification of both steady state and transient state in noisy process signals is important either in model identification and execution of real-time DDR routines. On the one hand, dynamic models have coefficients representing time-constants, which should only be adjusted to fit data from transient conditions. Therefore, detection of transients triggers the collection of data for dynamic modelling. On the other hand, static constraints do not represent transients, so they should only involve variables whose dynamics is negligible with respect to the dominant one. Note that, although a pure stationary model could cope with the data reconciliation task, chemical processes are inherently nonstationary, so some model parameters would need to be adjusted periodically to keep the models true to the process and functionally useful.

Since process variables are usually noisy, DDR needs to "see" through the noise and announce probable steady states or probable transient situations. Hence, the employed method needs to consider an appropriate time horizon, longer than the most recent pair of samples, in order to observe a local trend to confidently make any statement. So, some straightforward implementations for steady-state/transient detection would be statistical tests of the slope of a linear trend (computed by linear regression) in the time series of a moving horizon data window: if the process is in steady state, the slope will fluctuate near zero values.

3.1 Concept

The method recommended here is based on the fact that the variance of a signal measures the deviation from the mean value and should be constant in a stationary process. In a transient, the moving average of the signal will lag behind the change in the signal and the variance will increase. The method then uses the R-statistic, a ratio between two variances, measured on the same set of data by two approaches [4]. The idea is to take a transient condition which is barely detectable or decidedly inconsequential (per human judgment) and set the probably steady-state threshold for the R-statistic as an improbable low value, but not so low as to be improbably encountered when the process is truly at steady state.

The concept will be illustrated through Figure 1 taken from [5] where a set of data is represented over time by green dots. The method first calculates a filtered value of the process measurements, a moving average, indicated by the red curved line that lags behind the data. Then the variance in the data is measured by two methods. First, a moving variance is computed over the moving average and then the variance of the stationary process (obtained by differentiation of the data) is computed to normalize the first one.

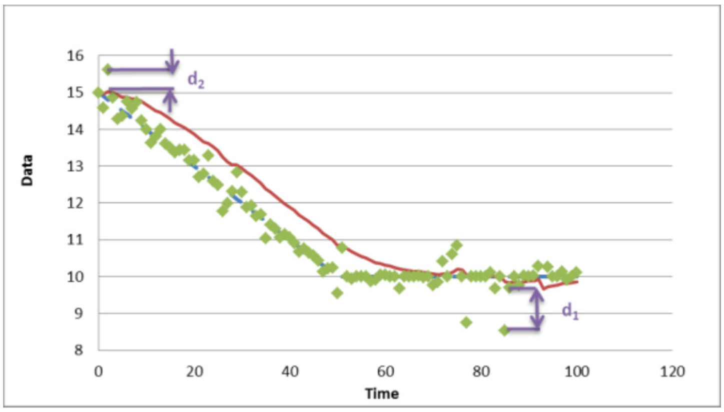


Figure 1: Noisy measurements (green diamonds), filtered data (solid red line) and deviations (purple arrows).

If the process is at steady state, then the filtered value of the measurement X_f will coincide with the average of the data. Then, a process variance ${\sigma_{d_2}}^2$ estimated with the moving average X_f will be ideally equal to ${\sigma_{d_1}}^2$ estimated for the stationary process. Thus, the ratio of the variances $r = \frac{{\sigma_{d_2}}^2}{{\sigma_{d_1}}^2} \cong 1$. Alternatively, if the process is in a transient, the filtered value X_f lags behind the process data and the variance as measured by ${\sigma_{d_2}}^2$ will be much larger than the one estimated by d_1, so $r = \frac{{\sigma_{d_2}}^2}{{\sigma_{d_1}}^2} \gg 1$.

The filtered value which provides an estimate of the data mean is computed by

$$X_f(k) = \lambda_1 X(k) + (1 - \lambda_1) \cdot X_f(k-1) \tag{1}$$

where $X(k)$ is the process variable at time sample k and λ_1 is a first-order filter factor. Similarly, the method to measure the variance ${\sigma_{d_2}}^2$ is computed by ν as:

$$\nu^2(k) = \lambda_2 \left(X(k) - X_f(k-1)\right)^2 + (1 - \lambda_2) \cdot \nu^2(k-1) \tag{2}$$

The previous value of the filtered measurement is used instead of the most recently updated value to prevent autocorrelation from biasing the variance estimate, keeping the equation for the ratio simple. In contrast, the variance ${\sigma_{d_1}}^2$ is estimated by δ using another filter based on sequential data differences as a way to convert a possible non-stationary process in a stationary one:

$$\delta^2(k) = \lambda_3 \big(X(k) - X(k-1)\big)^2 + (1 - \lambda_3) \cdot \delta^2(k-1) \tag{3}$$

The ratio of variances, the R-statistic to be compared to its critical values, may now be computed by the following simple equation

$$R = \frac{(2 - \lambda_1) \cdot \nu^2(k)}{\delta^2(k)} \tag{4}$$

as it follows a F distribution. Note that the coefficient $(2 - \lambda_1)$ in (4) is required to scale the ratio to represent the actual variance ratio [6]. Recommended values for the weighting factors are $\lambda_1 = 0.2$, $\lambda_2 = \lambda_3 = 0.1$ [7], which effectively mean that the most recent 45 data points are used to calculate the *R*-statistic.

Critical values for R are selected by the level of significance α, alternately the confidence level $1 - \alpha$, that the end user desires to achieve. If the computed *R*-statistic is greater than *R*-critical, then there is a $100(1-\alpha)$ percent confidence that the process is not at steady state. Consequently, a value of R less than or equal to R-critical means the process may be at steady state.

3.1 Proposed procedure

The above concept can be implemented by an algorithm that combines steady-state and transient identification to prevent immediate sequencing of computations if the system passes through a time where it is probably not at steady state. After the dynamic behavior is detected, the algorithm will allow the next set of conditions to begin after the process returns to steady state. In addition, a time limit for any one run in the experimental procedure can be explicitly included in order to identify any occurrence of either 1) a change was made and not detected or 2) the change made the process so unstable that steady state could not be obtained. Figure 2 illustrates the logic used for the automatic algorithm [7], which works as follows.

First, process data is obtained and the R-statistic value is computed. If this value is larger than its upper critical value, the algorithm determines that the process is probably not in steady state (path Y1), so the transient variable TS is set to 1. This is followed by a check to determine whether or not the time limit has been exceeded. If not (path N7), the algorithm wait for the next sample to get new data. If the time limit has been exceeded (path Y8), then the next run is implemented, the point of change (POC) time is recorded to analyse the recorded set of data, and the next sampling is observed.

If not in a transient, the algorithm checks whether the process is definitively at steady state (path Y4) or not (which means in an indeterminate state) by comparing with the lower critical value for the R-statistic. If indeterminate (path N3), and the time limit has not been exceeded (path N7), the algorithm wait for the next sample to get new data. If the time limit has been exceeded, (path Y8), the next run is implemented, the point of change (POC) time is recorded and the next sampling is observed.

If steady state has been detected (path Y4) and TS=1 (formerly the process was in a transient state, path Y5), TS is set to zero, and the next run is implemented. However, if TS was 0 (path N6) the process has not been in a TS, which means that the recent implementation of new conditions has not taken effect yet. In this case, if the time limit has not been exceeded (path N7), the next sampling is observed and new data is analyzed. If the time limit has been exceeded, (path Y8), the next run is implemented, the point of change time is recorded to analyze the recorded set of data, and the next sampling is observed.

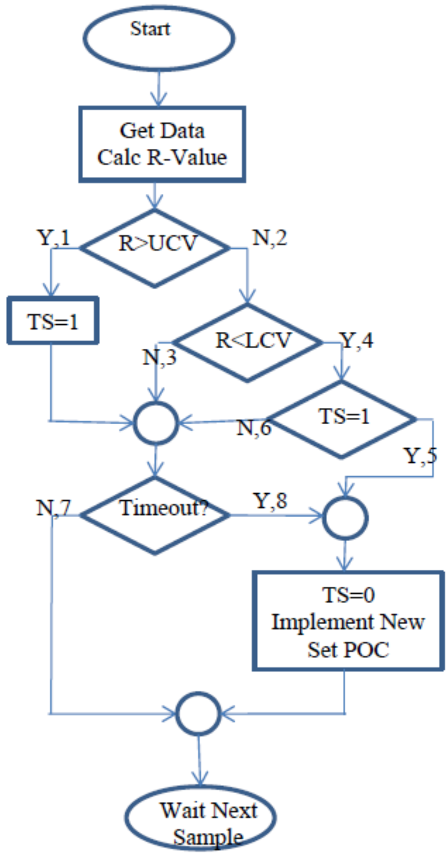


Figure 2: Algorithm for steady-state and transient identification.

4 Dynamic data reconciliation

The usual way of setting a data reconciliation problem, either dynamic or static, is to formulate a nonlinear optimization problem minimizing a cost function J, which is a weighted sum of the deviations between the measured data and their corresponding variables in the model, satisfying the nonlinear model equations plus other possible constraints. Depending on the type of process (slow or fast dynamics) and the operation mode (variation rate of the control set points), the dynamic data reconciliation can be implemented with a process model of more or less dynamic detail, i.e., replacing the fast, usually unobserved, dynamics by algebraic stationary constraints.

4.1 Problem formulation

Differently from classical reconciliation, in this case the data to be reconciled is not only a sample corresponding to a time instant, but a set of samples corresponding to a time window H, long enough to be able to capture the process dynamic behavior.

The general formulation can be expressed as follows:

$$\underset{x,\hat{u},\hat{p}}{\text{minimize}} \quad J = \sum_{i=1}^{s} \sum_{t=t_c-H}^{t_c} \frac{\gamma_i}{\sigma_i^2} \cdot \left(\hat{\theta}_i(t) - \theta_i(t)\right)^2 \tag{5}$$

$$\text{subject to:} \quad \frac{dx_i}{dt} - \Psi_i(x,\hat{u},\hat{p}) = 0 \quad i: 1, \dots, n \tag{6}$$

$$h_j(x,\hat{u},\hat{p}) = 0 \;\; j: 1, \dots, r; \qquad g_k(x,\hat{u}) \geq 0 \qquad k: 1, \dots, m \tag{7}$$

$$\hat{\theta} = C \cdot [x,\hat{u}]^T; \; U_{min} \leq \hat{u} \leq U_{max}\, ; P_{min} \leq \hat{p} \leq P_{max} \tag{8}$$

Where $x, \hat{u}, \hat{p}$ are decision variables (actually all model variables: states, inputs and parameters, see Figure 3), $\theta \in \mathbb{R}^s$ is the vector of all sensor measurements ($\hat{\theta}$ their corresponding estimates, i.e. variables belonging to the model), t_c is the current time instant, H is the length of the time window considered for optimization and Ψ are nonlinear functions representing the process dynamic constraints. All the terms of the cost function (5) have been normalized using the variability σ^2 of the measured variables and a user-defined weighting factor γ which represents, for instance, the user's confidence in the instruments.

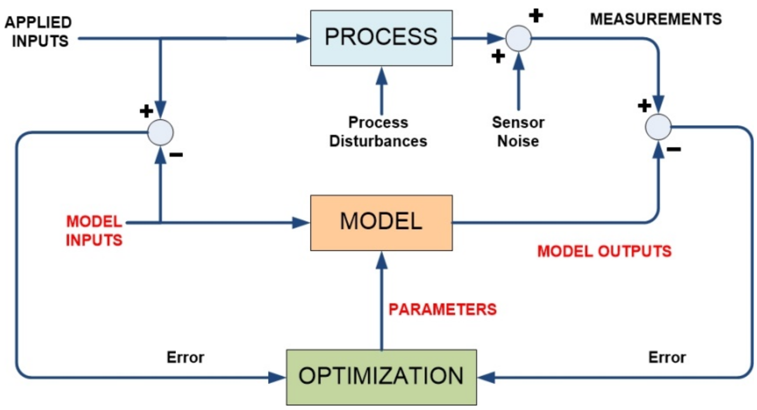


Figure 3: General data reconciliation scheme.

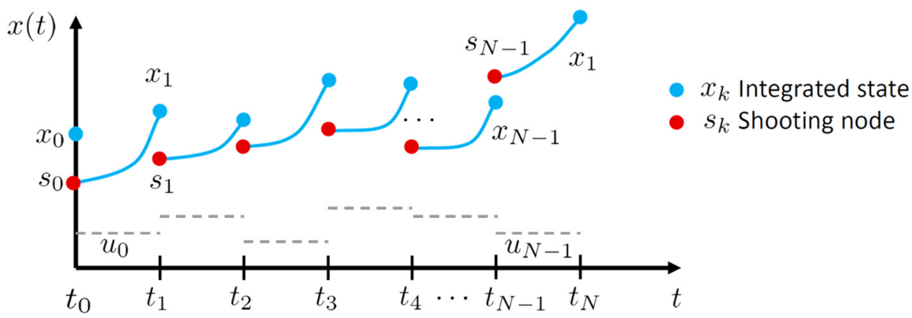
In contrast to the stationary data reconciliation, the above statement is a large dynamic optimization problem, because of the presence of differential equations (6) together with the need of considering many past samples to comply with such dynamics. Therefore, some strategies are usually adopted to facilitate the solution of the general problem: time discretization, input-deviation penalty, moving horizon data window [3], and an adaptive noise model [8].

4.1.1 Time discretization

In order to get an NLP from the above problem, the dynamic problem needs to be parameterized. An option is to solve the optimization iteratively by a sequential approach, where the independent degrees of freedom of (6) are discretized. Then an optimization problem with respect to these decision variables is setup, connected to a dynamic simulator where the differential equation set is solved numerically over the time window in order to compute the cost function and constraints. The so-called single-shooting approaches belong to this type. An alternative is using a simultaneous approach, where the states trajectories are also split in discrete intervals whose starting point (initial guess) is decision variable for the optimizer. In this way, multiple integrations are run simultaneously and the task of linking the end of each interval with the initial point of the next one is left for the optimizer [9]. Figure 4 below gives an overview of this approach.

Basic idea:

- Divide the horizon into N control stages
- In each subinterval $[t_k, t_{k+1}]$ parameterize the input $u(\tau)$ for $\tau \in [t_k, t_{k+1}]$
- Integrate the system from each **shooting node** s_k using the input u_k



- Integrate from each shooting node: $x_{k+1} = s_k + \int_{t_k}^{t_{k+1}} f(s(\tau), u_k)\, d\tau,$
- Add continuity constraints: $x_k = s_k$

Figure 4: Simultaneous multiple-shooting approach.

Furthermore, a pure simultaneous approach without involving successive integrator calls is also possible. In this case, the whole dynamics is directly discretized by Taylor-series approximation or orthogonal collocation methods so that (8) becomes an extended set of pure algebraic constraints. Then, we just need to solve a one-step large NLP optimization problem [10]. This last option is usually the more convenient for large-scale systems where real-time computation capabilities are preferred over simulation accuracy.

4.1.2 Input-deviation penalty

The formulation (5)-(8) explicitly parameterizes the input variables $u(t) \in \mathbb{R}^c$ along time, resulting in a large number of decision variables $u \in \mathbb{R}^{c \times H}$. In order to reduce the problem size, thinking in real-time DDR applications, it can be assumed that input measurements do not suffer of random errors but only of a possible systematic deviation Δu. Hence, instead of reconciling $u \in \mathbb{R}^{c \times H}$ variables, we significantly reduce the set to $\Delta u \in \mathbb{R}^c$ at the price of reducing the degrees of freedom. Thus, the DDR problem (5)-(8) reduces to:

$$\underset{x, \Delta u, \hat{p}}{\text{minimize}} \quad J = \sum_{i=1}^{s-c} \sum_{t=t_c-H}^{t_c} \frac{\gamma_{1i}}{\sigma_{1i}^2} \cdot \left(\hat{y}_i(t) - y_i(t)\right)^2 + \sum_{j=1}^{c} \frac{\gamma_{2i}}{\sigma_{2i}^2} \cdot \Delta u_i^2 \tag{9}$$

$$\text{subject to:} \quad \frac{dx_i}{dt} - \Psi_i(x, \hat{u}, \hat{p}) = 0 \quad i: 1, \dots, n; \qquad \hat{u} = u + \Delta u; \tag{10}$$

$$h_j(x, \hat{u}, \hat{p}) = 0 \quad j: 1, \dots, r; \qquad g_k(x, \hat{u}) \geq 0 \quad k: 1, \dots, m; \tag{11}$$

$$\hat{y} = C \cdot [x, \hat{u}]^T; \quad \Delta U_{min} \leq \Delta u \leq \Delta U_{max}; \quad P_{min} \leq \hat{p} \leq P_{max} \tag{12}$$

4.1.3 Moving-horizon window

A moving time window approach is useful to decrease the size of the discretized optimization problem. Therefore, it is important to choose an appropriate horizon length H: considering a large historical is very computationally demanding, so real-time constraints may not be satisfied, but if H is too small, the information available may not be enough to capture the dynamics, therefore to perform a sensible reconciliation.

The algorithm for moving-horizon DDR can be summarized in the following steps [11]:

1. Acquire process measurements at current time $t = t_c$
2. Minimize (5), under the constraints (6)-(7) over the time window $(t_c - (H-1)T_s \leq t \leq t_c)$, being $T_s = t_k - t_{k-1}$ the data sampling time.
3. Save x at time t_c as the reconciled signal for online control purposes
4. Repeat at the next time sample, $t_c + T_s$

One advantage of the moving window approach is that the only tuning parameter is the size of the history horizon H, if the weights γ_i in (5) are equal.

At this point is where DDR shares several features with other online estimation techniques such as moving-horizon estimation (also involving model-based optimization) [12] or augmented Kalman filters [13].

4.1.4 Adaptive noise model

Systems with an unknown noise model, i.e., unknown or varying σ_i, can also be addressed using the moving window approach; however the window should be substantially longer to be able to estimate the noise model as the sample standard deviation $\hat{\sigma}$ over the window. In order for this estimate to be statistically significant, this window should contain at least 50 to 100 points and the *true* (noise-free) signals must vary slowly over that window. If a sufficiently large number of points is used, then the random variable $\hat{\sigma}$ is approximately Gaussian with variance inversely proportional to the number of points. For a more thorough discussion of the statistics of $\hat{\sigma}$ refer to [14].

4.2 Enhanced formulation

The main obstacle of data reconciliation in industrial applications is the scarce number of available sensors that can provide an acceptable level of redundancy in the recorded data. This lack of redundancy leads to a model that is only able to calculate the system unknowns, using perhaps wrong measurements that cause the generation of erroneous solutions. In order to palliate this issue, two lines of action are explored: 1) artificially increasing the system redundancy and 2) mitigating the influence of gross errors in sensor measurements. The second one is later treated in Section 4.3, whereas an *enhanced DDR formulation* to increase redundancy is presented next.

The main idea behind this method is to increase the system redundancy by adding a set of "virtual" measurements $\theta_\pi \in \mathbb{R}^v$ that are not directly sampled by sensors but that are initially guessed and later reconciled, since they are treated as normal measurements [15]. These virtual quantities θ_π can be either slow-varying variables or constant parameters and their previous reconciled values $\hat{\theta}_{\pi i}$ are "re-injected" into the DDR problem at each time instant together with the corresponding a posteriori variance σ_π^2 [16]. Following this idea, the enhanced DDR problem from (5)-(8) can be mathematically expressed as follows:

$$\underset{x,\hat{u},\hat{p}}{\text{minimize}} \quad J = \sum_{t=t_c-H}^{t_c} \left(\sum_{i=1}^{s} \frac{\gamma_{1i}}{\sigma_i^2} \cdot \left(\hat{\theta}_i(t) - \theta_i(t)\right)^2 + \sum_{j=1}^{v} \frac{\gamma_{2i}}{\sigma_{\pi i}^2} \left(\hat{\theta}_{\pi i}(t) - \hat{\theta}_{\pi i}(t-1)\right)^2 \right) \tag{13}$$

$$\text{subject to:} \quad \frac{dx_i}{dt} - \Psi_i(x,\hat{u},\hat{p}) = 0 \quad i: 1,\dots,n \tag{14}$$

$$h_j(x,\hat{u},\hat{p}) = 0 \;\; j: 1,\dots,r; \qquad g_k(x,\hat{u},\hat{p}) \geq 0 \;\; k: 1,\dots,m \tag{15}$$

$$\hat{\theta}(t) = C \cdot [x,\hat{u}]^T;\; U_{min} \leq \hat{u} \leq U_{max}\,;\, P_{min} \leq \hat{p} \leq P_{max} \tag{16}$$

With the assumption that the true values of the re-injected virtual variables remain almost constant between consecutive samples. Under this assumption, the following relations are employed to update these variables for the next DDR execution:

$$\hat{\theta}_{\pi i}(t_c - H) \coloneqq \hat{\theta}_{\pi i}(t_c - H - 1); \quad \sigma_{\pi i}^2 \coloneqq \sum_{j=1}^{s+v} \frac{{S_{ij}}^2}{\sigma_i^2} \tag{17}$$

where S is the sensitivity matrix[2]. Figure 3 below summarizes the enhanced DDR procedure.

As in standard DDR, first principle models and extensive data sets are required. However, the virtual measurements carry information from the past time steps so, thanks to their re-injection, there is no need for fitting slow-varying variables (capturing long-term dynamics) over a large time window H (huge dataset). In this way, the size of the problem is greatly decreased, making these calculations easier for on-line applications.

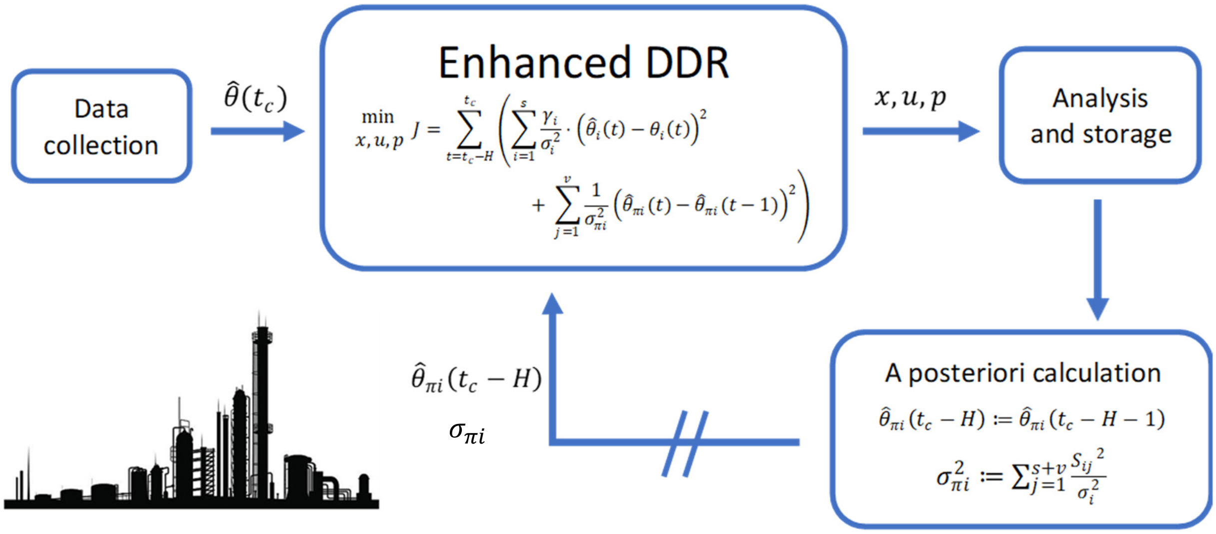


Figure 5: Flow chart for the enhanced DDR execution.

4.3 Treatment of gross errors

One of the issues of the data reconciliation based on a classical least-squares (LS) objective function is that the approach assumes that sensor measurements are affected by Gaussian noise with zero mean. However, this LS function is very affected by gross errors (i.e., systematic deviations from the expected coherent values) so the algorithm tries to spread these gross errors among all variables, thus distorting the estimations for the correct measurements. There are several approaches for dealing with gross errors in the literature, but only a few of them are practical for online DDR of industrial use. The policy summarized below intends to mitigate such gross-error effect by using *robust estimators* (also called *M-estimators*) [17] that are less sensitive to deviations from ideal Gaussian distributions.

The basic idea is to look at the bulk of measurements and limit the weight of variables affected by gross errors [18]. In contrast to classical weighted least-squares (5) or (13), which give quadratic importance to the errors, the robust estimators limit their contribution to the cost function when the error is large. In this way, the effect of the gross error is attenuated and the optimizer does not try to distribute the error among all the variables in order to avoid this high quadratic penalty.

There are some suitable robust estimators proposed in the literature like, for instance:

$$\text{Welsch} = \frac{c^2}{2}\left(1 - e^{-\left(\frac{r_i}{c}\right)^2}\right) \tag{18}$$

$$\text{Fair} = c^2\left(\frac{|r_i|}{c} - \log\left(1 + \frac{|r_i|}{c}\right)\right) \tag{19}$$

Where r_i stands for the estimation error and c is a user-defined parameter to tune the slope of the function. Among them, the *Fair estimator* has been chosen as the more suitable for practical large-scale implementations, because it is continuous (required for gradient-based nonlinear programming algorithms) and it gives a good tradeoff between complexity and performance (see Figure 6) [19].

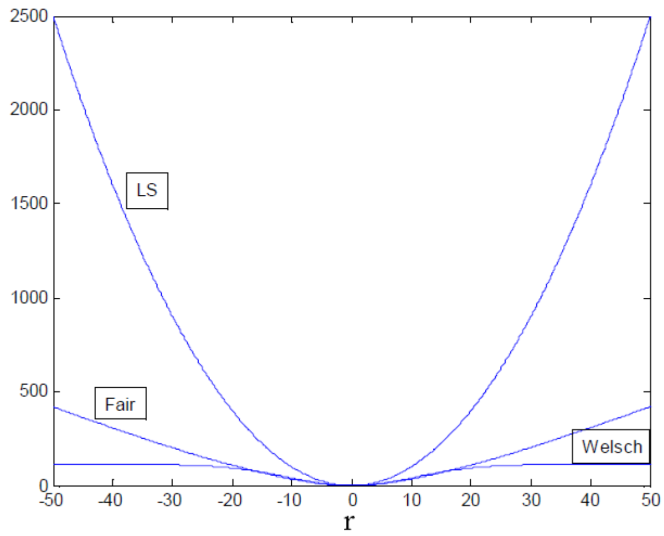


Figure 6: Least squares and M-functions represented for a range of values of the measurement error r_i.

Note that, in this approach, the gross errors are not detected and eliminated; instead, they remain in the data set but their effect on the solution is reduced, thus avoiding the propagation of the error to other instruments. There exist other iterative approaches which attempt to concentrate gross errors in the faulty sensors, eliminating those sensors from the data set afterwards (e.g. PCA-based tests) and then performing reconciliation again [20]. However, note that iterative procedures processing important datasets are computationally expensive to be executed online. As DDR in this report focuses on real-time aspects using a moving-horizon approach, these other iterative-elimination procedures are intentionally left out of the scope.

5 Case study: Multiple-effect evaporation plant

One of the biggest evaporation plants at Lenzing AG is chosen as proof of concept for the DDR methods summarized in this report. This plant gives service attached to the fiber-production process and its goal is to regenerate an acid flow coming from the spinning process, where fibers take form from the previously processed cellulose pulp.

5.1 System description & modelling

The Lenzing evaporation plant considered for this report is basically formed by several evaporation chambers and heat exchangers arranged in serial connection, a steam condenser and a cooling system. Figure 7 depicts a simplified scheme of the line, in which some single equipment (evaporation chambers and exchangers) have been lumped due to lack of measurements in between them.

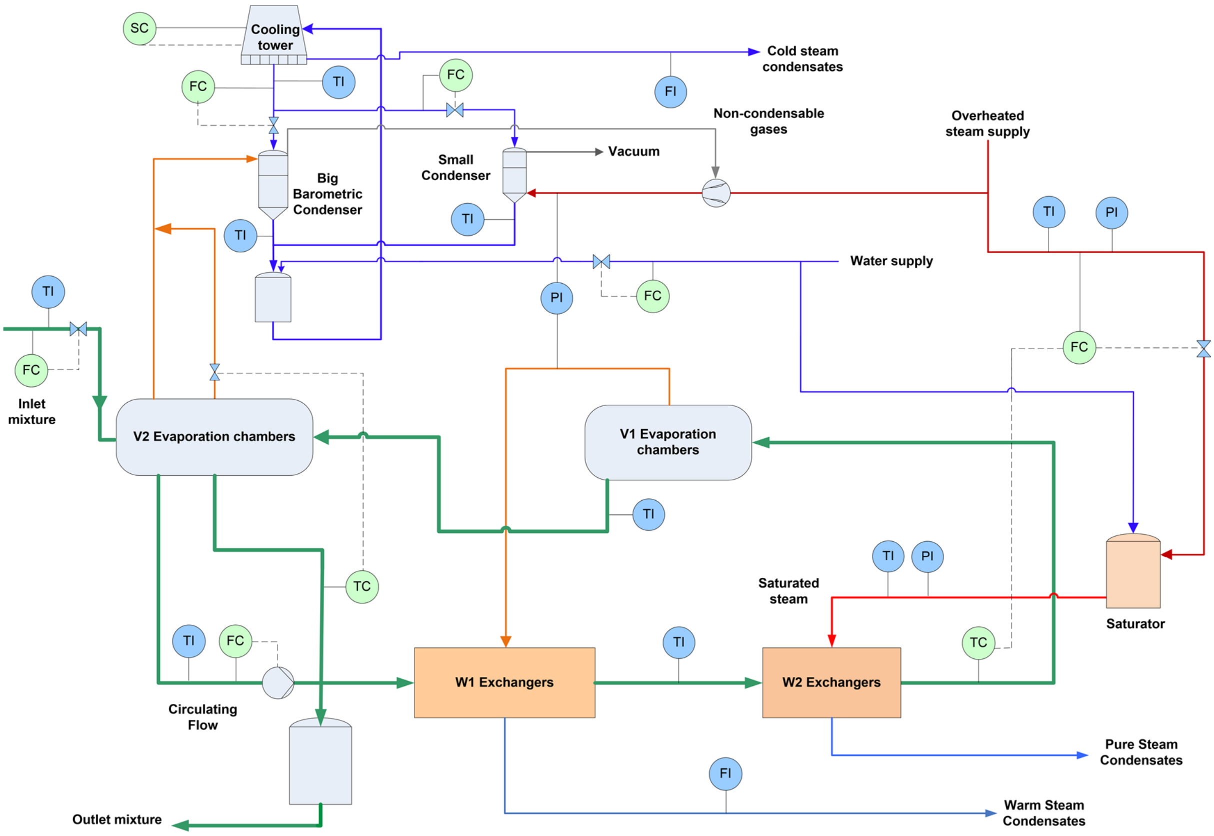


Figure 7: Simplified diagram of the evaporation plant with location of existent instrumentation: transducers (blue) and controllers (green).

The system works as a multiple-effect evaporation, achieved on the one hand thanks to the pressure drop in the chambers V_2 created by the condenser, and in the other hand to vacuum pumps connected to the evaporation chambers labelled as V_1. The evaporation plant, when connected to the main process, receives an input liquid mixture of water with acid and other chemical components plus residual of organic material. The goal is to concentrate the solution by removing certain amount of water. To achieve this, the acid bath goes through the line of heat exchangers W_1, W_2 in counter current with saturated-steam flows (some coming from the evaporators V_1 and other from a fresh steam generated in a boiler) to increase its temperature. Then, the hot mixture enters sequentially to the low-pressure chambers V_1, which forces a partial evaporation of water. Afterwards, an additional evaporation phase is performed in the last set of chambers V_2 thanks to the condenser, which sucks out steam by condensing it with cold water from a cooling tower. Finally, part of the concentrated liquid leaves the process and the rest mixes with the input, being recirculated through the process.

A nonlinear steady-state model of this system (whose core part is based in first principles) was previously developed for real-time optimization purposes [21]. The model equations are omitted for brevity (see the above reference for details) but can be summarized as follows :

- Equations of energy and mass balances taken in the evaporators V_1, V_2, heat exchangers W_1, W_2, steam condensers, steam saturator and in the cooling tower.
- Density relationships between mass and volumetric flows of the liquid mixture, water and steam as a function of temperature and/or pressure.
- Heat transmission between fluids in the exchangers: $Q = UA \times LMTD$, where Q is the transmitted heat, UA is the heat-transmission coefficient (to be estimated), and the logarithmic mean temperature difference ($LMTD$) has been computed using the Chen's approximation [22].
- Phase equilibriums in the evaporation chambers as a function of temperatures, pressure and concentrations.
- Psychometric conditions in the cooling tower and experimentally obtained cooling performance depending on the temperature difference of the cool water with the ambient.
- Relationship between the airflow through the cooling tower with the fan speed, including the experimentally identified convection effect due to in-out temperature difference.

In order to perform DDR, approximate first-order dynamics are added to the energy balances in the evaporation chambers, steam condensers and the cooling tower, trying to represent the energy accumulation due to the fluids residence time in such equipment :

$$m \cdot c_e \frac{dT}{dt} = \sum_i F_i \cdot H(T_i, C_i, P_i) - \sum_o F_o \cdot H(T_o, C_o, P_o) \tag{20}$$

Where, simplifying a lot for brevity, F_i and F_o represent the inlet and outlet mass flows with their respective stream features (temperature, concentration and pressure), $H(\cdot)$ states for the specific enthalpy function, m is the total mass inside the equipment, c_e represents the specific heat and T can be the average or outlet temperature of the medium where the energy is accumulated. Normally, the dominant dynamics in these equipment is the one of the liquid phase, so we choose the mass m, specific heat c_e and temperature T accordingly. Of course the concentrations and absolute mass of the liquid inside an equipment can vary with time too, but we neglected adding such dynamics in the mass balances because: a) it is faster than the one on the temperature and/or b) concentrations are not measured, so such dynamics cannot be checked.

Indeed, $m \cdot c_e$ can be lumped in a time constant τ and treated as a time-varying parameter to be estimated via enhanced DDR (Section 4.2).

5.2 Reconciliation results

A data historian of eight recent months of operation with the plant has been provided by Lenzing AG to test DDR procedures. The dataset provides values from all sensors depicted in Figure 7, recorded with a sampling time of five minutes. This sampling frequency is considered fast enough to cover the dominant plant dynamics, which takes around 30 minutes to stabilize after a setpoint change.

The system dynamics has been discretized by orthogonal collocation [10] using 2-degree interpolating polynomials. A moving time-window of $H = 7$ (corresponding to 35 min.) and the input-deviation approach (Section 4.1.2) are employed to define an enhanced DDR schema (Section 4.2). Some obtained results are presented and discussed below.

5.2.1 Example of gross-error detection

A selected reconciliation window of a week of operation is shown in Figure 8, where the measurements for the circulating flow of acid bath and the temperature of the saturated steam before entering W_2 heat exchangers are depicted, together with its reconciled values obtained from DDR.

At the beginning, it can be observed that some biases between the reconciled and measured values appear from time to time, especially in what seems the plant is near to steady state. This could be because the plant model in steady state is not perfect or some corrupted data affected the parameter estimation. Nevertheless, this is not considered a big issue, as these biases eventually reduce to acceptable values.

Nonetheless, note that, since the sixth day onwards, a systematic error is detected in the steam temperature: indeed this error is not constant but seems to increase with the time. However, no sensible deviation is observed in the circulating flow. This is a clear indicator of that something has happened around the last stage of heat exchangers: it could be a gross error in the measurement due to a fault in the temperature sensor, or a physical change in the process (equipment fault or operation mode).

It would be desirable to cross data with the maintenance historian to elucidate the origin of this error and, in case of online DDR, it would be recommendable to send a maintenance order to check the state of such sensor if the problem persists in time.

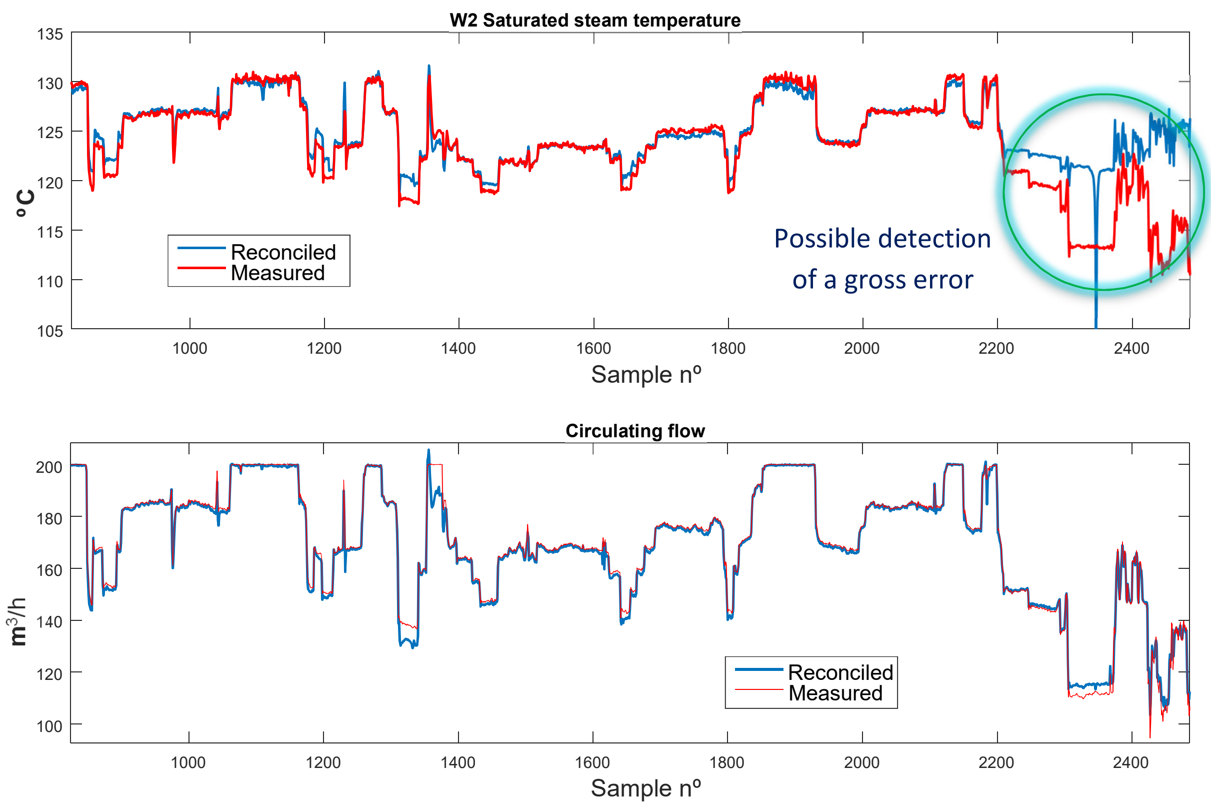


Figure 8: DDR execution during 8 days of operation.

5.2.2 Performance during transients

In order to evaluate the performance during transient behavior, a train of setpoint changes in the circulating flow and control temperature was scheduled in the plant, see Figure 9 and Figure 10. Some deviations between the measurements and the reconciled inputs can be observed just after some setpoint changes, especially in the circulating flow. This can be an indicator that we considered some slow dynamics in the model but they are faster in the real plant, so the estimated inputs by DDR are modified to fit the rest of the plant measurements. It could be also possible due to neglecting the sensors dynamics which, sometimes, might be important.

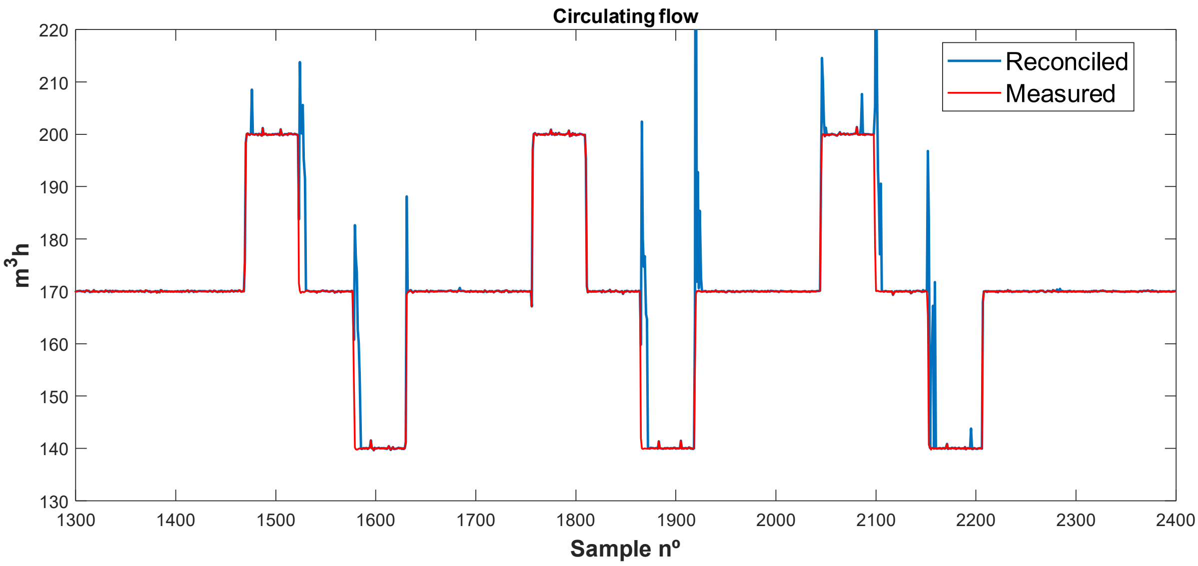


Figure 9: Induced changes in the circulating flow (control input).

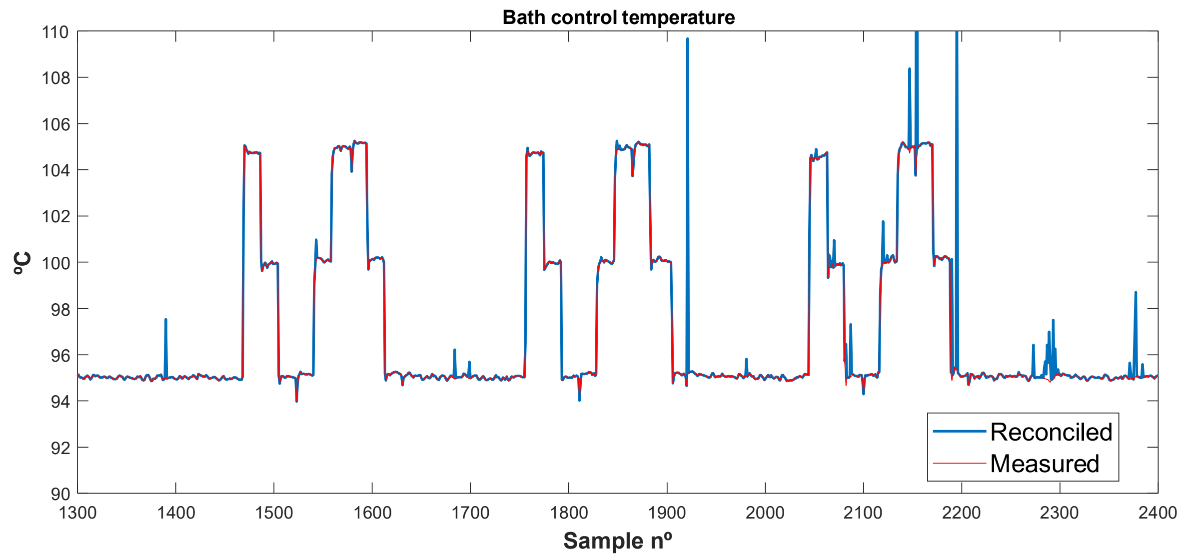


Figure 10: Induced changes in the temperature of the acid bath leaving W_2 (control input).

Figures 11 to 13 below show the reconciled values or some intermediate temperatures in the plant, for which we have measurements to compare. In general, the obtained estimates are spikier than the corresponding measurements, which could be again a problem of plant-model mismatch in our simple approximations of the system dynamics, or just a smoothing effect by the sensors due to their own dynamics (normally temperature sensors act as a low-pass filter).

Special mention needs to be done for Figure 11. There, a recurrent gap of about 2 degree C is observed between the reconciled values and the measurements. This may indicate a problem in our model with the steady-state equations in equipment around the acid-bath inlet, or an indicator of that this sensor needs recalibration.

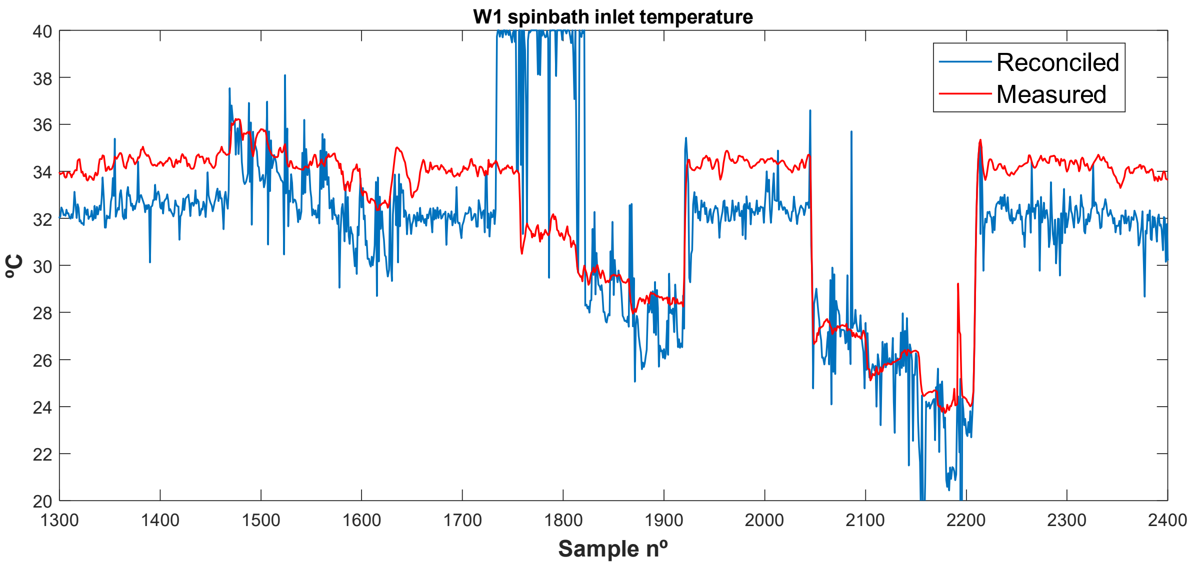


Figure 11: Variability in the temperature inlet of the bath recirculation.

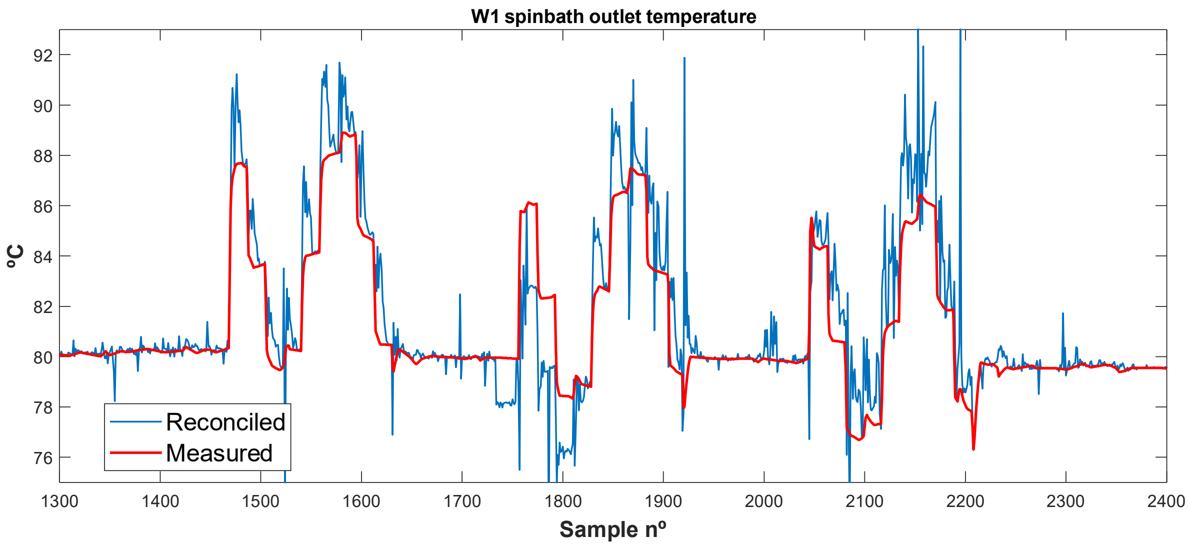


Figure 12: Temperature of the acid bath between exchanger stages W_1 and W_2.

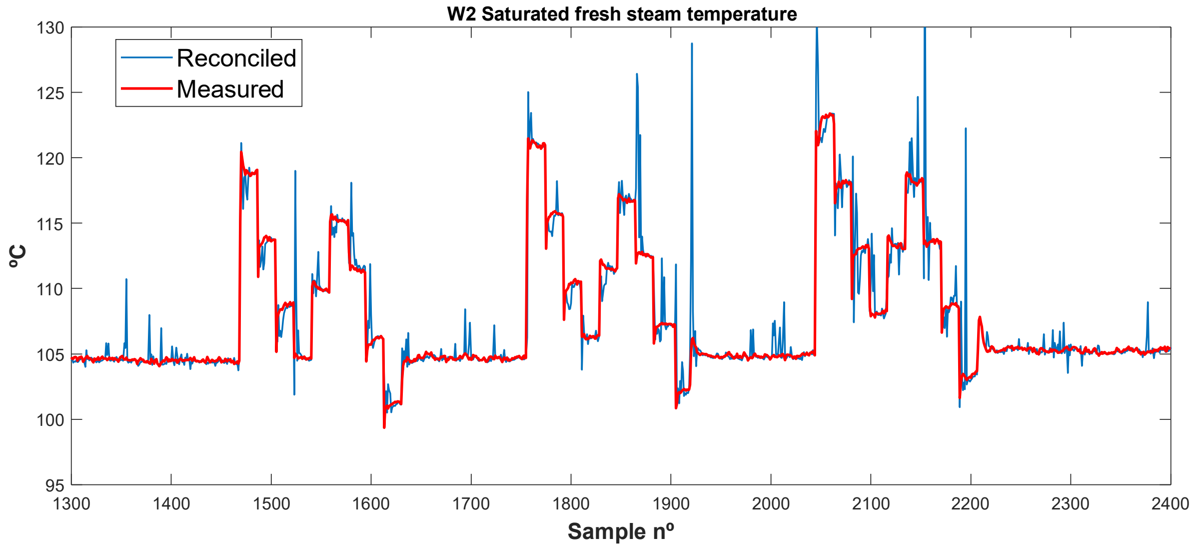


Figure 13: Temperature of the saturated steam at the W_2 inlet.

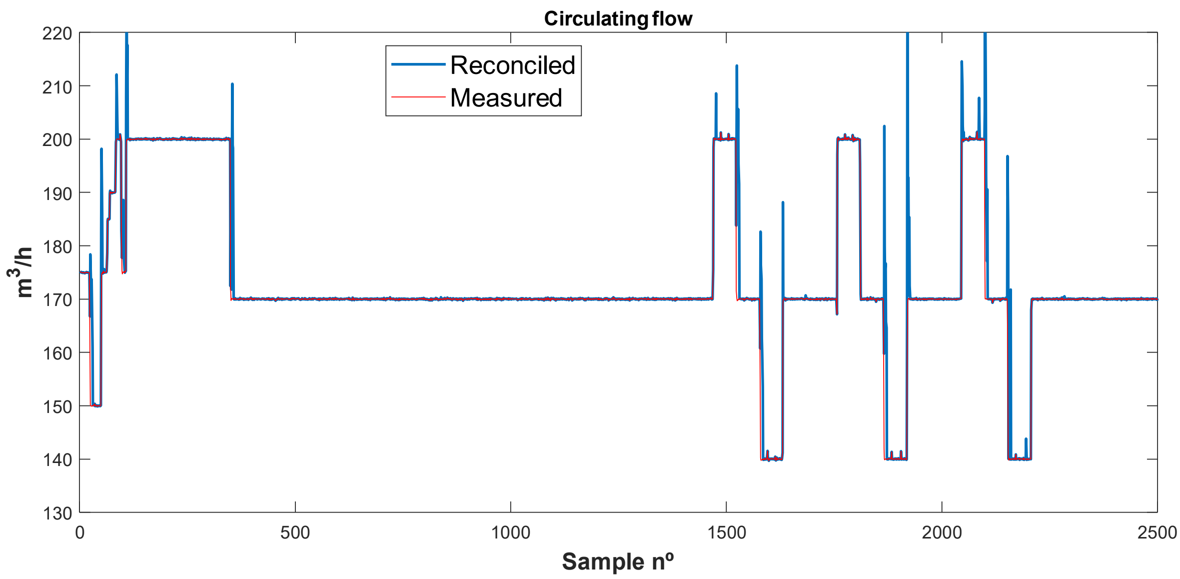
5.3 Time-varying parameter estimation

Apart from obtaining estimations for the model variables that are unmeasured, parameter estimations are provided by DDR as a byproduct. Among these, especially interesting are the estimations for slow-varying parameters, because they represent indeed the long-term dynamics which has not been considered initially in the model, such as fouling, degradation, catalyst deactivation, etc. This kind of dynamics is normally very difficult to model by first principles for a particular system: there are several influencing factors and the underlying physics is complex. Hence, estimations provided by DDR can serve as "soft sensors" from which data-driven equations can be obtained by regression or pattern identification among other known variables.

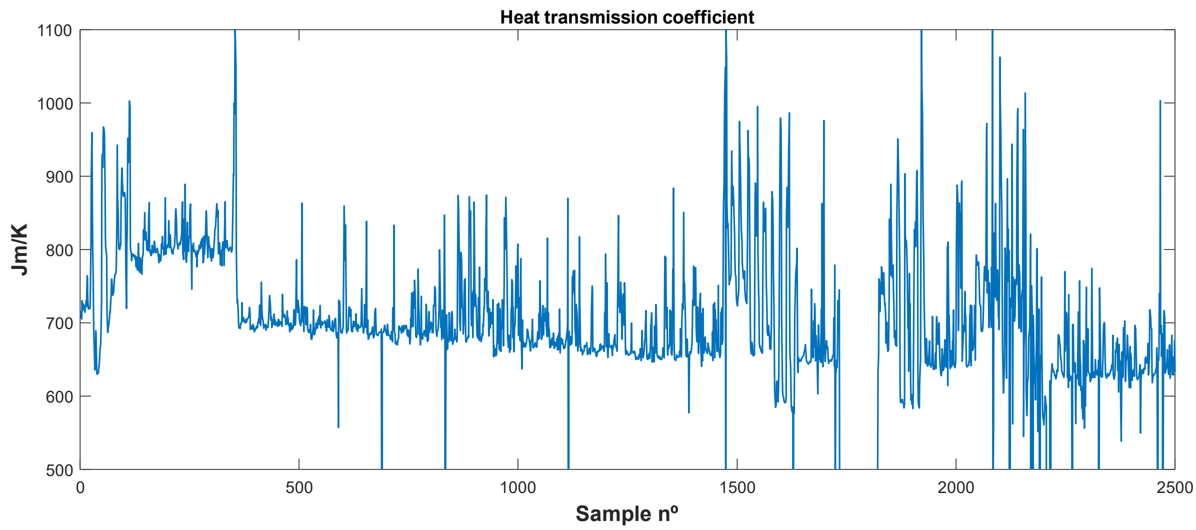
This is the case in the Lenzing evaporation plant, where the plant efficiency decreases with time due to progressive fouling in the heat exchanger sets W_1 and W_2. The fouling effect can be observed indirectly as an increase of the specific steam consumption (measurement) over a month of operation, or directly by the evolution of the heat-transmission coefficients UA (unmeasured). Therefore, we have run enhanced DDR in the test data considering the coefficients UA for W_1 and W_2 as parameters to be estimated. The results over a week of operation are displayed in Figure 14.

Although the estimation is not very smooth (typical behavior when estimating this kind of coefficients from noisy measurements) we can clearly observe a correlation between UA with respect to the circulating flow. This effect is coherent with the physics, as the heat transfer by convection is expected to increase with the flow (and vice versa).

In addition, we can observe a slow-varying trend over the time, where the heat-transmission coefficient decreases in average. This corresponds to the above-mentioned fouling effect, so that we could now decouple this data from the convection effect and try to identify a fouling model for further use in decision support systems (i.e., to predict optimal maintenance policies [21]).



(a) Induced changes in the circulating flow.



(b) Evolution of the heat-transmission coefficient.

Figure 14: Estimation of the heat-transmission coefficient at W_1 during 9 days of operation.

6 Concluding remarks

This deliverable has summarized the main ideas on dynamic data reconciliation to be applied in large-scale systems, putting special emphasis in allowing online implementations. In this way, the more suitable theoretical approaches have been reviewed and some have been tested in the multiple-effect evaporation plant of Lenzing AG.

As a first attempt, the obtained results were satisfactory enough to proof the DDR concepts in a large system. Nevertheless, the application of DDR to large processing plants has demonstrated to be challenging both from the theoretical and implementation aspects. One of the main difficulties encountered is related to the modelling : a representative dynamic model of the plant is required as a starting point to really trust DDR results, but large plants involve several complex processes difficult to model, even in steady state.

Another limiting factor is the quality of the sensors data, which needs to be checked carefully before reconciliation. In this report some ideas have been presented to palliate this issue. However, when the process is in an unconsidered operation mode or faulty data, the NLP optimization does not get a feasible solution, so estimations are lost during such time instants. In fact it has happened to us in our case study, where some fully uncoherent peaks can be observed in figures 11 and 12 from sample nº 1730 to 1830 approximately. Moreover, the treatment of all exceptions which can appear related to this issue is crucial (and also time consuming for the designer) to get a reliable set of values.

7 References

[1] M. Kujanpää, J. Hakala, T. Pajula, B. Beisheim, S. Krämer, D. Ackershott, M. Kalliski, S. Engell, U. Enste and J. L. Pitarch, Successful Resource Efficiency Indicators for process industries, vol. 290, Espoo: VTT Technology, 2017, p. 78.

[2] P. Kadlec, B. Gabrys and S. Strandt, "Data-driven Soft Sensors in the process industry," *Computers & Chemical Engineering,* vol. 33, no. 4, pp. 795-814, 2009.

[3] M. Leibman, T. Edgar and L. Lasdon, "Effcient data reconciliation and estimation for dynamic processes using nonlinear programming techniques," *Computers & Chemical Engineering,* vol. 16, pp. 963-986, 1992.

[4] N. Shrowti, K. Vilankar and R. Rhinehart, "Type-II Critical Values for a Steady-State Identifier," *Journal of Process Control,* vol. 20, no. 7, pp. 885-890, 2010.

[5] T. Huang, Steady State and Transient State Identification in an Industrial Process, MS Thesis. Oklahoma State University, 2013.

[6] R. Rhinehart, M. Su and U. Manimegalai-Sridhar, "Leapfrogging and Synoptic Leapfrogging: a new optimization approach," *Computers and Chemical Engineering,* vol. 40, pp. 67-81, 2012.

[7] R. Rhinehart, "Automated Steady and Transient State Identification in Noisy Processes," in *American Control Conference (ACC)*, Washington, DC, USA, 2013.

[8] J. Taylor and M. Laylabadi, "A novel adaptive nonlinear dynamic data reconciliation and gross error detection method," in *IEEE Conf. on Control Applications*, Munich, 2006.

[9] H. Bock, M. Diehl, D. Leineweber and J. Schlöder, "A Direct Multiple Shooting Method for Real-Time Optimization of Nonlinear DAE Processes," in *Nonlinear Model Predictive Control*, F. Allgöwer and A. Zheng, Eds., Basel, Springer, 2000, pp. 245-267.

[10] L. T. Biegler, A. M. Cervantes and A. Wächter, "Advances in simultaneous strategies for dynamic process optimization," *Chemical Engineering Science,* vol. 57, no. 4, pp. 575-593, 2002.

[11] J. Taylor and R. Moreno, "Nonlinear Dynamic Data Reconciliation: In-depth Case Study," in *IEEE Inter. Conf. on Control Applications (CCA)*, Hyderabad, 2013.

[12] V. M. Zavala, C. D. Laird and L. T. Biegler, "A fast moving horizon estimation algorithm based on nonlinear programming sensitivity," *Journal of Process Control,* vol. 18, no. 4, pp. 876-884, 2008.

[13] S. Bai, J. Thibault and D. D. McLean, "Dynamic data reconciliation: Alternative to Kalman filter," *Journal of Process Control,* vol. 16, pp. 485-498, 2006.

[14] J. Taylor, "Statistical performance analysis of nonlinear stochastic systems by the monte carlo method," *Trans. on Mathematics and Computers in Simulation,* vol. 23, no. 1, pp. 21-33, 1981.

[15] M. Bendig, Integration of Organic Rankine Cycles for Waste Heat Recovery in Industrial Processes, PhD Thesis: École Polytechnique Fédérale de Lausanne, 2015.

[16] G. Heyen, E. Maréchal and B. Kalitventzeff, "Sensitivity calculations and variance analysis in plant measurement reconciliation," *Computers & Chemical Engineering,* vol. 20, no. S1, pp. S539-S544, 1996.

[17] P. Huber, "Robust statistics," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed., Springer Berlin, 2014, pp. 1248-1251.

[18] N. Arora and L. Biegler, "Redescending estimators for Data Reconciliation and Parameter Estimation," *Computers & Chemical Engineering,* vol. 25, no. 11-12, p. 1585–1599, 2001.

[19] G. Shevlyakov, S. Morgenthaler and A. Shurygin, "Redescending M-estimators," *Journal of Statistical Planning and Inference,* vol. 138, pp. 2906-2917, 2008.

[20] M. Fuente, G. Gutierrez, E. Gomez, D. Sarabia and C. de Prada, "Gross error management in data reconciliation," in *9th International Symposium on Advanced Control of Chemical Processes*, Whistler, British Columbia, Canada, 2015.

[21] J. L. Pitarch, C. G. Palacín, C. de Prada, B. Voglauer and G. Seyfriedsberger, "Optimisation of the Resource Efficiency in an Industrial Evaporation System," *Journal of Process Control,* vol. 56, pp. 1-12, 2017.

[22] J. J. J. Chen, "Comments on improvements on a replacement for the logarithmic mean," *Chemical Engineering Science,* vol. 42, no. 10, pp. 2488-2489, 1987.