New Method to Perform Data Reconciliation with OpenModelica and ThermoSysPro

Daniel Bouskela[1] Audrey Jardin[1] Arunkumar Palanisamy[2]
Lennart Ochel[2] Adrian Pop[3]

[1]EDF Lab Chatou, France, {daniel.bouskela,audrey.jardin}@edf.fr
[2] RISE AB, Sweden, {arunkumar.palanisamy,lennart.ochel}@ri.se
[3] PELAB - Linköping University, Sweden, adrian.pop@liu.se

Abstract

Data reconciliation aims at improving the accuracy of measurements by reducing the effect of random errors in the data. This is achieved by introducing redundancies on the measured quantities in the form of constraints based on fundamental physical laws such as mass, momentum and energy balance equations. These constraints are called the auxiliary conditions. Modelica is an equational language that was conceived to express models based on first principle physics for the purpose of behavioral simulation. This paper shows how to reuse such models for the purpose of data reconciliation. The novelty is to automatically extract the auxiliary conditions from the Modelica model. Then the reconciled values are computed using a least square method constrained by the auxiliary conditions, as specified by the VDI 2048 standard. The new method has been implemented in OpenModelica. A simple example built with ThermoSysPro illustrates the method in detail.

Keywords: data reconciliation, Modelica, model reuse, cyber-physical systems, structural analysis

1 Introduction

The safe and efficient operation and maintenance of power plants rely on plant data. Therefore, ensuring the quality of plant measurements such as pressures, temperatures and mass flow rates is essential. However, plant data are subject to measurement errors that put a limitation on their efficient use for plant monitoring, diagnosis and prognosis because they lead to uncertainties in the assessment of the plant state. The consequence is a decrease in production because of safety regulations that put stringent limits on plant operation. It is therefore important to compute the best estimates of the measurement uncertainties in order to regain satisfactory operational margins. Best estimates can be obtained by combining statistics on the data with a priori knowledge from the expert expressed in the form of physical models. Data reconciliation has been conceived for the process industry with this principle in mind and is the subject of the VDI 2048 standard (VDI, 2017). It has been used for several process related issues such as finding lost megawatts in power plants (Langenstein et al., 2004) or detecting sensor and actuator faults in hydraulic systems (Bedjaoui et al., 2008). This paper follows the VDI 2048 standard methodology.

Data reconciliation aims at improving the accuracy of measurements by reducing the effect of random errors in the data. The main difference between data reconciliation and other data improvements techniques is that data reconciliation uses a model to express the physical constraints on the variables of interest and adjusts their measured values such that the estimates satisfy the constraints: the variables are thus reconciled. The physical constraints on the variables of interest are called the *auxiliary conditions*. The main benefit of introducing redundancy in the form of auxiliary conditions is that the estimates have lower uncertainties than the initial measurements.

Dedicated tools such as VALI (Belsim, 2021) exist to perform data reconciliation, but they require the physical models to be specifically developed for that purpose. This makes data reconciliation costly and difficult to use. A natural answer to that problem is to perform data reconciliation on existing Modelica models (Modelica Association, 2021), developed and validated for the general purpose of plant operation at large.

The objective of this paper is to present a new method to perform data reconciliation using Modelica models. The novelty of the method lies in the automatic extraction of the auxiliary conditions from the Modelica model. Once the auxiliary conditions are extracted, the variables of interest can be reconciled using the inputs provided by the user in the form of measurement data and correlation matrices, and the numerical procedure described in the VDI 2048 standard.

Although dynamic data reconciliation is possible (Bedjaoui et al., 2008; Bai and Thibault, 2010), the VDI 2048 standard considers only *steady-state data reconciliation*, which means that measurements are conducted while the system is under quasi steady-state conditions. For the parts of the system where this assumption is not valid, additional uncertainties must be added to fluctuating quantities. Therefore, the new

DOI
10.3384/ecp21181453

method presented in this paper only applies to static models, i.e. models with physical laws expressed in the form of algebraic equations. These models can be obtained by removing the differential part of the physical equations. Also, the VDI 2048 assumes that the auxiliary conditions are *exact physical laws*. This means that the model should be sufficiently representative of the real system, for instance that leaks, or other serious disturbances, are not overlooked. For the quantities such as temperatures that cannot be represented by exact physical laws, e.g. when dealing with thermal correlations, additional uncertainties must be added to the quantities to account for the uncertainties in the physical laws. The uncertainties on variables of interest that cannot be directly measured such as specific enthalpies can be obtained using uncertainty propagation techniques (Dutfoy et al., 2009).

The data reconciliation method is summarized in Section 2. The new algorithm to automatically extract the auxiliary conditions from a Modelica model is given and applied to perform data reconciliation on the simple example of a splitter in Section 3.

2 Data Reconciliation in a Nutshell

2.1 Mathematical Formulation

Let $\hat{x} = \{\hat{x}_i\}_{1\le i\le n}$ be measurements on physical quantities $x = \{x_i\}_{1\le i\le n}$ that are constrained by exact physical laws represented by a set of algebraic equations $\mathcal{C}$ such that

$$\mathcal{C}(\bar{x}) = 0 \tag{1}$$

where $\bar{x}$ denotes the vector of the true values of x. The true values are unknown, and the objective is to provide the best estimates for them. Notice that in general the measurements do not satisfy Equation (1): $\mathcal{C}(\hat{x}) \neq 0$.

The random unbiased error ε_i on measurement $\hat{x}_i$ is described by a Gaussian noise around $\bar{x}_i$:

$$\hat{x}_i = \bar{x}_i + \varepsilon_i \tag{2}$$

with

$$\varepsilon_i = \mathcal{N}(0,\sigma^2) \tag{3}$$

$\mathcal{N}(0,\sigma^2)$ being a Gaussian distribution of mean value 0 and standard deviation σ. This is valid according to the central limit theorem if a sufficient number of measurements are conducted with sufficient accuracy. The mean value of Equation (2) yields the true value of $\hat{x}_i$:

$$\bar{x}_i = E(\hat{x}_i) \tag{4}$$

However, computing the true value using Equation (4) requires a sufficient number of measurements with different sensors and measuring techniques to avoid biases, which is impractical when dealing with a large number of measured quantities.

To find an estimate of the true value, the VDI 2048 standard states that the Gaussian distribution is applicable, even with only one measured value for each variable x_i. This assertion is justified by the fact that each measured value deviates from the true value by a sum of random, mostly independent deviations that makes the central limit theorem applicable with good approximation.

The weight or half-width confidence interval for $\hat{x}_i$ is defined by:

$$w_{\hat{x},i} = \lambda_p \cdot \sigma_{\hat{x},i} \tag{5}$$

where $\sigma_{\hat{x},i} = \sqrt{E[(\hat{x}_i - \bar{x}_i)^2]}$ is the standard deviation of $\hat{x}_i$ and λ_p is the quantile of normal distribution with probability p. Then $\bar{x}_i$ lies within the confidence interval $\hat{x}_i \pm \lambda_p \cdot \sigma_{\hat{x},i}$ with the probability p:

$$P\left(|\bar{x}_i - \hat{x}_i| \le \lambda_p \cdot \sigma_{\hat{x},i}\right) = p \tag{6}$$

For $p = 95\%$, $\lambda_p = 1.96$ which yields the following confidence interval:

$$w_{\hat{x}_i} = \lambda_{95\%} \cdot \sigma_{\hat{x},i} = 1.96 \cdot \sigma_{\hat{x},i} \tag{7}$$

The covariance matrix of $\hat{x}$ is defined as:

$$S_{\hat{x}} = \begin{cases} S_{\hat{x},i,i} = {\sigma_{\hat{x},i}}^2 \\ S_{\hat{x},i,j} = r_{\hat{x},i,j} \cdot \sigma_{\hat{x},i} \cdot \sigma_{\hat{x},j} \end{cases} \tag{8}$$

where the correlation coefficients $r_{\hat{x},i,j}$ are such that $\left|r_{\hat{x},i,j}\right| \le 1$. $r_{\hat{x}}$ is the correlation matrix. Expressed as function of the weights, the covariance matrix is:

$$S_{\hat{x}} = \begin{cases} S_{\hat{x},i,i} = \left(w_{\hat{x},i}/\lambda_p\right)^2 \\ S_{\hat{x},i,j} = r_{\hat{x},i,j} \cdot w_{\hat{x},i} \cdot w_{\hat{x},j}/\lambda_p^2 \end{cases} \tag{9}$$

The estimated values are found in the form of the reconciled values of x which are denoted $\bar{\bar{x}}$ in the sequel. They are obtained by finding the point in the subspace defined by Equation (1) which is closer to measurements with lower uncertainties than measurements with higher uncertainties. They are thus computed using a least square method where the weighing matrix is the inverse of the covariance matrix. Therefore, the objective function is:

$$J(x) = (x - \hat{x}) \cdot S_{\hat{x}}^{-1} \cdot (x - \hat{x}) \tag{10}$$

and the minimization problem to be solved is:

$$\begin{cases} J(\bar{\bar{x}}) = \min_x \left((x - \hat{x}) \cdot S_{\hat{x}}^{-1} \cdot (x - \hat{x})\right) \\ \mathcal{C}(\bar{\bar{x}}) = 0 \end{cases} \tag{11}$$

The vector of improvements is defined as the difference between the estimated values and the measured values:

$$\nu = \bar{\bar{x}} - \hat{x} \tag{12}$$

The covariance matrices of the reconciled values and of the improvements, derived from the general formula of error propagation, are respectively:

$$S_{\bar{\bar{x}}} = \frac{\partial \bar{\bar{x}}}{\partial \hat{x}} \cdot S_{\hat{x}} \cdot \left(\frac{\partial \bar{\bar{x}}}{\partial \hat{x}}\right)^T \tag{13}$$

$$S_{\nu} = \frac{\partial \nu}{\partial \hat{x}} \cdot S_{\hat{x}} \cdot \left(\frac{\partial \nu}{\partial \hat{x}}\right)^T \tag{14}$$

Noticing that

$$D_{S_{\hat{x}}^{-1}}(\nu_1, \nu_2) = \nu_1 \cdot S_{\hat{x}}^{-1} \cdot \nu_2 \tag{15}$$

is a scalar product, then

$$J(x) = D_{S_{\hat{x}}^{-1}}(x - \hat{x}, x - \hat{x}) = \|x - \hat{x}\|^2_{S_{\hat{x}}^{-1}} \tag{16}$$

is the square of the distance between x and $\hat{x}$ weighted by $S_{\hat{x}}^{-1}$. Let r be the number of auxiliary conditions (i.e. the size of $\mathcal{C}$), which is also the redundancy level. Solving the minimization problem of Equation (11) amounts to finding a point $\bar{\bar{x}}$ on the r-dimensional surface defined by $\mathcal{C}(x) = 0$ which is the orthogonal projection of the measured point $\hat{x}$ on the surface $\mathcal{C}(x) = 0$. The projection is done according to a metrics such that the coordinates of the projected point are closer to measurements with lower uncertainties than to measurements with higher uncertainties. The uncertainties of the reconciled values are thus reduced because the distance from the true values to the orthogonal projection is always smaller than the distance to the initial point:

$$\|\bar{\bar{x}} - \bar{x}\|^2_{S_{\hat{x}}^{-1}} \le \|\hat{x} - \bar{x}\|^2_{S_{\hat{x}}^{-1}} \qquad (17)$$

Moreover, the higher the value of r, the higher the uncertainty reduction because each time r is increased by one unit, one additional orthogonal projection is performed to a smaller subspace that brings the reconciled values closer to the true values. Thus, data reconciliation provides the best estimate of the variables of interest from the measured values $\hat{x}_i$, the weights $w_{\hat{x}_i}$ and the auxiliary conditions $\mathcal{C}(\bar{x}) = 0$.

The optimization problem of Equation (11) can be solved using the Lagrange multiplier method. The Lagrangian is:

$$L(x, \lambda) = J(x) + 2 \cdot \lambda^T \cdot \mathcal{C}(x) \qquad (18)$$

where λ are the Lagrange multipliers. The values of x that yield the minimum value of $J(x)$ are obtained by solving the following equation system:

$$\begin{cases} \frac{\partial L}{\partial x} = 2 \cdot S_{\hat{x}}^{-1} \cdot \; (x - \hat{x}) + 2 \cdot \frac{d\mathcal{C}}{dx}^T \cdot \lambda = 0 \\ \frac{\partial L}{\partial \lambda} = \mathcal{C}(x) = 0 \end{cases} \qquad (19)$$

The vector of contradictions is defined as:

$$u = (\hat{x} - \bar{x}) - (\bar{\bar{x}} - \bar{x}) = \hat{x} - \bar{\bar{x}} \qquad (20)$$

Therefore, the vector of contradictions corresponds, in absolute value, to the vector of improvements. Its square value is:

$$J_0 = \|u\|^2_{S_{\hat{x}}^{-1}} = J(\bar{\bar{x}}) = u \cdot S_{\hat{x}}^{-1} \cdot u \qquad (21)$$

Because u is standard normally (i.e. $\mathcal{N}(0,1)$) distributed, J_0 being the square of u is a χ^2-distributed function of r degrees of freedom (according to VDI 2048). Therefore, the following relationship holds with statistical certainty of probability p:

$$J_0 \le \chi^2_{r,p} \qquad (22)$$

If Condition (22) is not satisfied, then the result for the reconciled values should be rejected because the vector of contradictions is too large. This can happen if some improvements are too large making the corresponding reconciled values fall out of their confidence ranges. This can be checked with the following individual tests:

$$|\bar{\bar{x}}_i - \hat{x}_i| / \sqrt{S_{v,i,i}} \le \lambda_p \qquad (23)$$

where $S_{v,i,i}$ is the i^{th} diagonal element of the covariance matrix of the improvements. From the physical viewpoint, the failure of Condition (23) means that the constraints are not fully representative of the actual system behavior (e.g., some system leaks are not modelled), thus that the assumption that the constraints are exact physical laws is not verified, or that the measurements are incorrect (e.g., due to faulty sensors or poor estimations of their confidence level).

2.2 Numerical Resolution

The VDI 2048 standard recommends linearizing Equation (1) under the assumption that the improvements v are small:

$$\mathcal{C}(x) = \mathcal{C}(\hat{x}) + \frac{d\mathcal{C}}{dx}(\hat{x}) \cdot (x - \hat{x}) \qquad (24)$$

and use an iterative method to solve Equation (19). This amounts to constructing the suite $\{x_k\}_{0 \le k \le N}$ such that:

$$\begin{cases} x_0 = \hat{x} \\ S_{\hat{x}}^{-1} \cdot (x_{k+1} - x_k) + \frac{d\mathcal{C}^T}{dx}(x_k) \cdot \lambda_k = 0 \\ \mathcal{C}(x_k) + \frac{d\mathcal{C}}{dx}(x_k) \cdot (x_{k+1} - x_k) = 0 \\ \bar{\bar{x}} = x_N \end{cases} \qquad (25)$$

N is chosen to satisfy the convergence criteria:

$$L(x_N, \lambda_N)/r < \varepsilon \qquad (26)$$

ε being a small number such as $\varepsilon = 10^{-10}$, and r being the size of $\mathcal{C}$.

Equation (25) can be rewritten as follows:

$$\begin{cases} x_0 = \hat{x} \\ x_{k+1} = x_k - S_{\hat{x}} \cdot F_k^T \cdot \lambda_k \\ F_k \cdot S_{\hat{x}} \cdot F_k^T \cdot \lambda_k = \mathcal{C}(x_k) \\ \bar{\bar{x}} = x_N \end{cases} \qquad (27)$$

with

$$F_k = \frac{d\mathcal{C}}{dx}(x_k) \qquad (28)$$

From Equation (19) and Equation (24), the reconciled values are:

$$\bar{\bar{x}} = \hat{x} - S_{\hat{x}} \cdot F_N^T \cdot (F_N \cdot S_{\hat{x}} \cdot F_N^T)^{-1} \cdot \mathcal{C}(\hat{x}) \qquad (29)$$

From equations (13), (14) and (29), the correlation matrices of the improvements and of the reconciled values are given by:

$$S_v = S_{\hat{x}} \cdot F_N^T \cdot F^* \qquad (30)$$

$$S_{\bar{\bar{x}}} = S_{\hat{x}} - S_v \qquad (31)$$

where F^* is the solution of the equation

$$(F_N \cdot S_{\hat{x}} \cdot F_N^T) \cdot F^* = F_N \cdot S_{\hat{x}} \qquad (32)$$

3 Performing Data Reconciliation with Modelica Models

3.1 Physical Laws and Boundary Conditions

A valid Modelica model is always a square model. A square model has as many unknown variables as equations to compute them. The model equations can be divided into two groups:

- The group of physical equations that represent physical laws such as the mass, momentum or energy balance equations, or empirical laws such as thermal or pressure losses correlations. This group is always underdetermined because physical laws express constraints between physical quantities, but do not provide any means to compute them in a unique way.
- The group of boundary conditions that provide the additional constraints to the group of physical equations to form a square system. Boundary conditions represent assumptions on the

environment of the system that are necessary to undertake a numerical experiment through simulation.

Let us take the Ohm's law as a simple example. The equation $U = R \cdot I$ expresses a constraint between the voltage U, the resistance R and the current I. To compute numerical values out of this equation, one must provide values for exactly two quantities in the form of boundary conditions, e.g., $U = 220\,V$ and $R = 50\,\Omega$, or provide a physical correlation to e.g. compute R as a function of the temperature T, in such case other boundary conditions related to the thermal condition of the system must be provided.

3.2 Well-posedness of the Data Reconciliation Problem

As boundary conditions do not represent physical laws, they cannot be part of the auxiliary conditions. Therefore, to use a valid Modelica model to represent the auxiliary conditions of a data reconciliation problem, the boundary conditions *that are related to the variables of interest* must be automatically removed from the Modelica model before computing the reconciled values. Each removed boundary condition must be replaced by a variable of interest in the sense explained in Section 3.6. It will be shown in the sequel that this action reduces by one unit the number of auxiliary conditions. When reducing the number of auxiliary conditions, it can happen that some variables of interest are not constrained anymore by any auxiliary condition.

Therefore, for the data reconciliation problem to be well-posed, the following conditions must be met:

$$f = n - r \geq 1 \tag{33}$$

$$r \geq 1 \tag{34}$$

$$\mathcal{C}(x_1, \dots, x_n) = 0 \tag{35}$$

where f is the number of degrees of freedom, n is the number of variables of interest and r is the number of auxiliary conditions (or number of redundancies). Condition (35) means that all variables of interest must appear in at least one auxiliary condition. The variables of interest that do not appear in any auxiliary condition cannot be reconciled (which means that their reconciled values are equal to their measured values).

A valid Modelica model being a square model, $f = 0$ and Condition (33) is thus violated. This is another way to state why the model must be pre-processed to be fit for data reconciliation. The extraction algorithm presented in the sequel decreases the value of r by removing boundary conditions until no boundary conditions related to the variables of interest are left. Violation of Conditions (34) and (35) can thus happen when a group of variables of interest is related to a larger group of boundary conditions. The extraction algorithm will always satisfy Condition (33) because static models always involve boundary conditions, at least for energy systems as shown in (El Hefni and Bouskela, 2019).

3.3 Simple Example: Splitter

This section presents an illustrating example of a splitter that will be our companion throughout the rest of the paper.

A splitter is a device that separates an incoming flow into two outgoing flows. The physical laws of the splitter are the mass, momentum and energy balance equations. There is one mass balance equation and one energy balance equation to account for the flow separation, and three momentum balance equations, one for the incoming pipe and one for each outgoing pipe to account for the pressure losses inside the pipes. In the following it is assumed that all mass flow rates are positive, the fluid flowing from the left to the right in each pipe, cf. Figure 1.

The mass balance equation in the mixing volume is:

$$0 = Q_1 - Q_2 - Q_3$$

where Q_i is the mass flow rate of the fluid in the i^{th} pipe.

The energy balance equation in the mixing volume is:

$$0 = h_1 \cdot Q_1 - h_2 \cdot Q_2 - h_3 \cdot Q_3 + W$$

where h_i is the specific enthalpy of the fluid in the volume upstream of the i^{th} pipe, and W is the heating power.

The momentum balance equations in the 3 pipes are:

$$P_{1,l} - P_{1,r} = k_1.Q_1^2$$

$$P_{2,l} - P_{2,r} = k_2.Q_2^2$$

$$P_{3,l} - P_{3,r} = k_3.Q_3^2$$

where $P_{i,l}$ and $P_{i,r}$ are resp. the pressure of the fluid entering (at the left) and exiting (at the right) the i^{th} pipe (subscript *l* stands for *left* and subscript *r* stands for *right*), and k_i is the pressure loss coefficient in the i^{th} pipe, which is assumed to be an exact parameter.

The pressure P inside the mixing volume is related to the pressures in the neighboring pipes:

$$P = P_{1,r}$$

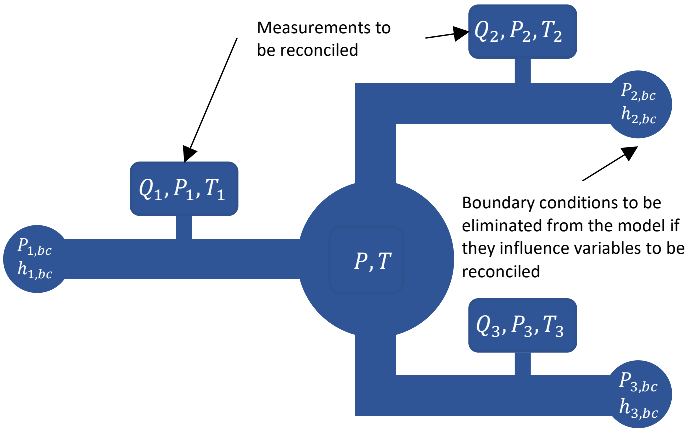
$$P = P_{2,l}$$

$$P = P_{3,l}$$

The specific enthalpies entering the outgoing pipes are equal to the specific enthalpy h in the mixing volume:

$$h_2 = h$$

$$h_3 = h$$



Measurements to be reconciled

Boundary conditions to be eliminated from the model if they influence variables to be reconciled

Figure 1. Splitter

The mean pressures inside the pipes are given, by:

$$P_1 = (P_{1,l} + P_{1,r})/2$$
$$P_2 = (P_{2,l} + P_{2,r})/2$$
$$P_3 = (P_{3,l} + P_{3,r})/2$$

where P_i is the mean pressure of the fluid in the ith pipe.

The specific enthalpies in the volumes are related to the temperatures by the following equations:

$$h = c_p \cdot T + b \cdot P$$
$$h_1 = c_p \cdot T_1 + b \cdot P_1$$
$$h_2 = c_p \cdot T_2 + b \cdot P_2$$
$$h_3 = c_p \cdot T_3 + b \cdot P_3$$

where T_i is the mean temperature of the fluid in the ith pipe. c_p is the specific heat capacity of the fluid and b accounts for the pressure dependence of the specific enthalpy. Those two last quantities are assumed to be exact parameters.

This model has 17 equations and 21 unknowns. Therefore 4 boundary conditions are needed. There are many different possibilities for choosing the boundary conditions. For instance, one may fix the pressure at each open end of the 3 pipes, or fix the pressure at the inlet of the incoming pipe and the mass flow rates inside the 2 outgoing pipes. Additionally, the specific enthalpy or the temperature of the fluid at the inlet of the incoming pipe must be fixed, so that 4 boundary conditions are properly fixed. The way to select the boundary conditions is important to define the proper scenarios to perform validation tests of the model, but does not matter for data reconciliation as they will be eliminated.

For the purpose of this example, the equations involving boundary conditions are:

$$P_{1,l} = P_{1,bc}$$
$$Q_2 = Q_{2,bc}$$
$$Q_3 = Q_{3,bc}$$
$$h_1 = h_{1,bc}$$

where $P_{1,bc}$, $Q_{2,bc}$, $Q_{3,bc}$ and $h_{1,bc}$ are the boundary conditions with fixed values.

3.4 The Set $\mathcal{C}$ of Auxiliary Conditions and the Set $\mathcal{S}$ of Intermediate Equations

The auxiliary conditions $\mathcal{C}(x) = 0$ must be automatically extracted from the Modelica model $\mathcal{M}(x, z) = 0$, where x are the variables to be reconciled, and z are the other variables of the model.

The equations in $\mathcal{C}(x) = 0$ almost always involve a subset y of z. This is why $\mathcal{C}(x) = 0$ will be denoted $\mathcal{C}(x, y) = 0$ in the sequel. The vector x will be called the *variables of interest* or the *known variables* as measurement values are provided for x. The vector y will be called the *intermediate variables*.

The extraction problem consists in extracting the set of auxiliary conditions $\mathcal{C}(x, y) = 0$ and the set $\mathcal{S}(x, y) = 0$ of intermediate equations that compute the intermediate variables from the known variables. Therefore, the system $\mathcal{C}(x, y) = 0$ is a non-square problem that has more variables of interest than equations, while $\mathcal{S}(x, y) = 0$ is a square system that has as many equations as intermediate variables.

3.5 Reformulating the Data Reconciliation Problem with Sets $\mathcal{C}$ and $\mathcal{S}$

Equation (27) is rewritten to reveal set $\mathcal{S}$:

$$\begin{cases} x_0 = \hat{x} \\ \mathcal{S}(x_0, y_0) = 0 \\ x_{k+1} = x_k - S_{\hat{x}} \cdot F_k^T \cdot \lambda_k \\ F_k \cdot S_{\hat{x}} \cdot F_k^T \cdot \lambda_k = \mathcal{C}(x_k) \\ \mathcal{S}(x_k, y_k) = 0 \\ \bar{\bar{x}} = x_N \end{cases} \tag{36}$$

with

$$F_k = \frac{d\mathcal{C}}{dx}(x_k, y_k) \tag{37}$$

F_k is computed by solving the following equation system:

$$\begin{cases} \frac{d\mathcal{C}}{dx} = \frac{\partial \mathcal{C}}{\partial x} + \frac{\partial \mathcal{C}}{\partial y} \cdot \frac{dy}{dx} \\ \frac{\partial \mathcal{S}}{\partial x} + \frac{\partial \mathcal{S}}{\partial y} \cdot \frac{dy}{dx} = 0 \end{cases} \tag{38}$$

The Jacobian matrices $\frac{\partial \mathcal{C}}{\partial x}$, $\frac{\partial \mathcal{C}}{\partial y}$, $\frac{\partial \mathcal{S}}{\partial x}$ and $\frac{\partial \mathcal{S}}{\partial y}$ can be computed analytically from sets $\mathcal{C}$ and $\mathcal{S}$.

3.6 Algorithm to Extract Set $\mathcal{C}$ and Set $\mathcal{S}$

The extraction algorithm relies on the BLT (Block Lower Triangular) decomposition of the equation system of the full Modelica model $\mathcal{M}$.

Table 1. BLT of the Splitter.

Variable	*Equation*	
$h_{1,bc}$	$h_{1,bc} = 100000.0$	binding
$Q_{3,bc}$	$Q_{3,bc} = 2.0$	binding
$Q_{2,bc}$	$Q_{2,bc} = 1.0$	binding
$P_{1,bc}$	$P_{1,bc} = 300000.0$	binding
Q_1	$0 = Q_1 - Q_2 - Q_3$	
Q_2	$Q_2 = Q_{2,bc}$	
Q_3	$Q_3 = Q_{3,bc}$	
P	$P = P_{1,r}$	
P_1	$P_1 = (P_{1,l} + P_{1,r})/2$	
P_2	$P_2 = (P_{2,l} + P_{2,r})/2$	
P_3	$P_3 = (P_{3,l} + P_{3,r})/2$	
$P_{1,l}$	$P_{1,l} = P_{1,bc}$	
$P_{1,r}$	$P_{1,l} - P_{1,r} = k_1.Q_1^2$	
$P_{2,l}$	$P = P_{2,l}$	
$P_{2,r}$	$P_{2,l} - P_{2,r} = k_2.Q_2^2$	
$P_{3,l}$	$P = P_{3,l}$	
$P_{3,r}$	$P_{3,l} - P_{3,r} = k_3.Q_3^2$	
h	$h_2 = h$	

h_1	$h_1 = h_{1,bc}$
h_2	$0 = h_1 \cdot Q_1 - h_2 \cdot Q_2 - h_3 \cdot Q_3$
h_3	$h_3 = h$
T	$h = c_p \cdot T + b \cdot P$
T_1	$h_1 = c_p \cdot T_1 + b \cdot P_1$
T_2	$h_2 = c_p \cdot T_2 + b \cdot P_2$
T_3	$h_3 = c_p \cdot T_3 + b \cdot P_3$

The BLT is obtained by assigning to each variable z of $\mathcal{M}$ the equation E_z that computes it. Because there is a bijection between the set of equations and the set of variables in $\mathcal{M}$, each equation E_z can be uniquely labelled by the name of the variable z that it solves. Therefore, the term Equation z will refer to the equation that solves z, as established by the BLT. The BLT of the Splitter is given in Table 1. To avoid any confusion between variables and equations, Equation z will be denoted $\breve{z}$. For instance, $\breve{Q}_1$ stands for equation $0 = Q_1 - Q_2 - Q_3$, cf. Table 1. A binding is a fixed value assigned to a variable (it is not an equation although it can be found in the BLT).

In the following, the BLT of model $\mathcal{M}$ will also be denoted by $\mathcal{M}$ because the BLT of $\mathcal{M}$ contains all the equations of $\mathcal{M}$. The set $\mathcal{M}'$ is the set $\mathcal{M}$ without the binding equations. In the example:

$$\mathcal{M}' = \mathcal{M} - \{h_{1,bc}, Q_{3,bc,} Q_{2,bc,} P_{1,bc}\}$$

The overall principle of the extraction algorithm is shown in Figure 2. The algorithm starts from set $\mathcal{C}'$ which is the set of equations of $\mathcal{M}'$ that compute the variables of interest:

$$\mathcal{C}' = \breve{x} \cap \mathcal{M}' \tag{39}$$

Let us assume that the variables of interest are:

$$x = \{x_i\} = \{Q_1, Q_2, Q_3, P_1, P_2, P_3, T_1, T_2, T_3, T\}.$$

Then:

$$\mathcal{C}' = \{\breve{x}_i\} \cap \mathcal{M}' = \{\breve{Q}_1, \breve{Q}_2, \breve{Q}_3, \breve{P}_1, \breve{P}_2, \breve{P}_3, \breve{T}_1, \breve{T}_2, \breve{T}_3, \breve{T}\}.$$

For each equation $\breve{x}_i$ in set $\mathcal{C}'$, set $\mathcal{S}_i$ is built by finding the equations in $\mathcal{M}'$ that compute the intermediate variables y_{ij} involved in $\breve{x}_i$ as a function of the variables of interest which are known variables. This procedure is called the chain rule in the following. Its formal specification is shown in Listing 1.

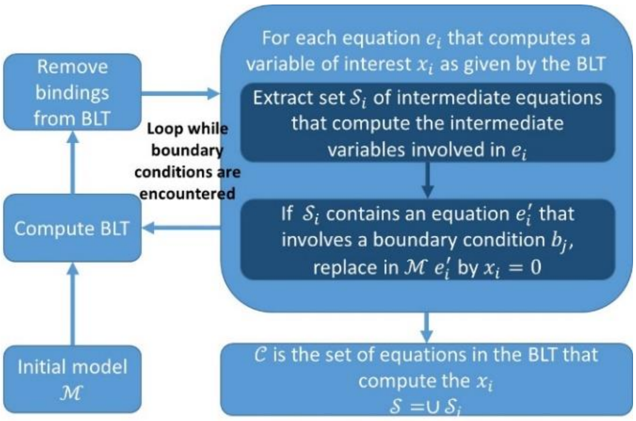

Remove bindings from BLT
Loop while boundary conditions are encountered
Compute BLT
Initial model $\mathcal{M}$
For each equation e_i that computes a variable of interest x_i as given by the BLT
Extract set $\mathcal{S}_i$ of intermediate equations that compute the intermediate variables involved in e_i
If $\mathcal{S}_i$ contains an equation e_i' that involves a boundary condition b_j, replace in $\mathcal{M}$ e_i' by $x_i = 0$
$\mathcal{C}$ is the set of equations in the BLT that compute the x_i
$\mathcal{S} = \cup\, \mathcal{S}_i$

Figure 2. Principle of the extraction algorithm

Listing 1. Procedure for extracting set $\mathcal{S}_i$

```
set S = empty set;
set V_eq = set of intermediate variables y in equation $\breve{x}_i$;
call extract (S, V_eq, status, fail_eq);
// S contains $\mathcal{S}_i$, status contains $r(\breve{x}_i)$, fail_eq contains $\widetilde{b(\breve{x}_i)}$
 define procedure extract (S, V, status, fail_eq)
    set status = SUCCEED;
    for each variable y in V
       if v is a boundary condition then
          set status = FAIL;
       else
          eq = equation in BLT – {bindings}  that computes y;
          if eq exists and eq is not in S then
             insert eq into S;
             set V_eq = set of intermediate variables y in eq;
             call extract (S, V_eq, status, fail_eq);
             if status == FAIL then
                set fail_eq = eq;
                exit procedure;
             end if;
          end if;
       end if;
    end for;
end procedure;
```

Let us apply the chain rule to equation $\breve{T}_3$. Equation $\breve{T}_3$ involves intermediate variables P_3 and h_3. This is denoted by $\breve{T}_3 \rightarrow \breve{P}_3$ and $\breve{T}_3 \rightarrow \breve{h}_3$. Then carrying on with this chain rule yields:

$$\breve{T}_3 \rightarrow \breve{h}_3 \rightarrow \breve{h}_2 \rightarrow \breve{h}_1 \rightarrow \breve{h}_{1,bc} \rightarrow stop.$$

The chain rule is stopped because $h_{1,bc}$ is a boundary condition that cannot be included in the data reconciliation problem. The outcome of the chain rule applied to equation $\breve{x}_i$ is denoted $r(\breve{x}_i)$. The boundary condition that made the chain rule fail for equation $\breve{x}_i$ is denoted $b(\breve{x}_i)$. If the outcome is positive, i.e. if no boundary condition has been encountered, then $r(\breve{x}_i) = true$. Else $r(\breve{x}_i) = false$. Then:

$$\mathcal{S}_i = \{\breve{y}_{ij} | r(\breve{x}_i) = true\} \tag{40}$$

Then $\mathcal{S}_{\breve{T}_3} = \emptyset$.

When applying the chain rule for all equations in set $\mathcal{C}'$, it turns out that $\breve{Q}_1$ is the only equation for which the chain rule is not stopped, in this case because $\breve{Q}_1$ does not involve any intermediate variable. As $\breve{Q}_1$ does not involve any intermediate variable, $\mathcal{S}_{\breve{Q}_1} = \emptyset$.

Set $\mathcal{S}$ is the union of all sets $\mathcal{S}_i$:

$$\mathcal{S} = \cup\, \mathcal{S}_i \tag{41}$$

Then $\mathcal{S} = \emptyset$.

Set $\mathcal{C}$ contains all equations $\breve{x}_i$ of $\mathcal{C}'$ whose associated set $\mathcal{S}_i$ has been completed without stopping the chain rule. $\breve{Q}_1$ is the only equation that complies with this rule, thus

$$\mathcal{C} = \{\breve{Q}_1\}$$

which corresponds to the mass balance equation.

The variables of interest that can be reconciled are the variables of interest involved in set $\mathcal{C}$ or set $\mathcal{S}$:

$$\bar{\bar{x}} = \{x_i \in x | x_i \in \mathcal{C} \cup \mathcal{S}\} \tag{42}$$

where $x_i \in \mathcal{C} \cup \mathcal{S}$ means that x_i is involved in at least one equation of $\mathcal{C} \cup \mathcal{S}$. Then $\bar{\bar{x}} = \{Q_1, Q_2, Q_3\}$. This means that all variables of interest in $x - \bar{\bar{x}}$ cannot be reconciled:

$$x - \bar{\bar{x}} = \{P_1, P_2, P_3, T_1, T_2, T_3, T\}$$

A larger set $\bar{\bar{x}}$, and thus a higher redundancy level, should be achievable as the model contains momentum and energy balance equations which are appropriate to reconcile pressures and temperatures. Therefore, the algorithm should carry on in order to get into $\bar{\bar{x}}$ as many variables of interest as possible.

To go forward, the idea is to replace in model $\mathcal{M}$ the boundary condition $b(\breve{x}_i)$ that made the chain rule fail for variable of interest x_i by the variable of interest x_i itself. The exact rule is to replace the equation $\overparen{b(\breve{x}_\imath)}$ that computes $b(\breve{x}_i)$ by equation $(x_i = 0)$. Applying this rule to $\breve{T}_3$ amounts to replacing $\breve{h}_{1,bc}$ by equation $(T_3 = 0)$. Applying this rule to all variables x_i such that $r(\breve{x}_i) = false$ yields a new model $\mathcal{M}_1$:

$$\begin{aligned} \mathcal{M}_1 = \mathcal{M} - \left\{\overparen{b(\breve{x}_\imath)} \middle| r(\breve{x}_i) = false\right\} \\ + \{(x_i = 0) | r(\breve{x}_i) = false\} \end{aligned} \tag{43}$$

For the example

$$\mathcal{M}_1 = \mathcal{M} - \left\{h_{1,bc}, Q_{3,bc,} Q_{2,bc,} P_{1,bc}\right\}$$
$$+\{(T_3 = 0), (P_3 = 0), (Q_3 = 0), (Q_2 = 0)\}$$

The BLT for $\mathcal{M}_1$ is given in Table 2.

Table 2. BLT of $\mathcal{M}_1$.

Variable	*Equation*	
$h_{1,bc}$	$h_{1,bc} = 100000.0$	binding
$Q_{3,bc}$	$Q_{3,bc} = 2.0$	binding
$Q_{2,bc}$	$Q_{2,bc} = 1.0$	binding
$P_{1,bc}$	$P_{1,bc} = 300000.0$	binding
Q_1	$0 = Q_1 - Q_2 - Q_3$	
Q_2	$Q_2 = 0$	binding
Q_3	$Q_3 = 0$	binding
P	$P = P_{1,r}$	
P_1	$P_1 = (P_{1,l} + P_{1,r})/2$	
P_2	$P_2 = (P_{2,l} + P_{2,r})/2$	
P_3	$P_3 = (P_{3,l} + P_{3,r})/2$	
$P_{1,l}$	$P_{1,l} = P_{1,bc}$	
$P_{1,r}$	$P_{1,l} - P_{1,r} = k_1 . Q_1^2$	
$P_{2,l}$	$P = P_{2,l}$	
$P_{2,r}$	$P_{2,l} - P_{2,r} = k_2 . Q_2^2$	
$P_{3,l}$	$P = P_{3,l}$	
$P_{3,r}$	$P_{3,l} - P_{3,r} = k_3 . Q_3^2$	
h	$h = c_p \cdot T + b \cdot P$	
h_1	$0 = h_1 \cdot Q_1 - h_2 \cdot Q_2 - h_3 \cdot Q_3 + W$	
h_2	$h_2 = h$	
h_3	$h_3 = h$	
T	$T = 0$	binding
T_1	$h_1 = c_p \cdot T_1 + b \cdot P_1$	
T_2	$h_2 = c_p \cdot T_2 + b \cdot P_2$	
T_3	$T_3 = 0$	binding

We now reapply the extraction algorithm to $\mathcal{M}_1$.

$$\mathcal{M}_1' = \mathcal{M}_1 - \left\{h_{1,bc}, Q_{3,bc,} Q_{2,bc,} P_{1,bc}, Q_2, Q_3, T, T_3\right\}$$
$$\mathcal{C}_1' = \{\breve{x}_i\} \cap \mathcal{M}_1' = \left\{\breve{Q}_1, \breve{P}_1, \breve{P}_2, \breve{P}_3, \breve{T}_1, \breve{T}_2\right\}.$$

Notice now that $\mathcal{C}_1'$ is smaller than $\mathcal{C}'$ by 4 units. Extracting set $\mathcal{S}$ yields:

$$\breve{Q}_1 \to success$$
$$\breve{P}_1 \to \breve{P}_{1,l} \to \breve{P}_{1,r} \to \breve{P} \to \breve{P}_{3,l} \to \breve{P}_{3,r} \to success$$
$$\breve{P}_2 \to \breve{P}_{2,l} \to \breve{P} \to \breve{P}_{3,l} \to \breve{P}_{3,r} \to \breve{P}_{2,r} \to success$$
$$\breve{P}_3 \to \breve{h}_3 \to \breve{h} \to \breve{P} \to \breve{P}_{3,l} \to \breve{P}_{3,r} \to success$$
$$\breve{T}_1 \to \breve{h}_1 \to \breve{h}_3 \to \breve{h} \to \breve{P} \to \breve{P}_{3,l} \to \breve{P}_{3,r} \to \breve{h}_2 \to success$$
$$\breve{T}_2 \to \breve{h}_2 \to \breve{h} \to \breve{P} \to \breve{P}_{3,l} \to \breve{P}_{3,r} \to success$$

Then:

$$\mathcal{C} = \left\{\breve{Q}_1, \breve{P}_1, \breve{P}_2, \breve{P}_3, \breve{T}_1, \breve{T}_2\right\}$$

and:

$$\mathcal{S}_{\breve{Q}_1} = \emptyset$$
$$\mathcal{S}_{\breve{P}_1} = \left\{\breve{P}_{1,l}, \breve{P}_{1,r}, \breve{P}, \breve{P}_{3,l}, \breve{P}_{3,r}\right\}$$
$$\mathcal{S}_{\breve{P}_2} = \left\{\breve{P}_{2,l}, \breve{P}, \breve{P}_{3,l}, \breve{P}_{3,r}, \breve{P}_{2,r}\right\}$$
$$\mathcal{S}_{\breve{P}_3} = \left\{\breve{h}_3, \breve{h}, \breve{P}, \breve{P}_{3,l}, \breve{P}_{3,r}\right\}$$
$$\mathcal{S}_{\breve{T}_1} = \left\{\breve{h}_1, \breve{h}_3, \breve{h}, \breve{P}, \breve{P}_{3,l}, \breve{P}_{3,r}, \breve{h}_2\right\}$$
$$\mathcal{S}_{\breve{T}_2} = \left\{\breve{h}_2, \breve{h}, \breve{P}, \breve{P}_{3,l}, \breve{P}_{3,r}\right\}$$

$$\begin{aligned} \mathcal{S} &= \mathcal{S}_{\breve{Q}_1} \cup \mathcal{S}_{\breve{P}_1} \cup \mathcal{S}_{\breve{P}_2} \cup \mathcal{S}_{\breve{P}_3} \cup \mathcal{S}_{\breve{T}_1} \cup \mathcal{S}_{\breve{T}_2} \\ &= \left\{\breve{P}_{1,l}, \breve{P}_{1,r}, \breve{P}, \breve{P}_{3,l}, \breve{P}_{3,r}, \breve{P}_{2,l}, \breve{P}_{2,r}, \breve{h}_3, \breve{h}, \breve{h}_1, \breve{h}_2\right\} \end{aligned}$$

Notice that the original model has a 21-equation algebraic system to be solved, whereas the extracted system for data reconciliation has only an 11-equation algebraic system to be solved (in fact two separate 11-equation algebraic systems, one for solving set $\mathcal{S}$ and the other for computing the Jacobian matrix of set $\mathcal{C}$, cf. resp. Equations (36) and (38)).

The variables of interest that can be reconciled are those who appear in set $\mathcal{C}$ or in set $\mathcal{S}$. All variables of interest appear in set $\mathcal{C}$ or in set $\mathcal{S}$, therefore all variables of interest can be reconciled:

$$\bar{\bar{x}} = \left\{\bar{\bar{Q}}_1, \bar{\bar{Q}}_2, \bar{\bar{Q}}_3, \bar{\bar{P}}_1, \bar{\bar{P}}_2, \bar{\bar{P}}_3, \bar{\bar{T}}_1, \bar{\bar{T}}_2, \bar{\bar{T}}_3, \bar{\bar{T}}\right\}$$

The redundancy level is 6 and the size of the algebraic system is divided by approximately 2. The extraction algorithm is completed■

If the set of variables of interest is

$$x = \{Q_1, Q_2, Q_3\},$$

then:

$$\mathcal{C} = \left\{\breve{Q}_1\right\}$$
$$\mathcal{S} = \emptyset$$
$$\bar{\bar{x}} = \left\{\bar{\bar{Q}}_1, \bar{\bar{Q}}_2, \bar{\bar{Q}}_3\right\}$$

All variables of interest can be reconciled, the redundancy level is 1 and there is no algebraic system to be solved.

If the set of variables of interest is

$$x = \{P_1, P_2, P_3\},$$

then:

$$\mathcal{C} = \emptyset$$
$$\mathcal{S} = \emptyset$$
$$\bar{\bar{x}} = \emptyset$$

No variable of interest can be reconciled. This is because there are too many boundary conditions related to the variables of interest. The data reconciliation problem is thus ill-posed.

3.7 Numerical Results

The inputs for the numerical computations are the measured values $\hat{x}_i$, the weights or half-width confidence intervals $w_{\hat{x},i}$ and the correlation matrix coefficients $r_{\hat{x},i,j}$.

In the following examples, the correlation matrix coefficients are equal to zero: $r_{\hat{x},i,j} = 0$ for $i \neq j$. Then only the measured values and weights are provided.

In the computation of Table 3 and Table 4, the values for the parameters are $k_1 = k_2 = k_3 = 1\ \mathrm{bar.kg^{-2}.s^{-2}}$, $c_p = 4.2\ \mathrm{kJ.kg^{-1}.°C^{-1}}$, $b = 0.19\ \mathrm{kJ.bar^{-1}}$ and $W = 1\ \mathrm{MW}$. One can verify that the reconciled weights are smaller than the measured weights.

Table 3. Inputs for the Splitter

Variable	*Measured value*	*Weight*
Q_1	2.5 kg/s	0.196
Q_2	1.15 kg/s	0.196
Q_3	1.25 kg/s	0.196
P_1	6.1 bar	0.392
P_2	2.55 bar	0.392
P_3	2.45 bar	0.392
T	114 °C	1.96
T_1	19 °C	1.96
T_2	113 °C	1.91
T_3	115 °C	1.91

Table 4. Reconciled values for the Splitter

Variable	*Reconciled value*	*Reconciled weight*	*Individual test*
Q_1	2.49413 kg/s	0.0521606	*true*
Q_2	1.20022 kg/s	0.120607	*true*
Q_3	1.2939 kg/s	0.120293	*true*
P_1	6.29266 bar	0.250971	*true*
P_2	2.46206 bar	0.274044	*true*
P_3	2.34524 bar	0.276126	*true*
T	114.124 °C	1.08222	*true*
T_1	18.5211 °C	1.7887	*true*
T_2	114.157 °C	1.08159	*true*
T_3	114.162 °C	1.08158	*true*

Listing 2. Tagging the boundary conditions

```
parameter Real Q0=100 "Fluid mass flow rate"
annotation(__OpenModelica_BoundaryCondition =
true);
  parameter Real h0=100000 "Fluid specific
enthalpy"
annotation(__OpenModelica_BoundaryCondition =
true);
```

In the computation of Table 5, only the mass flow rates are reconciled with the measured values of Table 3. One can verify that the reconciled weights are smaller than the measured weights but are larger than the values obtained when reconciling the mass flow rates with the pressures and the temperatures. This is consistent with the fact that more information leads to better accuracy.

Table 5. Reconciled values for the Splitter

Variable	*Reconciled value*	*Reconciled weight*	*Individual test*
Q_1	2.46667 kg/s	0.160033	*true*
Q_2	1.18333 kg/s	0.160033	*true*
Q_3	1.28333 kg/s	0.160033	*true*

In both calculations, the χ^2-test of Condition (22) is satisfied.

3.8 Interface with Modelica in OpenModelica

OpenModelica is an open source tool for the modelling and simulation of Modelica models (Fritzson et al., 2020). The data reconciliation interface with Modelica newly implemented in OpenModelica aims at giving the possibility to perform data reconciliation on a validated Modelica model without having to modify the model.

To perform data reconciliation, three to four actions are necessary:

1. Tag the boundary conditions.
2. Tag the variables of interest.
3. Provide the measured values and weights.
4. If necessary, provide the correlation coefficients.

In Modelica libraries, boundary conditions are most often to be found in specialized components such as mass flow rate, pressure and temperature sources and sinks. The tagging of boundary conditions is therefore permanent

Listing 3. Tagging the variables of interest

```
model TSP_Splitter_DR
  TSP_Splitter splitter(
     pipe1(Q(uncertain = Uncertainty.refine)),
     pipe2(Q(uncertain = Uncertainty.refine)),
     pipe3(Q(uncertain = Uncertainty.refine)),
     pipe1(Pm(uncertain = Uncertainty.refine)),
     pipe2(Pm(uncertain = Uncertainty.refine)),
     pipe3(Pm(uncertain = Uncertainty.refine)),
     volume(T(uncertain = Uncertainty.refine)),
     pipe1(T(uncertain = Uncertainty.refine)),
     pipe2(T(uncertain = Uncertainty.refine)),
     pipe3(T(uncertain = Uncertainty.refine)));
equation
end TSP_Splitter_DR;
```

Listing 4. Csv file for the measured values

```
Variable name; Measured value; Weight
splitter.pipe1.Q; 2.50; 0.196
splitter.pipe2.Q; 1.15; 0.196
splitter.pipe3.Q; 1.25; 0.196
splitter.pipe1.Pm; 6.1e5; 0.392e5
splitter.pipe2.Pm; 2.55e5; 0.392e5
splitter.pipe3.Pm; 2.45e5; 0.392e5
splitter.volume.T; 387; 1.96
splitter.pipe1.T; 292; 1.96
splitter.pipe2.T; 386; 1.91
splitter.pipe3.T; 388; 1.91
```

and should not interfere with the usual simulation activities (i.e. DAE integration from initial conditions). It also should not prevent the components to be used with tools that do not support data reconciliation. Therefore, the tagging of boundary conditions is done with a special annotation. In Listing 2, two boundary conditions are tagged: the fluid mass flow rate and the fluid specific enthalpy.

The variables of interest are tagged by the user in a different way for each data reconciliation problem. The tags should not modify the original Modelica model, and appropriate checks must be done by the tool to verify that the tags refer to existing and eligible variables. Therefore, the tagging of variables of interest is done with a special modifier. In Listing 3, the name of the original model of the splitter is TSP_Splitter. One instance of TSP_Splitter is placed in model TSP_Splitter_DR that is especially created to perform data reconciliation on the model TSP_Splitter. The variables of interest in model TSP_Splitter are tagged from model TSP_Splitter_DR using the `uncertain` modifier. Checks are performed by the tool to ensure that the tagged variables are eligible to be tagged as variables of interest. Thus, the original model is not modified and can be used in any Modelica tool. The embedding model can however only be used in tools supporting data reconciliation.

The measured values and weights are provided in a csv file with 3 columns, cf. Listing 4 where the values are given in SI units.

The correlation matrix is provided in an optional separate csv file. No correlation matrix file is given for the example.

3.9 Data Reconciliation with ThermoSysPro

ThermoSysPro is a Modelica library for the modelling and simulation of power plants and energy systems at large (El Hefni and Bouskela, 2019).

The ThermoSysPro model of the Splitter is shown in Figure 3. It is equivalent to the model proposed in Section 3.3, the equations being dispatched in the following specialized components: `SourceP` (pressure source), `SinkQ` (mass flow rate sink), `SingularPressureLoss` (pipes), `VolumeBTh` (splitter volume), `HeatSource` (thermal power). Therefore, for instance the variable `splitter.volume.T` in Listing 4 is the temperature T inside the splitter volume. The Modelica model has a total of 121 variables and 121 equations.

The model components used in the Splitter model are modified as follows.

1. `SourceP` and `SinkQ`: The boundary conditions are tagged as shown in Listing 2.
2. `VolumeBTh`: The temperature is computed from the specific enthalpy with equation $\breve{T}$ in Table 1.
3. `SingularPressureLoss`: The pressure loss is computed from the mass flow rate with equation $\breve{P}_{i,r}$ in Table 1 for $i = 1, 2, 3$. The temperature is computed from the specific enthalpy with equation $\breve{T}_i$ in Table 1 for $i = 1, 2, 3$. The mass flow reversal equation in each pipe which according to the upwind scheme should be (El Hefni and Bouskela, 2019, Eq. 4.114)

$$h_i = \begin{cases} h_{i,l} & \text{if } Q_i \geq 0 \\ h_{i,r} & \text{if } Q_i < 0 \end{cases}$$

is replaced by

$$h_i = h_{i,l}$$

under the assumption that mass flow rates are positive, in order to avoid dependencies with boundary conditions $h_{2,r}$ and $h_{3,r}$ so that Condition (35) can be satisfied and all temperatures can be reconciled. This replacement was performed manually but could be done automatically as the sign of Q_i is fixed and known beforehand.

Modifications in points 2 and 3 above are made to avoid numerical difficulties when solving the algebraic equations.

The extracted model has 10 variables to be reconciled, 6 auxiliary conditions and 41 intermediate equations. Therefore, the size of the algebraic system to be solved is divided by 3.

The results of the data reconciliation computation with ThermoSysPro are shown in Table 6. They are different from the results obtained without ThermoSysPro displayed in Table 4 because the two splitter models are equivalent, but not identical, however results stay within their confidence intervals.

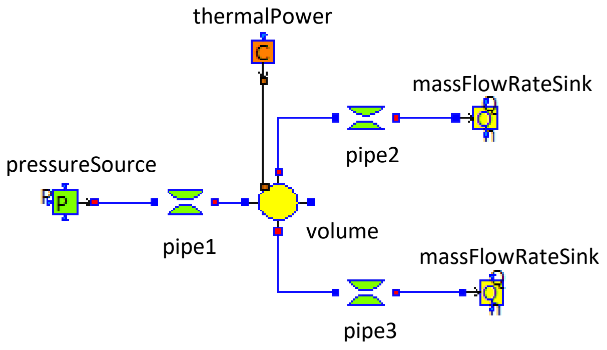


Figure 3. ThermoSysPro model of the splitter

Table 6. Reconciled values with ThermoSysPro

Variable	*Reconciled value*	*Reconciled weight*	*Individual test*
Q_1	2.49327 kg/s	0.0521266	*true*
Q_2	1.19978 kg/s	0.120619	*true*
Q_3	1.29349 kg/s	0.120306	*true*
P_1	6.29090 bar	0.250917	*true*
P_2	2.46296 bar	0.274003	*true*
P_3	2.34614 bar	0.276086	*true*
T	113.987 °C	1.08228	*true*
T_1	18.351 °C	1.78902	*true*
T_2	114.019 °C	1.08165	*true*
T_3	114.025 °C	1.08163	*true*

4 Conclusion and Future Work

Performing data reconciliation on a Modelica model requires to extract the auxiliary conditions that constrain the variables of interest from the Modelica model and remove the boundary conditions which are related to the variables of interest. This is why, contrary to data assimilation (Corona Mesa-Moles et al., 2019), data reconciliation cannot be performed on a Modelica model in a black box manner.

An extraction algorithm has been presented and applied on a small example with 121 equations built with the ThermoSysPro library. The extracted model has 10 variables to be reconciled, 6 auxiliary conditions and 41 intermediate equations, which shows that the extraction algorithm reduces significantly the size of the algebraic system to be solved (here by a factor of 3). Some modifications were required in the ThermoSysPro library to perform the numerical computations. They mainly consisted in simplifying the equations that compute the fluid properties and those that compute the specific enthalpy according to the upwind scheme. These simplifications around the measured values are acceptable because data reconciliation assumes that the differences between the measured and reconciled values are small.

This work demonstrates that Modelica can be used for other purposes than for initial value problems, and that the knowledge embedded in existing models can be utilized also for data processing.

Future work will consist in verifying that the method presented in the paper is applicable to larger models. A preliminary experiment was conducted on the model of a secondary side of a nuclear power plant to compute the reactor nominal power. The model has 2002 variables and equations and 26 variables to be reconciled. The extraction algorithm produced 23 auxiliary conditions and 553 intermediate equations. This shows that a higher level of redundancy and algebraic system reduction can be achieved on larger models, making paradoxically data reconciliation more efficient to perform on larger Modelica models than on smaller ones.

Future work will also consist in computing the values of the boundary conditions from the reconciled values, have the new method certified by the VDI 2048 standard committee and integrate the new modifier for tagging the variables of interest into the Modelica standard.

Acknowledgements

This work was partially supported by DGE (France) and Vinnova (Sweden) in the scope of the ITEA 3 OPENCPS project.

References

Bai, S. and Thibault, J. (2010). “*Dynamic Data Reconciliation*”. VDM Verlag. ISBN: 978-3-639-21779-7.

Bedjaoui, N., Litrico, X., Koenig, D., Ribot-Bruno, J. and Malaterre, P.O. (2008). “*Static and Dynamic Data Reconciliation for an Irrigation Canal*”. Journal of Irrigation and Drainage Engineering, 134: 778-787.

Belsim (2021). “*VALI Software Suite*”. URL: https://belsim.com/vali-software/ (visited on 2021-04-14).

Corona Mesa-Moles, L., Argaud, J.P., Jardin, A., Benssy, A. and Dong, Y. (2019). “*Robust Calibration of Complex ThermosysPro Models using Data Assimilation Techniques: Application on the Secondary System of a Pressurized Water Reactor*”. Linköping Electronic Conference Proceedings 157:56, s. 8.

Dutfoy A., Dutka-Malen I., Lebrun R. et al. (2009). “*OpenTURNS, an Open Source Initiative to Treat Uncertainties, Risks'N Statistics in a Structured Industrial Approach*”. In: *41èmes Journées de Statistique, SFdS, Bordeaux*.

El Hefni, B. and Bouskela, D. (2019). “*Modeling and Simulation of Thermal Power Plants with ThermoSysPro*”. Springer. ISBN: 978-3-030-05104-4.

Fritzson, P., Pop, A., Abdelhak, K., Asghar, A. et al. (2020). “*The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development*”. Modeling, Identification and Control. 2020;41(4):241-295.

Langenstein, M., Jansky, J., Laipple, B., Grauf, E., Schak, H., & Eitschberger, H. (2004). “*Finding megawatts in nuclear power plants with data reconciliation*”. In: *Proceedings of the 12th International Conference on Nuclear Engineering*. Vol. 2. American Society of Mechanical Engineers – ASME.

Modelica Association (2021). “*Modelica – A Unified ObjectOriented Language for Systems Modeling. Language Specification Version 3.5*”. Tech. rep. Linköping: Modelica Association. URL: https://www.modelica.org/documents/MLS.pdf.

VDI - Verein Deutscher Ingenieure (2017). VDI 2048 – Blatt 1 “*Control and quality improvement of process data and their uncertainties by means of correction calculation for operation and acceptance tests, VDI-Handbuch Energietechnik*”, ICS: 17.020, 27.010.

DOI 10.3384/ecp21181453