Cheng, Hui; Nikus, Mats; Jämsä-Jounela, Sirkka-Liisa


Published in:
Journal of Process Control

DOI:
10.1016/j.jprocont.2007.12.003

Published: 01/01/2008

Document Version
Peer reviewed version

Please cite the original version:
Fault diagnosis of the paper machine short circulation process using novel dynamic causal digraph reasoning

Hui Cheng*, Mats Nikus, Sirkka-Liisa Jämsä-Jounela

Aalto University, Laboratory of Process Control and Automation, Kemistintie 1, FI-02150, Finland

Abstract

This paper presents a novel dynamic causal digraph reasoning method for fault diagnosis and its application to the short circulation process of a paper machine. In order to improve the fault detection ability of the original causal digraph method, a residual modification approach that takes into account the direction of different fault effects is presented. An improvement of the isolation capability of the original method, an inference mechanism between the arcs of the graph, is also proposed to locate process faults on the arcs. The results from the application show that the proposed method, compared to the conventional method, is able to detect the correct nodes and to identify the responsible arcs when the system is affected by a process fault.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Causal digraph; Paper machine; CUSUM

1. Introduction

Research in the field of fault diagnosis has been very active since the 1970s. In order to meet the demands from industry concerning quality, efficiency and safety, numerous fault diagnosis methods have been developed. Of these, causal model based methods have attracted considerable attention since they were first developed by Iri et al. [5]. As a modelling method, the causal model is able to describe the system behaviour in terms of cause-effect relationships between the entities of a system [1], which is represented by a directed graph. The causal models can be used for many different purposes, such as simulation [7] and fault diagnosis [8].

The fault diagnosis method based on causal models was created by integrating causal theory with graph theory. Due to the broad definition of a cause–effect relationship, dozens of fault diagnosis methods have been developed on the basis of different interpretations of the relationship. In the first methods, the arcs in the digraph were labelled using a pure qualitative model, either by a positive or a negative sign, to explain the cause–effect relationship, while the nodes of the digraph depicted the levels of the variables as either high or low [5]. The forward inference mechanism, which is an important part of the SDG (signed digraph) method, was developed by adapting graph searching techniques for fault diagnosis purposes [14]. The introduction of fuzzy logic into the nodes and arcs by Shih and Lee [10,11] increased the qualitative information included in the causal model, leading to a decrease in the number of spurious fault diagnosis results, even though the inference mechanism as such was not apparently improved. By using piece-wise linear transfer functions (QTF), temporal information was introduced into the causal model for simulation and supervision purposes [7]. The next step was to use difference-algebraic equations to describe the cause–effect relationships in the dynamic causal diagnostic reasoning method. This enabled the development of the inverse inference mechanism. The inference mechanism simplifies the fault location step by using consistency tests for the local model, and by tracking the fault origin nodes in the MISO structure of the causal model [8]. One application of this method on the paper machine simulator with
the discrete state space models describing the cause–effect relationships was reported by Cheng et al. [15]. The latest development of the dynamic causal reasoning fault diagnosis method is the introduction of the set-membership approach, which considers the uncertainty in the modelling and fault detection phases [1].

As reported by Montmain and Gentil [8] and Fagarasan et al. [1], the dynamic causal digraph method is able to manage different types of fault: sensor, actuator and process faults. However, the ability of the dynamic causal digraph method to detect faults and handle process faults is limited. Firstly, the detection of the global residual may vanish in some cases owing to cancellation of the different fault effects, thus giving an incorrect detection result. Secondly, the method assumes that a change in the variable is the primary fault, which is not true for a process fault. In order to compensate for these problems, a method to separate different fault effects in residual generation is presented in this paper. Furthermore, an inference mechanism between arcs has been developed by the authors to locate the process fault on the arcs.

The paper is organized as follows. In Section 2, the dynamic causal reasoning method is described briefly. Section 3 presents the novel dynamic causal reasoning method, including the CUSUM method, separation of fault effects in residuals, and the inference mechanism between arcs for fault location on the arcs. In Section 4, the short circulation process of a paper machine and the simulation environment are introduced. Three different fault scenarios, description of the experiment and the fault diagnosis results for each fault scenario are given in Section 5, followed by the conclusions in Section 6.

2. Fault diagnosis using dynamic causal digraph

The dynamic causal digraph method performs the fault diagnosis in two steps: fault detection and fault isolation. Fault detection is carried out by the generation and detection of the simulation errors of the dynamic causal digraph model. Fault isolation is simplified by recursive calculation of the suspicion degrees of nodes in a detection set, which is formed of the variables detected by simulation errors, using fuzzy logic operations on the simulation and prediction errors. When the suspicion degree of the nodes is beyond a predefined threshold, then the corresponding nodes are suspected and used to form the fault propagation path. Even though the whole derivation of the equations for the algorithm is illustrated by difference equation models, the method can be extended to the general linear and nonlinear models. Furthermore, an application of the method in a nuclear power plant reported by Montmain and Gentil [8] clearly demonstrated the advantages of the method.

However, there are several drawbacks originating from the basic assumptions of the dynamic causal digraph method. Firstly, fault detection based on the simulation error $e_i$ given in Eq. (1), is not valid in some cases:

$$e_i = \sum_{j \in P_i} q^{-d_{ji}}A_{ji}(q^{-1})e_j + E_i(q^{-1})f_i(k)$$

$P_i$ in Eq. (1) is the set of all subscripts $j$ of the predecessors of node $i$ in the digraph, $e_j$ is the simulation errors from the predecessors of node $i$, and $f_i(k)$ is the local fault. The model from node $i$ to node $j$ is denoted as $\frac{q^{-d_{ij}}A_{ij}(q^{-1})}{A_i(q^{-1})}$, while the model for the local fault to the residual is $\frac{E_i(q^{-1})}{A_i(q^{-1})}$. Hence the above equation clearly shows that the simulation error $e_i$ for the node $i$ is the sum of the propagated fault effects from the predecessor nodes ($e_j$) and the local fault effect ($f_i(k)$). Therefore, in the case where different fault effects are in a different direction, the simulation error may vanish or be too small to be detected. For the same reason, canceling out of different fault effects can also occur in the calculation of the prediction errors.

It is also implied by Eq. (1) that the fault isolation based on the suspicion degree calculation is no longer valid even though the simulation error is detectable. The suspicion degree is calculated by comparing the simulation and prediction errors, with the implicit assumption that the different fault effects have the same sign and direction. When the assumption is not satisfied, the simulation error can be smaller than the prediction error which, in turn, results in an incorrect suspicion degree. Furthermore, because the ratio of the prediction error to the simulation error is used to calculate the suspicion degree, which is sensitive to the noise, singularity problems may occur when the signals are noisy. Finally, a threshold for the suspicion degree has to be set up in order to map the residual signal to the set $[0, 1]$, even though the residual was mapped into a real number between $[0, 1]$ by the fuzzy sets at the beginning of the fault isolation.

The last problem we consider here is that all the causal model based fault diagnosis methods assume that a primary fault is a change in a variable (a node in the digraph) rather than a change in the consistency between the variables (an arc in the digraph). In industrial applications, however, it is usually required to know the corresponding faulty process component. This requirement can be met in the case of measurement faults, since the nodes in the digraph model represent the sensor of the process, while faulty nodes alone cannot provide enough information to identify the faulty process components corresponding to the arcs in the digraph. One important topic is therefore to locate the responsible arc.

3. Novel dynamic causal digraph method

A novel dynamic causal digraph method designed to overcome the problems described in Section 2 is proposed in this section. The proposed method performs the fault detection and isolation in five steps as follows:

1. Generate the global (GR) and local residuals (LR).
2. Detect a possible abnormality in the residual signals using the CUSUM method.
3. Modify the residuals by considering fault effects with different directions and apply the CUSUM method to the modified residuals in order to form the detection set.
4. For the variables in the detection set, locate the primary fault and identify its nature by means of the fault isolation and nature rules.
5. In case of a process fault, an additional inference step between arcs is performed in order to locate the fault on the responsible arc(s).

Discrete-time state space models are used to describe the cause–effect relationships between variables in the causal models.

3.1. Residual generation with the dynamic causal model

The dynamic causal digraph produces two kinds of residual to be used in fault detection and isolation: global and local residuals. The global residual is produced for the purpose of fault detection by the difference between the measurement and the global propagation value:

\[ GR(Y) = Y(k) - \bar{Y}(k) \]  

(2)

where \( Y(k) \) is the measurement and \( \bar{Y}(k) \) is the global propagation value obtained by

\[
\bar{X}(k+1) = A\bar{X}(k) + B\bar{U}(k) \\
\bar{Y}(k) = C\bar{X}(k)
\]

(3)

where \( \bar{U}(k) = [\bar{U}_1(k), \ldots, \bar{U}_n(k)]^T \) is the lagged global propagation value from the predecessors in the graph model, \( \bar{X}(0) = x_0 \) is the initial state, and \( n \) denotes the number of inputs for variable \( Y \).

The local residuals are subcategorized into three types: individual local residuals (ILR), multiple local residuals (MLR) and total local residuals (TLR).

The individual local residual is produced by taking the difference between the measurement and the local propagation value with only one measured input, while all the others are propagation values from the parent nodes:

\[
X'(k+1) = AX'(k) + BU(m,k) \\
Y'(k) = CX'(k)
\]

(4)

\[
ILR_Y(m) = Y(k) - Y'(k)
\]

where \( U(m,k) = [\bar{U}_1(k), \ldots, U_m(k), \ldots, \bar{U}_n(k)]^T \), \( \bar{U}_i(k) \) is the propagated value from the predecessors, and \( i \neq m \), \( U_m(k) \) is the measurement for the parent node and \( X'(0) = x_0 \) is the initial state.

Similarly, the \( MLR_Y(P) \) is produced with input \( U(P,k) = [\bar{U}_1(k), \ldots, U_j(k), \ldots, \bar{U}_n(k)]^T \), \( j \in P \), \( P \) is the set of subscripts of the predecessors which use the measurement as an input. The \( TLR(Y) \) is produced with \( P = P_Y \), where \( P_Y \) is the set of subscripts of all the predecessors of \( Y \).

3.2. Residual evaluation with the CUSUM method

The CUSUM method is used to evaluate the generated residual signals with respect to its insensitivity to the noise and outliers in the measurements. Moreover, the CUSUM method is able to detect both positive and negative jumps in the mean of the residual signal. The direction of the change provides useful information for fault diagnosis.

The residual signals produced are mapped to the set \( \{0, 1,-1\} \) by the CUSUM method. In the faultless case, the residuals are assumed to be zero mean random signals, and in the case of a fault the mean value of the residual signal is changing in either a positive or a negative direction.

The CUSUM algorithm [4] for positive mean jumps is given by the following equations:

\[
\sum(k) = \sum(k-1) + \delta(k) - \mu_0 - \beta/2
\]

(5)

\[
\sum_{min}(k) = \min \left( \sum(k-1), \sum(k) \right)
\]

(6)

where \( \beta \) is a user-specified minimum detectable jump, \( \delta(k) \) is the residual signals, and \( \mu_0 \) is the mean value of the nominal signal. Whenever \( \sum(k) - \sum_{min}(k) > \lambda \), a jump has been detected. \( \beta \) and \( \lambda \) are design parameters, usually tuned according to the requirements for the false alarm and missed alarm rates. Parameter \( \lambda \) provides robustness to the fault detection but it also delays the detection. A similar algorithm for the detection of negative mean jumps can easily be obtained by modifying Eqs. (5) and (6).

In the rest of the paper, the result of the CUSUM calculation will be denoted as a function: \( CU() \).

3.3. Separation of fault effects in residuals

In order to manage the specific cases when \( GR \) and \( LR \) become too small to detect due to the cancellation of different fault effects, the fault effects with different directions are taken into account and separated. The proposed approach to perform this is given as follows:

1. Test whether detection of the residual vanishes because of the different fault effects, go to step 2.
2. Determine the fault effect which is opposite in direction to the residual in question.
3. Generate a new residual by excluding the effect of the fault found in step 2.

The calculation equations for the modified \( GR(GRM) \) are given as an example in the following. First a test is performed to determine whether detection of the GR vanishes with the propagated fault effect:

\[
(\text{Index}U \neq \phi) \land (|CU(MLR_Y(\text{Index}U))| == 1) \\
\land (|CU(GR(Y))| == 0) = True
\]

(7)

where \( \text{Index}U = \{ \|CU(GR(U_i))\| = 1, 1 \leq i \leq n \} \) is the set of subscripts of the input nodes having an alarm in the global
The symbol $\phi$ denotes the empty set; while the symbol $\land$ denotes the logic ‘AND’ operation and the equation with symbol $==$ returns a logic value (True/False).

If the condition in Eq. (7) is satisfied, then the fault effects from the propagation and the local fault, whose sum gives the $GR$ given in Eq. (1), cancel each other out and need to be separated on the basis of their directions. The propagated fault effects having opposite direction to $MLR_s(IndexU)$ needs to be identified from the following equation:

$$IndexUF = \begin{cases} \frac{CU(MLR_f(IndexU)) \cdot (CU(MLR_f(IndexU)))}{i} & -MLR_f(IndexU - \{i\}) > 0 \\ i \in IndexU \end{cases}$$

where $IndexUF$ is the set of subscripts of input nodes whose faulty effect needs to be excluded. In the above equation, $MLR_s(IndexU) - MLR_s(IndexU - \{i\})$, $i \in IndexU$ represents the propagated fault effect from the node $i$, while the $MLR_s(IndexU)$ are the local fault effects.

In the last step of the separation approach, the fault effects propagated from the nodes whose subscripts are in the set $IndexUF$ are excluded in GRM by the following equation:

$$GRM(Y) = Y(k) - \bar{Y}(k)$$

$$\bar{X}(k + 1) = X(k) + B\bar{U}(k) \quad \text{and}$$

$$\bar{U}(k) = [\bar{U}_1(k), \ldots, U_i(k), \ldots, \bar{U}_n(k)] \quad \forall i \in IndexUF$$

$$\bar{Y}(k) = C\bar{X}(k)$$

where the tilde signs are used to separate the variables from those in Eqs. (2) and (3), due to the modification in the input.

Using a similar procedure, the $ILR$ and $MLR$ are modified ($ILRM$ and $MLRM$) for cases when different fault effects cancel each other out. Subsequently, the $GRMs$ are evaluated by the CUSUM method for fault detection, and the detected nodes form the detection set. For the nodes in the detection set, the $ILRMs$ and $MLRMs$ are evaluated by the CUSUM method for the fault isolation described in the next section.

### 3.4. Fault isolation reasoning with rules

Fault isolation is performed recursively for the nodes in the detection set by using a set of rules. These isolation rules, developed by Montmain and Gentil [8], are converted into a table for the convenience of implementation, as shown in Table 1.

After fault isolation, the nature of the fault is identified by testing how the faults propagate to their children nodes. If the fault propagates through the digraph globally, it is identified as a process fault; otherwise it is defined as a local measurement fault. The rules for identifying the nature of the fault are given in Table 2.

#### 3.5. Locating the process fault on the arcs

In the case of a process fault, in addition to locating the fault on the variables, locating it on the arcs is also desirable. However, the MISO structure of the digraph causes problems by generating spurious results. If one specific node $Y$ with $n$ input arcs in the digraph is identified as the fault origin for one process fault, then the number of sets of suspected arcs which can explain the fault on the node is $2^n - 1$, $n \geq 1$. In the case where a large number of input arcs exist, or there are more than one fault origins identified, then the number of sets of suspected arcs is very high. In order to decrease the number of spurious results, a new inference mechanism between the arcs is proposed for evaluating the sets of suspected arcs.

The idea behind the proposed inference mechanism is to test the consistency between the sets of suspected arcs formed from fault origins and the knowledge of the output arcs from the same node. This knowledge is introduced into the digraph by a knowledge matrix, and a consistency test is performed by matrix manipulations. Only the sets of suspected arcs which are consistent with the knowledge matrix of the digraph are considered as possible results.

The knowledge matrix of the whole digraph is formed by the knowledge matrices of each node which has output arcs. The knowledge matrix of a node is derived from the process knowledge concerning its output arcs. Due to conservation constraints such as material or energy balances, an inconsistency between two variables will also usually

### Table 1

<table>
<thead>
<tr>
<th>$CU(\text{GR}(Y))$</th>
<th>$CU(\text{TLR}(Y))$</th>
<th>$CU(\text{ILR}(m))$</th>
<th>$CU(\text{ILR}(i))$</th>
<th>$CU(\text{MLRY}(P_1))$</th>
<th>$CU(\text{MLRY}(P_2))$</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1/−1</td>
<td>0</td>
<td>0</td>
<td>1/−1</td>
<td>0</td>
<td>1/−1</td>
<td>Fault propagates from the parent node $m$</td>
</tr>
<tr>
<td>1/−1</td>
<td>0</td>
<td>1/−1</td>
<td>1/−1</td>
<td>1/−1</td>
<td>0</td>
<td>Fault propagates from the nodes with subscript in $P_2$</td>
</tr>
<tr>
<td>1/−1</td>
<td>1/−1</td>
<td>1/−1</td>
<td>1/−1</td>
<td>1/−1</td>
<td>1/−1</td>
<td>Local fault on variable $Y$</td>
</tr>
</tbody>
</table>

*a* $\forall i \not= m, i \in P_y, m \in P_1, m \not= P_2, P_y$ is the set of subscripts of parents nodes of the node $Y$.  
*b* $\forall i, m, i \in P_y, m \in P_1, \forall P_1, P_2 \subseteq P_y$.

### Table 2

<table>
<thead>
<tr>
<th>$CU(\text{GR}(X))$</th>
<th>$CU(\text{TLR}(X))$</th>
<th>Fault nature</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/−1</td>
<td>1/−1</td>
<td>Local fault for that child node</td>
</tr>
<tr>
<td>1/−1</td>
<td>0</td>
<td>Process fault for the faulty node</td>
</tr>
<tr>
<td>0</td>
<td>1/−1</td>
<td>Measurement fault for the faulty node</td>
</tr>
</tbody>
</table>

*a* $X$ is the subscript of any child nodes of the node $Y$. 

---

result in inconsistencies between other variables when a fault occurs. In the digraph model, the output arcs from the same node usually have relationships between each other such that, whenever one of the output arcs is inconsistent, some of the other output arcs are inconsistent as well. In order to represent this type of knowledge for a node, \( U \), in a digraph, the matrix \( M_U \) is specified as

\[
M_U(i,j) = \begin{cases} 
1, & \text{if the inconsistent arc } (U,i) \text{ results in the inconsistency of } (U,j) \\
0, & \text{otherwise}
\end{cases}
\]

The dimension of the matrix \( M_U \) for node \( U \) is \( n \times n \), if \( n \) is the number of output arcs which a node has. From the definition, the matrix \( M_U \) is not necessarily symmetric since the knowledge of the output arcs is directional. The knowledge matrix \( M \) for all the arcs in the digraph model is obtained by merging the knowledge matrices of every individual node in the digraph model, given as

\[
M = \text{blockdiag}(M_1, M_2, \ldots, M_l)
\]

where \( l \) denotes the number of nodes that have output arcs in the digraph model. The knowledge matrix \( M \) has the dimension \( N_a \times N_a \), where \( N_a \) is the number of arcs in the digraph, and each row or column represents one arc in the digraph.

The consistency test between the sets of suspected arcs and the knowledge matrix \( M \) is performed by matrix manipulation. First, one row vector \( sv \) with length \( N_a \) is formed for one specific set of suspected arcs:

\[
sv(i) = \begin{cases} 
1, & \text{if } ARC(M,i) \in S, \ 1 \leq i \leq N_a \\
0, & \text{otherwise}
\end{cases}
\]

where \( ARC(M,i) \) gives the arc corresponding to the \( i \)-th row in the matrix \( M \). The set \( S \) of suspected arcs is evaluated by multiplying the row vector \( sv \) and the matrix \( M \) of the digraph. The set \( S \) is considered as a possible result only when the number of nonzero entries in the obtained row vector does not change when compared to the row vector \( sv \):

\[
NUM(sv) = NUM(sv \cdot M)
\]

where \( NUM(\cdot) \) gives the number of nonzero entries in the vector.

Two important properties can be derived from Eq. (13). Firstly, if there is no knowledge available about the output arcs, then the knowledge matrix \( M \) for the digraph will be the identity matrix, resulting in \( sv = sv \cdot M \). Thus, Eq. (13) will be satisfied for every set of suspected arcs and no spurious results can be removed from the possible results. Secondly, the knowledge matrix \( M \) usually consists of bidirectional knowledge, which means that if one faulty arc results in another faulty arc then the opposite is true. However, this is not always the case and a non-bidirectional knowledge matrix does exist. Moreover, the non-bidirectional knowledge matrix \( M \) produces more spurious results but also provides some robustness in the case that the process knowledge is not so certain. An example is given in Table 3 for the node \( U \) with two output arcs. For the non-bidirectional knowledge matrix \( M_U \) two sets of suspected arcs are accepted, while for the bidirectional knowledge matrix \( M_U \) only one set is accepted. On the other hand, the non-bidirectional knowledge matrix \( M_U \) gives robustness to the inference in the case that the available process knowledge has uncertainty. From Table 3 it can be seen clearly that the second element in the first row of matrix \( M_U \) is defined as 0, if the concerning knowledge is uncertain. Furthermore, all the suspected sets accepted by matrix \( M_U \) are also accepted by matrix \( M_U \), which means correct fault diagnosis result is not rejected. However, more spurious fault diagnosis results are obtained at the cost of robustness.

**Fig. 1** illustrates the operation of the proposed inference mechanism between arcs. The assumption is made that the nodes \( Y_2 \) and \( Y_3 \) are identified as fault origins for the process fault. The knowledge between the output arcs from the same node is first introduced into the digraph:

\[
M_U = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad M_U = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \Rightarrow M
\]

\[
= \text{blockdiag}(M_{U_1}, M_{U_2}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}
\]

The corresponding arcs of each row of matrix \( M \) are \( \langle U_1, Y_1 \rangle, \langle U_1, Y_2 \rangle, \langle U_2, Y_2 \rangle \) and \( \langle U_2, Y_3 \rangle \). Due to the fact that node \( Y_2 \) has two input arcs, and node \( Y_3 \) has one input arc, there are three sets of suspected arcs, \{\langle U_1, Y_2 \rangle, \langle U_2, Y_2 \rangle, \langle U_2, Y_3 \rangle\} \{\langle U_1, Y_2 \rangle, \langle U_2, Y_2 \rangle\} and \{\langle U_2, Y_2 \rangle, \langle U_2, Y_3 \rangle\}, which are possible results. For the first set, the

<table>
<thead>
<tr>
<th>Suspected sets</th>
<th>( M_U )</th>
<th>( M_U )</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1, 1}</td>
<td>Accepted</td>
<td>Accepted</td>
</tr>
<tr>
<td>{0, 1}</td>
<td>Rejected</td>
<td>Rejected</td>
</tr>
<tr>
<td>{1, 0}</td>
<td>Rejected</td>
<td>Accepted</td>
</tr>
</tbody>
</table>

Table 3: Comparison between a bidirectional and non-bidirectional knowledge matrix
sv vector is formed as [0 1 1 1] according to Eq. (12), and the result of the multiplication of sv and M is
\[
NUM(sv) = 3 = NUM(sv \cdot M) = NUM([[1 1 2 2]]) = 4
\]
(15)

The result vector has four nonzero entries while the sv vector only has three, which implies that the first set of suspected arcs cannot both explain the identified fault origins and be consistent with the knowledge between output arcs. For the same reason, the second set of arcs is excluded from the possible results. For the last set, sv vector is [0 0 1 1] and the result of the multiplication of sv and M is:
\[
NUM(sv) = 2 = NUM(sv \cdot M) = NUM([[0 0 2 2]])
\]
(16)

The result vector has two nonzero entries, and so has the vector sv, which indicates that the third set is the possible result. It can be seen that the proposed inference mechanism between arcs is able to decrease the number of spurious sets of suspected arcs.

4. Testing environment

The proposed novel causal digraph method was tested on the paper machine short circulation process, which was simulated using the Advanced Process Simulation Environment (APROS) developed by VTT (Technical Research Centre of Finland). A description of the short circulation process and the simulation environment are given in this section.

4.1. Description of the short circulation process

The short circulation is a crucial part of the papermaking process, with several important functions. The dilution of the fiber suspension entering the process to a suitable consistency for the headbox takes place in the short circulation, in a mixing process where low-consistency water from the wire pit is mixed with high-consistency stock. The second important task of the short circulation is the removal of impurities and air. This task is performed in the hydrocyclones, machine screens and the so-called deculator. The short circulation also improves the economy of the process because the valuable fibers and filler materials that pass through the wire are recycled. Being the intermediate process between the stock preparation and a former, the short circulation process is also very important for paper quality control, because the most important paper qualities such as basis weight, ash rate and fiber orientation are controlled in the short circulation process.

The short circulation process starts after the machine chest. The machine chest is usually followed by a thick stock pump and a basic weight valve, which is used for basis weight control. The thick stock from the machine chest is pumped to the wire pit and mixed with white water and filler under the control of the filler valve. The diluted

![Fig. 2. APROS paper machine model [12].](image-url)
A typical setup of the papermaking process [2,3]. The process components were parameterized according to the basic simulation. The control loops were added and the built-in paper machine model was provided by VTT for valves respectively, all of which are in cascade control by the basis weight valve, the filler valve, and a set of steam basis weight, ash rate and the moisture rate are controlled stock fed as material input. The paper quality variables, sure that the short circulation process receives a constant consistency controller before the machine chest makes in Fig. 2 within the APROS paper machine environment. The control loop structure is shown closed loop process. The control loop structure is shown in Fig. 2 within the APROS paper machine environment. The consistency controller before the machine chest makes sure that the short circulation process receives a constant stock fed as material input. The paper quality variables, basis weight, ash rate and the moisture rate are controlled by the basis weight valve, the filler valve, and a set of steam valves respectively, all of which are in cascade control structures. Finally, the jet speed ratio of the pulp from a headbox to the web speed is controlled by the centrifugal feed pump.

The test of the novel dynamic causal digraph method was carried out in the APROS simulation environment. The APROS paper machine simulator provides validated model algorithms for the process components, such as equipment and automation, of the paper machine. These component models are mainly based on first principles. The simulation model is constructed by drawing the process flowsheet and setting up the parameters of the components [6]. One built-in paper machine model was provided by VTT for the basic simulation. The control loops were added and the process components were parameterized according to a typical setup of the papermaking process [2,3].

Five control loops were added to the basic model in order to enable testing of the proposed method on the closed loop process. The control loop structure is shown in Fig. 2 within the APROS paper machine environment. The consistency controller before the machine chest makes sure that the short circulation process receives a constant stock fed as material input. The paper quality variables, basis weight, ash rate and the moisture rate are controlled by the basis weight valve, the filler valve, and a set of steam valves respectively, all of which are in cascade control structures. Finally, the jet speed ratio of the pulp from a headbox to the web speed is controlled by the centrifugal feed pump.

The setups of the pulp recipe, process component parameters, and the grade of the produced paper are given in Table 5.

### 5. Case studies and results

This paper provides three case studies concerning fault detection and isolation on the paper machine short circulation process. The three studied faults are: (1) basis weight valve (actuator fault), (2) fiber consistency in the deculator (sensor fault), and (3) filler retention drop on the wire section (process fault). The three fault scenarios, the experimental procedures and the results for each fault scenario are presented in the remainder of this section.

#### 5.1. Fault scenarios

Three different types of fault were introduced into the APROS paper machine model in sequence in order to test the proposed method. The first fault was an actuator basis weight valve fault. In the APROS model, the fault was introduced by increasing the parameter ‘nominal pressure drop’ of the basis weight valve from 30 to 36 kPa. In reality, the corresponding fault is a blockage of the basis weight valve due to a fiber floculation phenomenon, which makes the opening of the valve’s flow area smaller than normal.

The second fault introduced was a measurement fault on the fiber consistency in the deculator. A drift fault with a slope of $3.5 \text{e}^{-66/\text{s}}$ was added to the fault-free measurement defic.

The third fault introduced was a process fault, in which the filler retention on the wire dropped. In the APROS model the fault was simulated by changing the retention setup for filler from 45% to 40%. Because of the smaller size of the filler compared to the fibers in the stock, the retention rate for the filler is relatively low. The decrease in

### Table 4

<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
<th>Type</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>baval</td>
<td>Basis weight valve opening</td>
<td>Actuator</td>
<td>–</td>
</tr>
<tr>
<td>wpfc</td>
<td>Filler consistency in the wire pit</td>
<td>Measurement</td>
<td>%</td>
</tr>
<tr>
<td>wpfic</td>
<td>Fiber consistency in the wire pit</td>
<td>Measurement</td>
<td>%</td>
</tr>
<tr>
<td>fival</td>
<td>Filler adding valve opening</td>
<td>Actuator</td>
<td>–</td>
</tr>
<tr>
<td>defc</td>
<td>Filler consistency in the deculator</td>
<td>Measurement</td>
<td>%</td>
</tr>
<tr>
<td>defic</td>
<td>Fiber consistency in the deculator</td>
<td>Measurement</td>
<td>%</td>
</tr>
<tr>
<td>hbfc</td>
<td>Filler consistency in the headbox</td>
<td>Measurement</td>
<td>%</td>
</tr>
<tr>
<td>hhfc</td>
<td>Fiber consistency in the headbox</td>
<td>Measurement</td>
<td>%</td>
</tr>
<tr>
<td>feedpump</td>
<td>Headbox feed pump rotation</td>
<td>Actuator</td>
<td>%</td>
</tr>
<tr>
<td>totalflow</td>
<td>Mass flow into the headbox</td>
<td>Calculated value</td>
<td>kg/s</td>
</tr>
<tr>
<td>bw</td>
<td>Basis weight of paper</td>
<td>Measurement</td>
<td>g/m²</td>
</tr>
<tr>
<td>ash</td>
<td>Ash consistency of paper</td>
<td>Measurement</td>
<td>%</td>
</tr>
</tbody>
</table>

### Table 5

Setup of the paper machine simulator [2,3]

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Unit/ Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pulp recipe (Broke:CP:TMP)</td>
<td>30:21:49</td>
<td></td>
</tr>
<tr>
<td>Fiber retention on wire</td>
<td>58</td>
<td>%</td>
</tr>
<tr>
<td>Filler retention on wire</td>
<td>50</td>
<td>%</td>
</tr>
<tr>
<td>Number of nips in wet press</td>
<td>3</td>
<td>Type: shoe</td>
</tr>
<tr>
<td>Linear pressure on the nips</td>
<td>70, 90, 120</td>
<td>kN/m</td>
</tr>
<tr>
<td>Steam supply pressure for dryer</td>
<td>500</td>
<td>kPa</td>
</tr>
<tr>
<td>Web speed</td>
<td>16</td>
<td>m/s</td>
</tr>
<tr>
<td>Web width</td>
<td>8</td>
<td>m</td>
</tr>
<tr>
<td>Paper basis weight</td>
<td>50</td>
<td>g/m²</td>
</tr>
<tr>
<td>Paper ash rate</td>
<td>18</td>
<td>%</td>
</tr>
<tr>
<td>Paper moisture rate</td>
<td>7</td>
<td>%</td>
</tr>
<tr>
<td>Basis weight valve nominal pressure drop</td>
<td>30</td>
<td>kPa</td>
</tr>
<tr>
<td>Filler weight valve nominal pressure drop</td>
<td>100</td>
<td>kPa</td>
</tr>
</tbody>
</table>
the filler retention will directly affect the ash rate of the final paper. Although the quality control for the ash rate maintained the paper quality following the setpoint, the paper machine was running inefficiently. Furthermore, the filler transportation ability was affected considerably. Moreover, since it is difficult to transfer filler to the final product, it will accumulate in the short circulation, which increases the wear of process devices like pumps, pipes and valves. As a result of the above, a filler retention drop fault can cause serious problems that require as early as possible detection and identification.

5.2. Description of the experiment

The case studies on the short circulation process were carried out in 3 steps. First the linearity of the paper machine process was tested in an open loop in order to evaluate whether linear state-space models can accurately describe the process behaviour. The structure of the causal digraph model was also decided in this phase, as shown in Fig. 3. The result of the test according to the digraph structure indicated that the paper machine model used by APROS is relatively linear when the paper grade defined in Table 5 was produced.

Secondly, the simulation in an open loop was run for modelling the cause-effect relationships in the digraph, during which the basis weight valve, filler valve and feed pump for the headbox were adjusted manually. The discrete-time state space models for the causal digraph model were identified by training the recurrent neural networks with linear nodes only. The NNDT software was used for the training task [9]. The obtained models are given in Table 6.

The last step of the experiment was to run the simulation with the control loops closed. In this phase, three different artificial faults were introduced into the APROS model and the proposed method was applied to detect and identify the faults. The total simulation time was 33,800 s (about 9.4 h), and the sampling time was 10 s. The three faults were introduced into the simulator in sequence at the time intervals 1800–5800 s, 9800–15,800 s and 17,800–25,800 s, respectively.

5.3. Diagnosis results for the actuator fault on basis weight valve

A fault occurring in the basis weight valve will affect the whole paper machine through the short circulation process. Since the quality controller keeps the basis weight variable following the setpoint, the fault may be difficult to notice. With the causal digraph, however, the global residual which tests the consistency between measurements and the globally simulated value shows that five variables in the digraph are suspected fault candidates. In order to form the correct set, the condition given in Eq. (7) was tested for all the nodes whose parent nodes have global

```
Table 6
State space models for the causal digraph of the short circulation process

<table>
<thead>
<tr>
<th>Nodes</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>totalflow</td>
<td>0</td>
<td>1.83989</td>
<td>1.00000</td>
</tr>
<tr>
<td>defc</td>
<td>[0.03349, 0.25251]</td>
<td>[-0.24461, 0.19556, 0.00342, 0.00434]</td>
<td>[-0.05293, -0.06521]</td>
</tr>
<tr>
<td></td>
<td>[0.00560, 0.05136]</td>
<td>[848.36680, -880.69628]</td>
<td>[-0.00008, -0.00009]</td>
</tr>
<tr>
<td>hbfc</td>
<td>0.76066</td>
<td>5.16024</td>
<td>0.04564</td>
</tr>
<tr>
<td>wpfc</td>
<td>0.91508</td>
<td>0.46859</td>
<td>0.07497</td>
</tr>
<tr>
<td></td>
<td>[1.80258, 0.17127]</td>
<td>[-0.04786, 0.26995, 0.00145, 1.46192]</td>
<td>[-0.01142, -0.00190]</td>
</tr>
<tr>
<td></td>
<td>[-4.94689, -0.04189]</td>
<td>[343.93241, 1974.26405]</td>
<td>[-0.00173, -0.00097]</td>
</tr>
<tr>
<td></td>
<td>[0.01142, 0.00521]</td>
<td>[4.18264, 0.06838]</td>
<td>[-0.01238]</td>
</tr>
<tr>
<td></td>
<td>[0.93235, 0.47458]</td>
<td>[353.31435, 353.31435]</td>
<td>[0.06663, 0.10453]</td>
</tr>
<tr>
<td></td>
<td>[-0.08148, 1.08965]</td>
<td>[27359.09031, 8649.37986]</td>
<td>[0.57723, 0.43597]</td>
</tr>
<tr>
<td></td>
<td>[0.89696, 0.01069]</td>
<td>[-3778.31082, 2297.99595]</td>
<td>[-0.30720, 0.25465]</td>
</tr>
<tr>
<td></td>
<td>[0.11879, 0.89669]</td>
<td>[-2688.71103, 1756.49659]</td>
<td>[-1.15704, 0.12563]</td>
</tr>
</tbody>
</table>
```

*a* In this table five decimals are reported although all the 15 decimals were used in the simulations.
residual detection in the digraph. The results of the test in Table 7 showed that there was no cancellation of different fault effects. For example, for the variable $bw$, the condition in Eq. (7) is not satisfied because $|CU(\{GR\}(bw))| = 1$ and $|CU(\{MLR_{j}(IndexU)\})| = |CU(\{ILR_{bw}(hbfc)\})| = 0$ shown in Figs. 4 and 6, respectively. Thus, the separation approach for different fault effects was not needed, and five variables $bw$, $ash$, $defic$, $hbfic$, $wpfic$, as shown in Fig. 4 at time period 1800–5800 s, formed the detection set. Note

| Nodes | IndexU | $|CU(\{GR\}(Y))|$ | $|CU(\{MLR_{j}(IndexU)\})|$ | Condition in Eq. (7) |
|-------|--------|-----------------|------------------|---------------------|
| defic | wpfic  | 1               | 1                | Not satisfied       |
| hbfic | defic  | 1               | 0                | Not satisfied       |
| wpfic | hbfic  | 1               | 0                | Not satisfied       |
| ash   | hbfic  | 1               | 0                | Not satisfied       |
| bw    | hbfic  | 1               | 0                | Not satisfied       |

Fig. 4. Global residuals for the variables $bw$, $ash$, $defic$, $hbfic$, $wpfic$, $defc$, $hbfic$ and $wpfc$. 
that the actuator variables baval, fival and feedpump are not shown in Fig. 4, due to the absence of faults during the simulation period.

LRs are generated for the variables in the detection set and fault isolation rules are applied. In the example shown in Fig. 5, the TLR(bw) indicates that during time 1800–5800 s the alarm on the variable bw is not local but originates from the parent nodes, while the TLR(defic) indicates that it is the fault origin. Moreover, the generated ILRs shown in Fig. 6 imply that the alarm on bw is from the parent node hbfic during time 1800–5800 s. As the final fault diagnosis result, the fault propagation path is formed as shown in Fig. 7, where all the variables in the detection set have been processed.

The fault was identified as a process fault by the nature rules: it propagated globally through the digraph and this studied fault is the blockage of the basis weight valve which is considered as a component in the process. Location of the process fault on the arc is performed in order to find the responsible process component. However, spurious results were produced due to the MISO structure of the digraph. In Fig. 7, the input arcs for fault origin node defic are \( \langle \text{totalflow, defic} \rangle, \langle \text{baval, defic} \rangle, \langle \text{fival, defic} \rangle \) and \( \langle \text{wpfic, defic} \rangle \). Thus, the number of sets of suspected arcs is \( (2^4 - 1) = 15 \), which are \{\( \langle \text{totalflow, defic} \rangle, \langle \text{totalflow, defic} \rangle, \langle \text{baval, defic} \rangle \}\}.

In order to perform the inference between arcs, knowledge about the relationship between the output arcs was introduced into the digraph model. The knowledge matrices for nodes were formed first based on the process knowledge of paper machines.

For the node totalflow the knowledge matrix

\[
M_{\text{totalflow}} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 1 \\
0 & 1 & 1
\end{bmatrix}
\]

was formed. It represents the relationship between the arc \( \langle \text{totalflow, bw} \rangle, \langle \text{totalflow, defic} \rangle \) and \( \langle \text{totalflow, defc} \rangle \). It can be seen from matrix \( M_{\text{totalflow}} \) that the inconsistency of arc \( \langle \text{totalflow, bw} \rangle \) does not result in an inconsistency of other output arcs, since the arc \( \langle \text{totalflow, bw} \rangle \) represents the cause-effect relationship between the mass flow through the headbox and the basis weight variable of the paper, and the arcs \( \langle \text{totalflow, defic} \rangle \) and \( \langle \text{totalflow, defc} \rangle \) are located before the headbox in the process. It can also be seen from matrix \( M_{\text{totalflow}} \) that whenever one of the arcs \( \langle \text{totalflow, defic} \rangle \) and \( \langle \text{totalflow, defc} \rangle \) is faulty, then the other one is also faulty. This is based on the fact that, in the process, the relationship between mass flow through the headbox and the fiber consistency in the deculator is correlated with the relationship between the mass flow and the filler consistency in the deculator, since they physically share the same pipe from the deculator.

![Fig. 5. TLR (bw) and TLR (defic).](image-url)
For the node baval the knowledge matrix
\[ M_{\text{baval}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \]
was formed. It represents the relationship between
the arc \langle \text{baval, defic} \rangle and \langle \text{baval, defc} \rangle. Theoretically, if the
actuator fault in the basis weight valve causes the inconsistency
of \langle \text{baval, defic} \rangle or \langle \text{baval, defc} \rangle, then another arc
should be faulty at the same time, since the relationship
between the control signal and the mass flow through the
basis weight valve changed due to the fault, thus subse-
sequently changing the relationship on the arc \langle \text{baval, defic} \rangle and \langle \text{baval, defc} \rangle. However, the dilution effect represented
by the arc \langle \text{baval, defc} \rangle is so small that the fault on it can-
not always produce residuals on the node defc that are
large enough for detection when the fault on arc
\langle \text{baval, defc} \rangle produces large enough residuals for detection.
On the other hand, if the fault on arc \langle \text{baval, defc} \rangle is signif-
icant enough to detect, then the fault on \langle \text{baval, defc} \rangle must be
significant enough for detection. Based on the above, the
noncommutative matrix
\[ M_{\text{baval}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \]
is used to repre-
sent the knowledge between arc \langle \text{baval, defc} \rangle and
\langle \text{baval, defc} \rangle. For the same reason, the matrix
\[ M_{\text{fival}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]
is obtained for the arcs
\langle \text{fival, defc} \rangle and \langle \text{fival, defc} \rangle.

For the node hbfc the knowledge matrix
\[ M_{\text{hbfc}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \]
was formed. It represents the relationship
between arc \langle \text{hbfc, bw} \rangle, \langle \text{hbfc, ash} \rangle and \langle \text{hbfc, wpfc} \rangle. This
sub matrix is obtained from the knowledge of the filler
material balance. Whenever the filler flows to the wire pit
decrease or increase, the filler flows to final production
(ash and bw) increase or decrease. The same matrix
\[ M_{\text{hbfc}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \]
is obtained for the arcs \langle \text{hbfc, bw} \rangle,
\langle \text{hbfc, ash} \rangle and \langle \text{hbfc, wpfc} \rangle.
For the nodes feedpump, defic, defc, wpfic and wpfc, which have only one output, the scalar 1 was assigned as their knowledge matrices.

The knowledge matrix $M$ for the whole digraph is formed by merging these knowledge matrices for nodes as shown below:

$$
M_{\text{feedpump}} = 1; \quad M_{\text{totalflow}} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 1 \\
0 & 1 & 1
\end{bmatrix}; \quad M_{\text{baval}} = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
0 & 1
\end{bmatrix};
$$

$$
M_{\text{fival}} = \begin{bmatrix}
1 & 1 \\
0 & 1
\end{bmatrix}; \quad M_{\text{defic}} = 1; \quad M_{\text{defc}} = 1
$$

$$
M_{\text{hbfic}} = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}; \quad M_{\text{hpfc}} = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix}; \quad M_{\text{wpfic}} = 1;
$$

$$
M_{\text{wpfc}} = 1
$$

$M = \text{blockdiag}(M_{\text{feedpump}}, M_{\text{totalflow}}, M_{\text{baval}}, M_{\text{fival}}, M_{\text{defic}}, M_{\text{defc}}, M_{\text{hbfic}}, M_{\text{hpfc}}, M_{\text{wpfic}}, M_{\text{wpfc}})
$$

The corresponding arcs represented by each row of the matrix $M$ are: \{feedpump, totalflow\}, \{totalflow, bw\}, \{totalflow, defic\}, \{totalflow, defc\}, \{baval, defic\}, \{baval, defc\}, \{fival, defic\}, \{fival, defc\}, \{defic, hbfic\}, \{defic, hpfc\}, \{hpfc, wpfic\}, \{hpfc, defc\}, \{hpfc, wpfc\}, \{wpfc, defc\}, \{wpfc, defc\}.

With the incident matrix $M$ representing the knowledge, the 15 sets of suspected arcs were tested with Eq. (13). The set of suspected arcs \{totalflow, defic\}, \{baval, defic\}, \{fival, defic\} is considered as an example. The $sv$ for this set is:

$$
sv = [0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
$$

and the test with Eq. (13) is performed as follows:

$$
\text{NUM}(sv) = 3 \neq \text{NUM}(sv \cdot M) = \text{NUM}([0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0]) = 5
$$

Thus the set \{totalflow, defic\}, \{baval, defic\}, \{fival, defic\} cannot be a possible result. The set \{baval, defic\}, \{wpfc, defc\} is considered as another example. The $sv$ for this set is

$$
sv = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
$$

and the test with Eq. (13) is performed as follows:

$$
\text{NUM}(sv) = 2 = \text{NUM}(sv \cdot M) = \text{NUM}([0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]) = 2
$$

Thus, this set is accepted as a possible result. After the test of all the sets of suspected arcs, only three of them were left as possible results: \{baval, defic\} and \{baval, defc\}, \{wpfc, defc\} and \{wpfc, defc\}. The number of possible results is decreased from 15 to 3 by using the inference mechanism between arcs.

The fault diagnosis results, i.e. the three sets of suspected arcs, provide valuable information for identifying the faulty process component in the case of an actuator fault, which is considered to be a special case of process fault. The arc in the first set is \{baval, defic\}, which corresponds to the process components: a basis weight valve and a hydrocyclone. Similarly, for the second set of suspected arcs, the hydrocyclone is suspected since it is located on both of the arcs; for the third set of suspected arcs, only the hydrocyclone is suspected to be faulty. So the possible faulty process components are the basis weight valve and hydrocyclone. Moreover, the direction of the $TLR-defic$, as shown in Fig. 5, indicates that the direction of the fault is negative, which implies that the fiber transport ability to the deculator has decreased. The reason for this could be either that the flow area of the basis weight valve has decreased or that the acceptance rate of fiber in the hydrocyclone has decreased.

The results presented above clearly show the advantages of the new method compared to the conventional one. In the term of fault detection, both the new method and the conventional method detected five variables, defic, hbfic, wpfic, bw, ash. However, the conventional method only indicated the fault origin variable, defic, whereas the proposed method improved the fault isolation result by indicating the sets of suspected arcs, \{baval, defic\}, \{wpfc, defc\} and \{wpfc, defc\}. Furthermore, the possible faulty process components, the basis weight valve and the hydrocyclone were identified.

### 5.4. Diagnosis results for the sensor fault on fiber consistency in deculator

The second faulty scenario is a measurement fault for the fiber consistency in the deculator. It was simulated during the time period 9800–15,800 s. The $GR$s shown in Fig. 4 indicate that, during this time period, only the variable defic produced an alarm. In the graph only variable hbfic has a parent node that has global residual detection. The test of fault effects separating condition in Eq. (7) was performed. Since $|CU(GR(hbfic))| = 0$ and $|CU(ILR_{ILR}(Index-U))| = |CU(ILR_{hbfic}(defic))| = 0$, there was no need to do the fault effects separation. The fault isolation rules clearly infer that the fault is local, and the fault nature rules infer that it is a measurement fault. Finally, the fault propagation path for the time period (9800–15,800 s) is shown as the fault diagnosis result in Fig. 8.
Table 8: Evaluation of conditions for the fault effect separation in fault scenario 3

| Nodes | IndexU | \(|CU(\text{GR})| \) | \(|CU(\text{MLR}(\text{IndexU}))| \) | Condition in Eq. (7) |
|-------|--------|-----------------|-----------------|---------------------|
| defc  | wpfc   | 1               | 0               | Not satisfied       |
| hbfc  | defc   | 1               | 0               | Not satisfied       |
| wpfc  | hbfc   | 1               | 1               | Not satisfied       |
| ash   | hbfc   | 1               | 1               | Not satisfied       |
| bw    | hbfc   | 0               | 1               | Satisfied           |

In this sensor fault case, the fault detection and isolation results were the same for both the proposed method and the conventional method. Due to the nature of the fault, neither the fault separation nor the inference between arcs was performed.

5.5. Diagnosis results for the process fault of the filler retention drop on the wire section

The third fault scenario is a retention drop of filler on the wire section, which can be characterized as a process fault. The fault was simulated during the time period 17,800–25,800 s. The variables ash, defc, wpfc, hbfc were detected from the \( GR \)s shown in Fig. 4 for the period 17,800–25,800 s, and the detection set was formed.

However, the \( TLR(bw) \) in Fig. 5 was detected as a negative fault by the CUSUM method, which implies that the \( GR \) and \( ILR_{bw}(\text{hbfc}) \) shown in Figs. 4 and 6, respectively, become undetectable even though a local fault exists. The reason for this is that the calculations of \( GR \) and \( ILR_{bw}(\text{hbfc}) \) use the global value of hbfc, which has a global alarm, as shown in Fig. 4.

The approach for separating fault effects was applied to the variables whose parent nodes have global detection. The results were shown in Table 8.

The condition in Eq. (7) is satisfied for the variable bw as

\[
(\text{IndexU} \neq \phi) \land (|CU(MLR_{bw}(|\text{IndexU}|))| = 1) \land (|CU(\text{GR}(bw))| = 0) = \text{True} \quad (22)
\]

where \( MLR_{bw}(\text{IndexU}) = ILR_{bw}(\text{hbfc}) \) was shown in Fig. 6.

Since only one element exists in the set \( \text{IndexU} \) for variable bw, it was tested with the Eq. (8). The sign \( CU(MLR_{bw}(\text{hbfc})) \) was negative as shown in Fig. 6. The residual \( MLR_{bw}(\{\text{hbfc}\} - \{\text{hbfc}\}) = MLR_{bw}(\phi) = GR(bw) \) was around zero as shown in Fig. 4, so the sign \( CU(MLR_{bw}(\text{hbfc})) - MLR_{bw}(\{\text{hbfc}\} - \{\text{hbfc}\}) \) is negative. Hence the fault effect from hbfc was found to be opposite to the local fault of bw, since the set defined in Eq. (8) is given as

\[
CU(MLR_{bw}(\text{hbfc})) \cdot (CU(MLR_{bw}(\{\text{hbfc}\})) - MLR_{bw}(\{\text{hbfc}\} - \{\text{hbfc}\})) > 0 \Rightarrow \text{IndexU} = \{\text{hbfc}\} \quad (23)
\]

The residuals \( GRM \) and \( ILM_{bw}(\text{hbfc}) \) were generated for the variable bw according to Eq. (9) and evaluated by the CUSUM method, as shown in Fig. 9. The detection result of \( GRM \) during the period 17,800–25,800 s adds the variable bw into the detection set. The fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

The residuals \( GRM \) and \( ILM_{bw}(\text{hbfc}) \) were generated for the variable bw according to Eq. (9) and evaluated by the CUSUM method, as shown in Fig. 9. The detection result of \( GRM \) during the period 17,800–25,800 s adds the variable bw into the detection set. The fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.

Location of the process fault is performed in order to find the responsible process component. However, spurious results were produced due to the multiple fault origins and multiple input arcs for each fault origin. In Fig. 11, the fault isolation rules were applied to the variables in the detection set, and the fault propagation path is shown in Fig. 11. Three fault origins, ash, bw, and wpfc, were located due to the detection of the \( TLR(\text{ash}), TLR(\text{wpfc}) \) and \( TLR(bw) \) shown in Figs. 10 and 5, respectively. Moreover, the fault nature rules indicate that it is a process fault, since the fault propagates through the global model.
Thus, the set \{h_{hbfc,wpfc}, h_{hbfc,ash}, h_{hbfc,bw}\} cannot be a possible result. The set \{h_{hbfc,wpfc}, h_{hbfc,ash}, h_{hbfc,bw}\} is considered as another example. The sv for the set is

\[sv = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0]\]

and the test with Eq. (13) is performed as follows:

\[NUM(sv) = 3 = NUM(sv \cdot M) = NUM([0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 3 \ 3 \ 3 \ 0 \ 0]) = 3\]  

Thus, this set is accepted as a possible result. After the test of all the sets of suspected arcs, only two were left as possible results: \{h_{hbfc,wpfc}, h_{hbfc,ash}, h_{hbfc,bw}\} and \{h_{hbfc,wpfc}, h_{hbfc,ash}, h_{hbfc,bw}, \text{totalflow,bw}\}. The number of possible results decreased from 21 to 2 by using the inference mechanism between arcs.

The fault diagnosis results, i.e. the two sets of suspected arcs, provide valuable information in identifying the faulty process component in the case of a process fault. The first arc in the first set \{h_{hbfc,wpfc}, h_{hbfc,ash}, h_{hbfc,bw}\} is \(h_{hbfc,wpfc}\), which corresponds to the process components: a wire section and the white water tray. Similarly, the arcs \(h_{hbfc,ash}\) and \(h_{hbfc,bw}\) correspond to the process components: a wire section, a wet press and a drying group. Thus the suspected process component is the wire section, since it is located on all three arcs. The same result is obtained from the second set. Moreover, the results also indicated that the fault occurred for the filler material but not for the fiber material. Finally, the direction of the TLR(wpfc), as shown in Fig. 10, implies that the direction of the fault is the drop in the filler retention, rather than an increase in the filler retention.

In this fault scenario, the fault effects separation approach was applied to locate correct fault origins; with the inference mechanism between arcs, the number of the possible fault results decreased significantly and the process fault was located on the arcs, which subsequently helps to identify the faulty process component, i.e. a wire section. The example of this fault scenario illustrates the necessity and the power of the modified dynamic causal digraph for industrial applications.
The improvement of the proposed method over the conventional one can be seen clearly from the comparison of the results. Firstly in the term of fault detection, the conventional method only detected four variables, defc, hbfc, wpfc, ash, while the proposed method was able to detect one more variable, bw, with the modified residuals. Secondly, the result of the fault isolation was improved by the inference between the arcs. The results from the conventional method was the identification of the fault origin variables, wpfc and ash, while also the suspected arcs were identified by the proposed method as \{h_{\text{hbfc}}, \text{defc}\}, \{h_{\text{hbfc}}, \text{wpfc}\}, \{h_{\text{hbfc}}, \text{ash}\}, \{h_{\text{hbfc}}, \text{bw}\}\} and \{h_{\text{hbfc}}, \text{wpfc}\}, \{h_{\text{hbfc}}, \text{ash}\}, \{h_{\text{hbfc}}, \text{bw}\}\}.

The improvement of the proposed method over the conventional one can be seen clearly from the comparison of the results. Firstly in the term of fault detection, the conventional method only detected four variables, defc, hbfc, wpfc, ash, while the proposed method was able to detect one more variable, bw, with the modified residuals. Secondly, the result of the fault isolation was improved by the inference between the arcs. The results from the conventional method was the identification of the fault origin variables, wpfc and ash, while also the suspected arcs were identified by the proposed method as \{h_{\text{hbfc}}, \text{wpfc}\}, \{h_{\text{hbfc}}, \text{ash}\}, \{h_{\text{hbfc}}, \text{bw}\}\} and \{h_{\text{hbfc}}, \text{wpfc}\}, \{h_{\text{hbfc}}, \text{ash}\}, \{h_{\text{hbfc}}, \text{bw}\}\}.

Table 9

<table>
<thead>
<tr>
<th>Faults</th>
<th>Fault type</th>
<th>Dynamic causal digraph</th>
<th>Novel dynamic causal digraph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault 1</td>
<td>Actuator fault (treated as process fault)</td>
<td>Nodes: defc, hbfc, wpfc, bw, ash</td>
<td>Fault origin: defc, {h_{\text{hbfc}}, \text{wpfc}}, {h_{\text{hbfc}}, \text{ash}}, {h_{\text{hbfc}}, \text{bw}}} and {h_{\text{hbfc}}, \text{wpfc}}, {h_{\text{hbfc}}, \text{ash}}, {h_{\text{hbfc}}, \text{bw}}}</td>
</tr>
<tr>
<td>Fault 2</td>
<td>Sensor fault</td>
<td>Nodes: defc</td>
<td>Fault origin: defc</td>
</tr>
<tr>
<td>Fault 3</td>
<td>Process fault</td>
<td>Nodes: defc, hbfc, wpfc, ash</td>
<td>Fault origin: defc</td>
</tr>
</tbody>
</table>

Fig. 11. Modified dynamic causal digraph (17,800–25,800 s).
In addition, the corresponding possible faulty process component was identified as a wire section.

6. Conclusions

A novel dynamic causal digraph reasoning method for fault diagnosis in a paper machine short circulation process was presented in this paper. The detectability and the capacity to handle process faults were improved with the method. The detectability was enhanced by separating different fault effects in the residual generation, and the presentation of the inference mechanism between arcs allows the method to locate the process fault on the arcs.

Three different fault scenarios were tested with the proposed method in a paper machine short circulation process: an actuator fault (treated as a process fault), a sensor fault and a process fault. The results for the second fault scenario, i.e. a sensor fault, showed that the conventional dynamic method was able to handle this case well. In the third scenario, however, it could not detect the variables correctly or form the correct detection set. The correct nodes were detected for the third fault with the help of the fault effects separation approach. Moreover, the fault diagnosis result for the first and third scenario was improved further by locating the process fault on the arcs using the proposed inference mechanism. The final result also showed that much more information was provided by the novel method, which helped in identifying the faulty process component. The improvements offered by the new method are highlighted as a summary of comparison of results in Table 9.

Acknowledgements

The authors gratefully acknowledge the Finnish Funding Agency for Technology and innovation for funding this work, which was carried out performed within the project Production 2010. VTT is gratefully acknowledged for making available the APROS simulator.

References