Tikkala, Vesa-Matti; Zakharov, Alexey; Jämsä-Jounela, Sirkka-Liisa

Detection and Isolation of Oscillations Using the Dynamic Causal Digraph Method

Published in:
Conference on Control and Fault-Tolerant Systems (SysTol'10), Nice, France, October 6-9, 2010

DOI:
10.1109/SYSTOL.2010.5675972

Published: 01/01/2010

Document Version
Peer reviewed version

Please cite the original version:
Detection and Isolation of Oscillations Using the Dynamic Causal Digraph Method

Vesa-Matti Tikkala, Alexey Zakharov, Sirkka-Liisa Jämsä-Jounela

Abstract—This paper proposes a modification to the dynamic causal digraph (DCDG) method in order to address the detection and isolation of oscillations in a process. The proposed detection method takes advantage of the properties of residual signals generated by the DCDG method by studying their zero-crossings. The method is tested in an application to a board making process and the results are presented and discussed.

I. INTRODUCTION

Demands to keep industrial processes running efficiently with a high rate of utilization are increasing constantly due to the tightening global competition. Since, the modern industrial processes are complex and large-scale, operator-based monitoring cannot guarantee early enough detection and reliable diagnosis of the faults and abnormalities. Therefore, the detection and diagnosis of different abnormal and faulty conditions in the processes have become increasingly important.

Common problems causing inefficient operation and production losses in the process industry are oscillations. Oscillatory disturbances readily propagate in the process and cause extensive variation in the process variables. The oscillations are usually originated under feedback control, and they may have various causes which have been categorized by Thornhill and Horch into non-linear and linear causes [1]. Non-linear causes include for example extensive static friction in the control valves, on-off or split range control, sensor faults, process non-linearities and hydrodynamic instabilities. The most common linear causes are poor controller tuning, controller interaction and structural problems involving process recycles [1]. According to Choudhury et al., valve stiction is, however, the most common cause of these oscillations in control loops [2].

The detection and diagnosis of oscillations have been previously addressed by data-based methods which study, for example, the properties of controller error signals [3], [4], the spectral properties [5] or the nonlinearity of the measurement signals [6], [7]. Also, a variety of multivariate methods, such as principal component analysis [8] and non-negative matrix factorization [9], have been applied to solve this diagnosis task. A recent trend has been to introduce process information into the diagnosis of plant-wide oscillations. Applications in which the process connectivity information has been integrated into data-based analyses to check hypotheses on the fault origin, have been presented [10], [11].

This paper aims at further development of the dynamic causal digraph (DCDG) method by addressing the detection and isolation of plant-wide oscillations. A detection algorithm, which is able to deal with oscillatory residuals, is proposed and integrated into the DCDG method. In this paper, the modified DCDG method is used to detect and isolate low-frequency oscillations caused by a valve stiction fault in a board machine process.

The paper is organized as follows. In Section II, the dynamic causal digraph method and the new detection algorithm are introduced. The process and the test environment are described in Section III. The results of the testing are presented in Section IV followed by the conclusions in Section V.

II. ENHANCED DCDG METHOD FOR THE DETECTION AND ISOLATION OF OSCILLATIONS

The dynamic causal digraph method employs the process knowledge formalized as a causal digraph model in order to perform the ordinary fault diagnosis tasks as presented in [12], [13]. In the enhanced DCDG, the detection of faults is performed using the proposed method which observes the zero-crossings in the residuals generated by a comparison of cause-effect models and the process measurements. Next, the isolation is carried out by applying a set of inference rules to the residuals in order to extract the fault propagation path. Finally the arcs in the digraph that explain the faulty behavior are identified. The enhanced DCDG method is described in more detail in the following.

A. Fault detection

Fault detection is performed in two steps: residual generation and fault detection in the residuals using the modified cumulative sum (CUSUM) algorithm.

1) Residual generation with dynamic models: The dynamic causal digraph produces two kinds of residual to be used in fault detection and isolation: global (GR) and local residuals (LR). The global residual is produced from the difference between the measurement and the global propagation value:

$$GR(Y) = Y(k) - \hat{Y}(k),$$

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

$$\hat{Y}(k) = f_Y(\hat{U}(k-1), \hat{U}(k-2), \ldots),$$

where $f_Y$ is a discrete-time model describing the cause-effect relationship from $n$ predecessor nodes $U_i$ to node $Y$. 

978-1-4244-8154-5/10/$26.00 ©2010 IEEE
\( \hat{U}(k - \tau) = \{\hat{u}_1(k - \tau), \ldots, \hat{u}_n(k - \tau)\} \) are the lagged global propagation values from the predecessors with time lags \( \tau = 1, 2, \ldots \) depending on the system order.

The local residuals are subcategorized into three types: individual local residuals (ILR), multiple local residuals (MLR) and total local residuals (TLR) [14].

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:

\[
\begin{align*}
    I LR_{k}^{\mu} &= Y - \bar{Y}, \\
    \bar{Y}(k) &= f_Y(\bar{U}(m, k - 1), \bar{U}(m, k - 2), \ldots),
\end{align*}
\]

where

\[
\bar{U}(m, k - \tau) = \left\{ \bar{u}_i(k - \tau) \right\},
\]

\[
\bar{u}_i(k - \tau) = \left\{ \begin{array}{ll}
    u_i(k - \tau), & i \neq m \\
    u_i(k - \tau), & i = m
\end{array} \right., \quad 1 \leq i \leq n.
\]

\( \hat{u}_i(k) \) is the lagged global propagated value from the predecessors, and \( u_i(k - \tau) \) is the measurement for the \( i \)-th parent node.

Similarly, the \( MLR_{k}^{\mu} \) is produced as

\[
\begin{align*}
    M LR_{k}^{\mu} &= Y - \bar{Y}, \\
    \bar{Y}(k) &= f_Y(P_{k}^{\mu}, k - 1), P_{k}^{\mu}, k - 2, \ldots),
\end{align*}
\]

where

\[
\bar{U}(P_{k}^{\mu}, k - \tau) = \left\{ \bar{u}_i(k - \tau) \right\},
\]

\[
\bar{u}_i(k - \tau) = \left\{ \begin{array}{ll}
    u_i(k - \tau), & i \notin P_{k}^{\mu} \\
    u_i(k - \tau), & i \in P_{k}^{\mu}
\end{array} \right., \quad 1 \leq i \leq n.
\]

\( P_{k}^{\mu} \) is the set of indices of the predecessors which use the measurement as an input. The \( T LR(Y) \) is produced with \( P_{k}^{\mu} = P_{k} \), where \( P_{k} \) is the set of indices of all the predecessors of \( Y \).

The residual generation scheme follows the DCDG method developed in [14].

2) Fault detection using the modified CUSUM method: The proposed detection method utilizes the cumulative sum (CUSUM) method presented in [15], by applying it to the detection of a change in the mean and variance of the zero-crossings in the residual signals. The CUSUM algorithm is defined for a positive change as follows:

\[
\begin{align*}
    U_0 &= 0, \\
    U_n &= \sum_{k=1}^{n} d(k) - \mu_0 - \frac{\beta}{2}, \\
    m_n &= \min_{0 \leq k \leq n} U_k,
\end{align*}
\]

where \( \beta \) is a user-specified minimum detectable change, \( d(k) \) the observed signal with nominal mean value equal to \( \mu_0 \). Whenever \( U_n - m_n > \lambda \), a change is detected, where \( \lambda \) is a design parameter, usually tuned according to the requirements for the false alarm and missed alarm rates.

The signal observed by the CUSUM algorithm, called the detection signal, is defined as follows

\[
d(k) = \max \left\{ \Delta t(k), \frac{\sigma^2_{\Delta t}(k)}{\Delta t(k)} \right\},
\]

where \( \max \{\cdot\} \)-operator takes the maximum of its arguments, \( \Delta t(k) \) and \( \sigma^2_{\Delta t}(k) \) are the mean and the variance of the time between consecutive zero-crossings in a residual, respectively. Both \( \Delta t \) and \( \sigma^2_{\Delta t} \) are calculated in a moving window of length \( l: [e(k-l), e(k)] \), where \( e(k) \) is the residual.

In normal operation, when the residuals are assumed to be zero-mean Gaussian noise, \( \Delta t(k) \) and \( \sigma^2_{\Delta t}(k) \approx 2 \), since the probability of \( e(t) \) having a different sign than \( e(t-1) \) is 0.5 for all \( t \). Therefore, the nominal mean value of the observed signal in (7) can be set as \( \mu_0 = 2 \) and \( \beta \) and \( \lambda \) are then tuned to obtain robust detection with minimal false alarms. The window length \( l \) must be selected to be larger than one half of the expected period in the residual.

B. Fault isolation

1) Isolation of the fault propagation path: Fault isolation is performed recursively for the detected nodes by using a set of rules. These isolation rules, developed by Montmain and Gentil in [14], are converted into a table for the convenience of implementation, as shown in Table I. After the isolation the nature of the fault is determined by using rules in Table II.

<table>
<thead>
<tr>
<th>CU(GR(X))</th>
<th>CU(TLR(X))</th>
<th>Fault nature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/1</td>
<td>1/1</td>
<td>Local fault for that child node</td>
</tr>
<tr>
<td>1/1</td>
<td>0/1</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>

* \( X \) is the subscript of any child node of the node \( Y \).

2) Isolation of the faulty process component: In the case of a process fault, in addition to locating the fault on the variables (nodes), locating it on the arcs is also desirable. However, the MISO structure of the digraph causes problems by generating multiple possible results as \( 2^n - 1 \), \( n \geq 1 \), where \( n \) is the number of input arcs of the fault origin node(s).

In order to decrease the number of possible results, an inference mechanism between the arcs proposed in [16] is used. The inference mechanism is based on an inter-arc knowledge matrix \( M \) defined for node \( U \) as follows

\[
M_U(i, j) = \left\{ \begin{array}{ll}
    1, & \text{if inconsistency in arc } (U, i) \\
    0, & \text{causes inconsistency to } (U, j) \\
    0, & \text{otherwise}
\end{array} \right.,
\]

where \( i \) and \( j \) refer to the matrix rows and columns, respectively.

Next, each set of suspected arcs is tested in order to determine whether the fault may be caused exactly by the current set of arcs. In order to do it the matrix \( M \) is
multiplied with a vector representing the suspected arc set, which is defined as follows

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

(10)

where \(ARC(M, i)\) gives the arc corresponding to the \(i\)th row in the matrix \(M\). \(S\) is the set of suspected arcs. If the number of non-zero elements of \(sv\) have changed, the current suspected set of arcs must be excluded.

### III. Description of the Process and the Valve Stiction Faults

This test focuses on the stock preparation of the board machine at Stora Enso’s mills in Imatra, Finland. The simulations are run on a board machine simulator model in the APROS simulation environment.

#### A. The Board Machine Process

The board making process begins with the preparation of raw materials in the stock preparation section, as shown in the flowsheet in Fig. 1. Different types of pulp are refined and blended according to a specific recipe in order to achieve the desired composition and properties for the board grade to be produced. The consistency of the stock is controlled with dilution water.

The blended stock passes from the stock preparation to the short circulation. First, the stock is diluted in the machine chest to the correct consistency for web formation. The diluted stock is then pumped with a fan pump, which is used to control the basis weight of the board, to cleaning and screening. Next, the stock passes to the head box, from where it is sprayed onto the wire in order to form a solid board web.

The excess water is first drained through the wire and later by pressing the web board between rollers in the press section. The remaining water is evaporated off in the drying section using steam-heated drying rolls.

The variables used in the causal digraph model of the stock preparation are listed in Table III and the structure of the digraph model of the stock preparation section is presented in Fig. 2.

#### B. Valve Stiction Faults

A control valve is the most common final control element used in the process industry [2]. Therefore, the diagnosis of faults in valves is of great importance. Stiction, short for static friction, is a problem in control valves since it can cause significant disturbances in the process variables. A valve suffering from excessive stiction sticks when the control signal, for example, changes the direction and does not move until the force required to move the valve shaft exceeds a certain limit. When the valve starts to move, it jumps and then follows the control signal before it sticks again. A sticking valve is likely to cause oscillations when it is involved in a control loop.

The stiction in valves has been modelled and studied e.g. in [17] and [18]. This paper considers a stiction fault in a pressure control valve which causes the control loop to oscillate and disturbs the operation of the plant.

### TABLE I

<table>
<thead>
<tr>
<th>Fault isolation rules of the dynamic causal digraph</th>
</tr>
</thead>
<tbody>
<tr>
<td>(CU(\text{Gr}(Y)))</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1/-1</td>
</tr>
<tr>
<td>1/-1</td>
</tr>
<tr>
<td>1/-1</td>
</tr>
</tbody>
</table>

* \(\forall i \neq m, i \in P_Y, m \in P_j, m \neq P_j, P_Y\) is the set of subscripts of parent nodes of the node \(Y\).

** \(\forall i, m, i \in P_Y, m \in P_Y, \forall P_i, P_j \subseteq P_Y\).
IV. TESTING AND RESULTS

A. Simulation Environment and Fault Simulation

The Imatra board machine model was developed by Stora Enso and VTT in the APROS environment. It was originally constructed on the basis of modeling and simulation studies carried out during 1998–2002 for Stora Enso's Imatra mills. It has been previously used for grade change simulations and in studies reported by Lappalainen et al. in [19].

A valve stiction fault was simulated in the stock preparation part of the board machine using the APROS board machine model. The faulty valve is located in the CTMP line and is used to control the feeding pressure $p_c$ of the blend chest. The two-parameter data-driven valve stiction model proposed by Choudhury et al. in [18] was implemented in the APROS simulation software for the simulation. The deadband and slip-jump parameters of the stiction model were set to $S = 0.06$ and $J = 0.06$ respectively. That is, the stiction is set to be 6% of the operation range of the valve. The fault was evoked by a step change to the setpoint of $p_c$.

The fault occurring at $t = 100$ causes an oscillation with a period of approximately 120 samples, which affects most of the variables in the stock preparation. Fig. 3 shows the measured variables during the fault simulation and it demonstrates clearly the effect of the fault in the process.

B. Fault Detection and Isolation Results

First, the global residuals for all variables were produced by comparing the measured values of the variables and the estimates generated using the dynamic causal digraph model. Then, the detection signals were produced by calculating the mean and the variance of zero-crossings in the global residuals. The proposed detection method was applied to analyse the residuals in order to detect the faulty nodes. The parameters of the modified CUSUM method were set to the following: $\beta = 10$, $\lambda = 4$ and $l = 200$. The global residuals, detection signals and the detection results for variables $f_c$ and $c_e$ are presented in Fig. 4. The fault is detected in both signals $GR(f_c)$ and $GR(c_e)$. The change in the detection signal $d_f(k)$ was detected for the first time at $k = 103$, three time instants after the fault occurred. However, the detection result is not reliable until $k = 170$. The global residual of $c_e$ is detected later at $k = 215$.

Local residuals were generated in order to carry out the inference to isolate the origin of the fault. The local residual, the detection signal and the detection results of $c_e$ are shown in Fig. 5. The detection signal changes slightly after the fault occurrence, but no detection is however made.

The performance of the proposed detection method is satisfactory. The faults are detected with a reasonable delay and no false alarms are generated. Detection in variable $c_{mc}$, based on the structure of the process and the forecast of the fault propagation, was also expected. However, based on
The work presented in this paper represents the first step in addressing the detection and isolation of faults causing oscillatory behaviour in a process using the DCDG method. In future, the aim is to generalize the diagnosis methodology by developing new detection methods that are able to cover a wider range of faults occurring in industrial processes.

REFERENCES


V. CONCLUSIONS

A method for detecting oscillatory residual signals was presented in this paper. The method was integrated into the DCDG fault diagnosis method and tested in an application to a board making process.

The proposed method enables the detection and isolation of oscillations caused by valve stiction faults in the process by exploiting the statistical properties of the residual signals. The results show that the proposed detection method is able to detect the fault successfully and to provide the information required for fault isolation.