



HAL
open science

Diagnosability improvement of dynamic clustering through automatic learning of discrete event models

Nathalie A Barbosa, Louise Travé-Massuyès, Victor H Grisales

► **To cite this version:**

Nathalie A Barbosa, Louise Travé-Massuyès, Victor H Grisales. Diagnosability improvement of dynamic clustering through automatic learning of discrete event models. IFAC-PapersOnLine, 2017, 50 (1), pp.1037-1042. 10.1016/j.ifacol.2017.08.214 . hal-02004430

HAL Id: hal-02004430

<https://hal.science/hal-02004430>

Submitted on 1 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Diagnosability improvement of dynamic clustering through automatic learning of discrete event models

Nathalie A. Barbosa ^{*,**} Louise Travé-Massuyès ^{**}
Victor H. Grisales ^{***}

^{*} *Universidad Nacional de Colombia, Department of Electrical and Electronics Engineering, Bogotá, Colombia*
(e-mail: nabarbosa@unal.edu.co)

^{**} *LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France, 7, Avenue du Colonel Roche, F-31031 Toulouse, France.*
(e-mail: louise@laas.fr)

^{***} *Universidad Nacional de Colombia, Department of Mechanical and Mechatronics Engineering, Bogotá, Colombia*
(e-mail: vhgrisalesp@unal.edu.co)

Abstract: This paper deals with the problem of improving data-based diagnosis of continuous systems taking advantage of the system control information represented as discrete event dynamics. The approach starts from dynamic clustering results and, combining the information about operational modes, automatically generates a discrete event system that improves clustering results interpretability for decision-making purposes and enhances fault detection capabilities by the inclusion of event related dynamics. The generated timed discrete event system is adaptive thanks to the dynamic nature of the clusterer from which it was learned, namely *DyClee*. The timed discrete event system brings valuable temporal information to distinguish behaviors that are non-diagnosable based solely on the clustering itself.

© 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Clustering, DES, Diagnosability, Machine learning, Situation assessment

1. INTRODUCTION

The problem of diagnosing continuous systems, discrete event systems and systems exhibiting both, continuous and discrete dynamics, has been proposed in many forms (Travé-Massuyès et al., 2001; Sampath et al., 1996; Bayouhd et al., 2008; Gaudel et al., 2015) from which the following definition of diagnosability is extracted:

Definition 1. (Diagnosability). A system is diagnosable with a given set of sensors if and only if (i) for any relevant combination of sensor readings (measurements) there is only one diagnosis candidate and (ii) all faults of the system belong to a candidate diagnosis for some sensor readings (Console et al., 2000; Travé-Massuyès et al., 2001).

In complex processes, the physical plant is mostly continuous, so the set of sensors are usually related to continuously changing variables. Nevertheless, the system control is often performed by a supervisory controller that imposes discrete switching behavior between several operating modes (Bayouhd et al., 2008). This approach, discrete in a high level of abstraction but continuous in the low level dynamics is known as *hybrid*. Definition 1 implies that it is not possible to differentiate the occurrence of

one behavior over another if the set of sensors used to observe them is statistically similar. In such cases, the two behaviors are non-diagnosable (or equivalently the fault underlying the faulty behavior is non-diagnosable) and more information has to be introduced in order to differentiate them. This information may come from the discrete behavior (Bayouhd and Travé-Massuyès, 2014).

Clustering techniques establish a representation of the process behaviors directly from the data acquired by sensors. The sensed quantities are also known as features. They work by grouping samples in the feature space and then assessing the process situation online by tracking the process measurements (which is called ‘situation assessment’) (Kempowsky et al., 2006). The above definition of diagnosability, based on sensor readings, is the most appropriate in the clustering-based (or data based) situation assessment context. Note however that definitions better suited to model based approaches have been proposed for continuous systems (Travé-Massuyès et al., 2006) and discrete-event systems (Lin and Lin, 1993; Sampath et al., 1995). An analysis of the correspondance of these definitions can also be found in (Travé-Massuyès et al., 2006).

If dynamic clustering (Aggarwal et al., 2003; Angelov, 2011; Kwak et al., 2015) is used, the ageing dynamics of the process can be integrated as cluster drifts (Barbosa Roa et al., 2015). The dynamism of the clusterer allows correct tracking of evolving systems without generating

* This work was partially supported by the Administrative Department of Science, Technology and Innovation COLCIENCIAS, Colombia.

an increased number of non representative clusters, the number of clusters then remaining finite.

In spite of this advanced capability, if diagnosis is performed along a pure online clustering approach, the order in which clusters are identified and recognized in time is not recorded and is lost. In this paper an improvement in the diagnostic capabilities of the dynamic clustering algorithm that we presented in (Barbosa Roa et al., 2015) is introduced. The algorithm, called *DyClee*, is improved with the knowledge of the order in which process discrete dynamics change and with event-based temporal information.

It is well accepted that, for several purposes among which diagnosis, large scale dynamic systems involving continuous processes can be viewed as discrete event systems (DES) at some level of abstraction (Sampath et al., 1996). In (Kempowsky et al., 2006) the system trajectory is described by a sequence of fuzzy classes to which the actual situation (normal or faulty) belongs. (Sampath et al., 1996) achieve system diagnosis from a set of finite-state models of the subsystems to be diagnosed and then building a finite-state machine known as the *diagnoser* based on observable events. The diagnoser can also serve as the basis for diagnosability analysis. Decentralised DES diagnosis approaches have been proposed to achieve on-line diagnosis while reducing diagnosis computational requirements (Pencolé and Cordier, 2005; Cordier et al., 2007).

In this paper we show that *DyClee* can be used to learn a timed DES from which it can distinguish behaviors that were non-diagnosable before. Indeed, the DES allows to incorporate event based knowledge (for example coming from the control system) that is more adequately described with discrete dynamics. In addition, the DES provides temporal information that is not present in the classification itself.

The developments made in this paper are based on the following hypotheses:

Hypothesis 1. Normal and abnormal behaviors are statistically different from each other.

Hypothesis 2. The system expert is able to provide the labels of the (finite) set of nominal operational modes as well as the set-point changes to transition between them.

Hypothesis 3. Changes in the system set-points happen in a much slower timescale than system dynamics.

The paper is organized as follows: Section 2 introduces the benchmark and the simulation scenario used to illustrate the developments. Section 3 introduces *DyClee* and presents the clustering results on the test case. Section 4 presents the method to build a DES from dynamic clustering results. Section 5 shows how diagnosability is improved by the introduction of the DES in the test case. Finally, section 6 concludes the paper and provide some ideas for future work.

2. TEST CASE: THE CSTH

The continuous stirred tank heater (CSTH) is a benchmark of a stirred tank in which hot (50°) and cold (24°) water are mixed and further heated using steam; the final mix is then drained using a long pipe (Thornhill et al.,

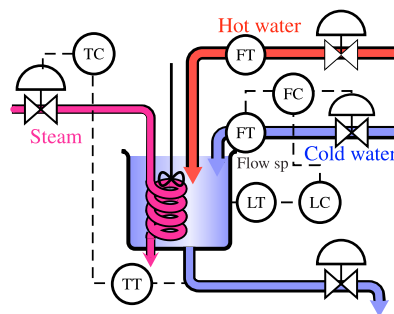


Fig. 1. The continuous stirred tank heater

Table 1. Test scenario: Multiple fault simulation over a month timespan. (Simulation time in $sec \times 10^5$)

t	Event	t	Event
0	Start at $OP1$	9.0	e -leak 3 starts
1.5	e -leak 1 starts	9.6	e -leak 3 fixed
2.4	e -leak 1 fixed	12.0	Changed to $OP2$
3.5	e -leak 1 restarts	15.0	S_{valve} stuck 10%
3.8	e -leak 2 starts	15.6	Valve repaired
4.5	Leaks 1 & 2 fixed	18.0	e -leak 4 starts
5.5	S_{valve} stuck 0%	18.4	e -leak 4 fixed
5.8	Valve repaired	20.0	HW_{valve} stuck 40%
6.5	HW_{valve} stuck 10%	20.6	Valve repaired
7.0	Valve repaired	24.2	End

2008), as shown in Figure 1. Process inputs are set-points for the cold water, hot water and steam valves. Process outputs are cold water flow, tank level and temperature. Process inputs and outputs represent electronic signals in the range 4 – 20mA. The benchmark is tested in closed-loop. PID controllers are used to guide the plant for the suggested operation points. The difference between the operational point (OP) $OP1$ and $OP2$ is reflected in whether or not the hot water flow is used.

In (Barbosa Roa et al., 2016) we implemented dynamic faults as evolving leaks or pipe clogging in this benchmark and presented two different scenarios to prove *DyClee* capacity to track state drift and multiple non-persistent faults. The second scenario evidenced how the no inclusion of event related information, e.g. operating mode, reduce *DyClee* capacity to distinguish the occurrence of faults when its observable behavior is similar to that of a normal state. In this paper we use the same scenario to show how the *DyClee* diagnosis capabilities can be improved.

Test scenario description: Several faults between evolving leaks (e -leaks) and stuck valves were simulated. The total simulation time of this scenario is equivalent to a timespan of a month (2.419.200 seconds) in which the plant works half of the time in $OP1$ and the other half in $OP2$. The faulty events included in this scenario are detailed in Table 1 and its time of occurrence is illustrated along with CSTH output signals in Figure 2 ¹

3. DYNAMIC CLUSTERING ALGORITHM

DyClee algorithm is conformed by two clustering stages one based on distance and one based on density. The algorithm macro-description is shown in Figure 3. System

¹ In figure 2 e -leaks are denoted as l_i and stuck valves are denoted as s_j .

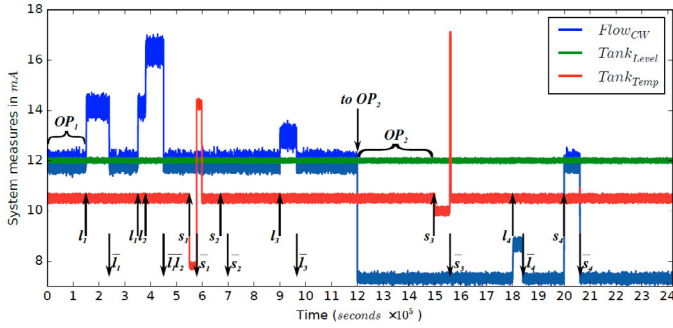


Fig. 2. CSTDH output measurements and faulty events.

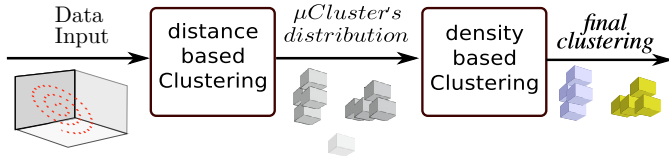


Fig. 3. Principle of *DyClee*

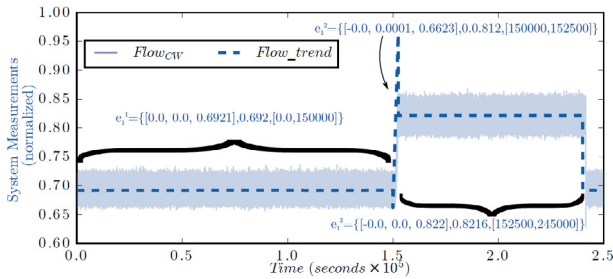


Fig. 4. Cold water flow episodes found between $t = [0, 2.5 \times 10^5]$

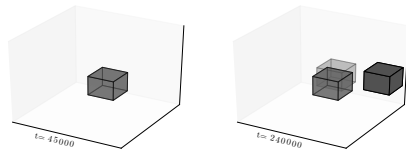


Fig. 5. Output of the distance based stage at $t = 0.45 \times 10^5$ and $t = 2.5 \times 10^5$

data is considered to arrive in stream. Data streams take the form of time series providing the values of the signals measured on a given process at each sampled time. When data arrives, a preprocessing stage finds the underlying trend by using polynomial fit. Using the found polynomial coefficients as trend context TC and adding some set of auxiliary variables AV the signal can be represented in a time interval T_i as $e(x_i) = \{TC, AV, T_i\}$. $e(x_i)$ is called a *episode* of the signal x_i . As an example the episodes found by *DyClee* for the cold water flow in the tested scenario at $t = [0, 2.5 \times 10^5]sec$ are shown in Figure 4.

The distance-based clustering stage groups episodes into μ -clusters that are summarized representations of the data statistical and temporal information. The definition of μ -cluster is given in (Barbosa Roa et al., 2016).

In order to maintain an up-to-date structure allowing to track system evolution, μ -clusters are weighted with a forgetting function. Figure 5 show two snapshots of the μ -clusters distribution as 3d-representation choosing the

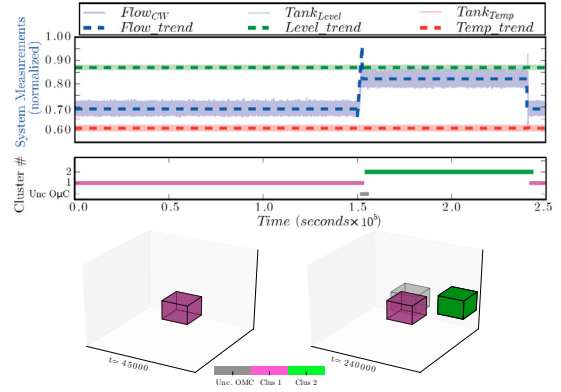


Fig. 6. Output of the density based stage at $t = 2.5 \times 10^5$

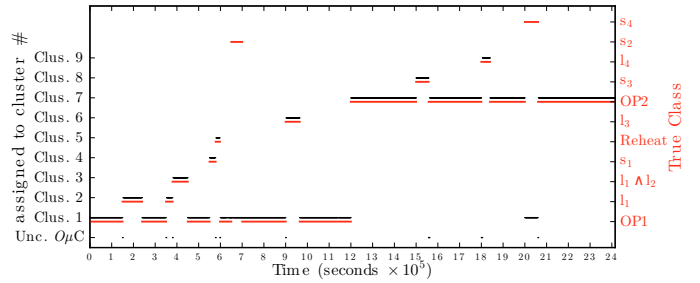


Fig. 7. *DyClee* clustering results reported in (Barbosa Roa et al., 2016) vs Scenario true classes

features with the biggest variance. The μ -cluster opacity represents its density.

The density-based stage analyses the distribution of those μ -clusters whose density is considered as medium or high and creates the final clusters by a density based approach, that is, dense μ -clusters that are close enough (connected) are said to belong to the same cluster. A μ -cluster is qualified as one of three options: dense μ -cluster ($D\mu C$), semi-dense μ -cluster ($S\mu C$) or low density (outlier) $O\mu C$ depending on whether its density exceed two chosen thresholds: the average and the median density of a μ -cluster group G_k .

The results of this density based stage at $t = 2.5 \times 10^5$ are illustrated in the Figure 6. This stage assigns the label to each μ -cluster, allowing the visualization of the results. Note that *DyClee* works under a non-supervised learning paradigm, so, only the system measures are necessary as input to the algorithm.

Figure 7 compares the classes, i.e. operation modes, discovered by *DyClee* and the true operation modes. The label associated to these clusters is related to their order of detection, and 'Unc. $O\mu C$ ' represents the non-representative behavior caused by extremely noised samples or by transition states (outliers). There is full concordance with true operation modes, except for operation modes s_2 (non observable) and s_4 that are confused with $OP1$. This result is not actually false because the later is indeed undiscriminable from operation mode $OP1$.

4. BUILDING DES FROM DYNAMIC CLUSTERING

To improve the diagnostic capabilities of *DyClee* a DES is built from its clustering results and adding the information

about the desired OP , provided in the system log files. In industrial environments, the set of set-points that form an OP is sometimes equal or bigger in size than the set of output system measurements. Considering the size of the set-points vector and under Hypothesis 3, we consider that adding all these variables to the clustering stage would increase its complexity and reduce its efficiency. Instead, we propose to use the information of the system OP as a discrete variable representing the system discrete state (under Hypothesis 2).

Timed automata (with guards), as defined below, are chosen as DES formalism. With this choice we aim at improve diagnosability by adding to the already modeled time-driven dynamics new event-driven dynamics.

Definition 2. (Timed Automaton, Alur and Dill (1994)). A time automaton \mathcal{A}_t is a tuple $\langle E, \mathcal{S}, \mathcal{S}_0, Tf_t, \mathcal{S}_m, C \rangle$ where S is a finite set of states, E is a finite set of events, $\mathcal{S}_0 \subseteq S$ is a finite set of start states, $Tf_t : \mathcal{S} \times E \times 2^{\mathcal{CK}} \times \Upsilon(\mathcal{CK}) \rightarrow \mathcal{S}$ is the *timed transition function*. \mathcal{S}_m is the set of accepted or final states and \mathcal{CK} is a finite set of clocks.

When the automaton is in the state s and the event a is detected, the automaton changes state from s to s' iff the clock constraint δ is fulfilled, that is the tuple $\langle s, a, \iota, \delta \rangle \in Tf_t$. The subset $\iota \subseteq C$ makes the clocks to be reset with this transition.

The state of the system is now given by a tuple formed by the clusterer state (provided by the cluster label and representing the continuous behavior) and the discrete state (provided by the OP) of the system.

In this paper \mathcal{A}_t is built automatically from the clusterer transition matrix found directly from *DyClee* clustering results in the following way:

- E : The system arrivals and departures in and from clusters are used as events, i.e. for each characterized cluster Cl_i an arrival event $Clus_i$ and a departure event \overline{Clus}_i are created.
- \mathcal{S}^0 : The set of states of the automaton is given by the set of natural clusters. This set is incremented if new clusters are created.
- \mathcal{S}_0 : The initial state is the first characterized cluster.
- Tf_t : The transition function is constructed incrementally as new clusters are detected. The time constraints are described as time intervals $[\underline{\tau}_{ij}, \bar{\tau}_{ij}]$ representing the minimum and maximum time of transition between two clusters Cl_i and Cl_j ². This interval is updated as more transitions are observed.
- \mathcal{S}_m : The DES implementation estimates the set of accepted states.
- \mathcal{CK} : One clock is created for each characterized cluster. Clock notation is cc_i , with i been the index of Cl_i .

4.1 Adding information to the timed automaton

DyClee clustering results include temporal information that is included in the Timed Automata formalism. The transition table columns col_i and rows r_i correspond to the

² This time interval corresponds to the time in which *DyClee* tracks the current state as transitional or unknown, i.e. the system measurements are considered as low density samples and assigned to $\odot\mu$ -cluster.

system states and each cell ij stores a transition interval $[\underline{\tau}_{ij}, \bar{\tau}_{ij}]$ characterizing the time constraints $cc_i \geq \underline{\tau}_{ij}$ and $cc_i \leq \bar{\tau}_{ij}$. The diagonal of the matrix (cell ij with $i = j$) include the minimum and maximum \mathcal{D}_i registered values. To reset each clock a set of complementary states \mathcal{S}^c is added to \mathcal{S}^0 , $\mathcal{S} = \mathcal{S}^0 \cup \mathcal{S}^c$. These complementary states represent the fact of *not been in* a cluster. The clock cc_i is set to zero (denoted as $cc_i := 0$) when the \overline{Clus}_i event is detected and the system goes from state Cl_i to state $\sim i$, with $\sim i \in \mathcal{S}^c$.

Another temporal information that is important in process monitoring refers to how long the system remains in a particular state. This time is denoted as \mathcal{D}_i . To better describe this information the median of the past stays is added to the timed automaton \mathcal{A}_t . The Cl_i cluster's median time of stay is called $\hat{\mathcal{D}}_i$.

Reflecting clusterer evolution in the \mathcal{A}_t is crucial to keep the system automaton updated. The following explains how parametric and structural clusterer changes are represented in the DES.

Modeling structural changes To handle the structural changes in the clusterer the following procedures are used:

Cluster creation When the k^{th} cluster is created the following changes are performed in the \mathcal{A}_t automatically:

- the state Cl_k and the state $\sim k$ are added to \mathcal{S}
- the counter cc_k is added to \mathcal{CK}
- a k^{th} row is added to the transition matrix
- a k^{th} column is added to the transition matrix
- $Clus_k$ and \overline{Clus}_k are added to the event set E

Cluster elimination When a cluster is eliminated, the state remains in the \mathcal{A}_t but its graphical representation changes to a smaller gray circle, which intuitively places the cluster as old.

Cluster merge When two or more clusters are merged, the transition matrix is rebuilt using the union of both states. In the case that both states have arriving transitions departing from the same state the time constraints are set as the interval union of those of the previous transitions. The same principle applies to transitions departing from the old states to a same destination. It is worth noting that the label assigned to a merged cluster, corresponds to the oldest original label.

Cluster split When a cluster is split into two or more clusters, the transitions are recalculated from available history (recent data). The transitions that cannot be confirmed by looking at available history are depicted as dotted lines until confirmed or deprecated. A transition is deprecated if in a time-span corresponding to three times the sum of all elements in $\hat{\mathcal{D}} = \cup_i \hat{\mathcal{D}}_i$. Deprecated transitions do not cause loss of information under hypothesis 4, below:

Hypothesis 4. A system is considered as a regular system, i.e. the system behavior is repeatable. Any possible transition from a system state to another repeats with some frequency.

Modeling parametric changes Changes in the cluster parameters are not reflected directly in the DES, nevertheless, since they are also important to the system operator,

Table 2. Final labels found pairing the continuous and discrete information

Cluster	<i>a</i>	New label	Cluster	<i>a</i>	New label
1	<i>OP1</i>	1	5	<i>OP1</i>	5
	<i>OP2</i>	1'	6	<i>OP1</i>	6
2	<i>OP1</i>	2	7	<i>OP2</i>	7
3	<i>OP1</i>	3	8	<i>OP2</i>	8
4	<i>OP1</i>	4	9	<i>OP2</i>	9

Table 3. Transition table of the simulated scenario

From / To	<i>Cl</i> ₁	<i>Cl</i> ₂	<i>Cl</i> ₃	<i>Cl</i> ₄	<i>Cl</i> ₅	<i>Cl</i> ₆	<i>Cl</i> ₇	<i>Cl</i> ₈	<i>Cl</i> ₉	<i>Cl</i> ₁₀
<i>Cl</i> ₁	# [97500, 297500]	2	0	1	0	1	1	0	0	0
<i>TI</i>	[2500, 10000]	[-]	[-]	[20000]	[-]	[10000]	[40000]	[-]	[-]	[-]
<i>Cl</i> ₂	# 1	[25000, 77500]	1	0	0	0	0	0	0	0
<i>TI</i>	[0,0]	[-]	[10000]	[-]	[-]	[-]	[-]	[-]	[-]	[-]
<i>Cl</i> ₃	# 1	0	[57500]	0	0	0	0	0	0	0
<i>TI</i>	[0,0]	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[-]
<i>Cl</i> ₄	# 0	0	0	[5000]	1	0	0	0	0	0
<i>TI</i>	[-]	[-]	[-]	[-]	[12500]	[-]	[-]	[-]	[-]	[-]
<i>Cl</i> ₅	# 1	0	0	0	[5000]	0	0	0	0	0
<i>TI</i>	[2500]	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[-]
<i>Cl</i> ₆	# 1	0	0	0	0	0	0	0	0	0
<i>TI</i>	[0,0]	[-]	[-]	[-]	[-]	[52500]	[-]	[-]	[-]	[-]
<i>Cl</i> ₇	# 0	0	0	0	0	0	[157500, 352500]	1	1	1
<i>TI</i>	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[40000]	[10000]	[0,0]
<i>Cl</i> ₈	# 0	0	0	0	0	0	0	1	1	1
<i>TI</i>	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[5000]	[15000]	9
<i>Cl</i> ₉	# 0	0	0	0	0	0	0	0	[27500]	0
<i>TI</i>	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[0,0]	[-]	[-]
<i>Cl</i> ₁₀	# 0	0	0	0	0	0	0	0	0	[57500]
<i>TI</i>	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[2500]	[-]	[-]

a descriptive table including the following information is also available.

- Time of last assignation of system data to any μ -clusters in cluster k . This indicates the last time in which the cluster was active.
- Number of μ -clusters conforming the cluster.
- Median of cluster densities.
- Cluster center of gravity calculated, for each of the j features, as:

$$C_g^j = \frac{\sum_k C_k^j * D_k}{\sum_k D_k} \quad (1)$$

- Minimum of the feature ranges for the cluster
- Maximum of the feature ranges for the cluster

Cluster drift can be easily detected by the change in the cluster center of gravity C_g and/or an augmented number of μ -cluster.

5. IMPROVING DIAGNOSABILITY

As shown in Figure 7, **DyClee** can successfully track online the process and its evolution, however, the faulty state s_4 (hot water valve stuck) is confused with *OP1*, since both have the same observable behavior. The key information that allows to distinguish between these two behaviors is the system desired *OP* represented by the discrete state.

As stated in section 4 in order to build a timed automaton from **DyClee** clustering results, the clusterer states have to be paired with the discrete states. The association is made using the time as key. Table 2 shows the pairs and the new labels. The transition table considers the new cluster labels. Even if the \mathcal{A}_t is generated on-line at the same time as the clustering results, for the sake of brevity only the final results on the scenario are going to be depicted. The transition table generated for the tested scenario is shown in Table 3.

Figure 8 represents the generated automaton \mathcal{A}_t at the end of the simulated Scenario. The states are represented as nodes. The initial state is determined as Cl_1 (the first encountered state). The transition functions correspond to all the non-empty cells in the transition matrix and

are represented as edges in Figure 8. To determine the set of accepted states, the information about the number of *OPs* is used, so, in this case there are two accepted states, namely Cl_1 and Cl_2 (double-circle representation).

The only events depicted explicitly are the *OP*, nevertheless each transition from a Cl_i state to a $\sim i$ state implies the $\overline{Clus_i}$ event and each transition from a $\sim i$ state to a Cl_j state implies a $Clus_j$ event. Each cluster state includes, below the cluster identification, the information related to the median time that the system had remained in each of these states. As stated in section 4 one clock is created for each state. Clock constraints (transitions from $\sim i$ to Cl_j) and resets (transitions from in Cl_i to $\sim i$) are included in the graph according to the notation explained before. The probability of a transition is represented as $P_{i \rightarrow j}$. A red colored font indicates the current state. The orange polygon form depicts the node that could not have been generated using only the clustering results. In this node the information of the clusterer state is also depicted as reference.

Summarizing:

- $E: \{Clus_1, Clus_2, Clus_3, Clus_4, Clus_5, Clus_6, Clus_7, Clus_8, Clus_9, Clus_{10}, Clus_{11}, Clus_{12}, Clus_{13}, Clus_{14}, Clus_{15}, Clus_{16}, Clus_{17}, Clus_{18}, Clus_{19}, Clus_{20}, OP1, OP2\}$
- $S: \{Cl_1, Cl_2, Cl_3, Cl_4, Cl_5, Cl_6, Cl_7, Cl_8, Cl_9, Cl_{1'}, \sim 1, \sim 2, \sim 3, \sim 4, \sim 5, \sim 6, \sim 7, \sim 8, \sim 9, \sim 1'\}$
- $S_0: \{Cl_1\}$
- $\langle Cl_1, (\overline{Clus_1}), cc_1, \rangle \rightarrow \sim 1$
- $\langle \sim 1, (Clus_7, OP2), cc_1 \in [40000, 40000] \rangle \rightarrow Cl_7$
- $Tf_i: \langle \sim 1, (Clus_2, OP1), cc_1 \in [2500, 10000] \rangle \rightarrow Cl_2$
- $\langle \sim 7, (Clus_1, OP2), \rangle \rightarrow Cl_{1'}$
- ...
- $S_m: \{Clus_1, Clus_7\}$
- $CK: \{cc_1, cc_2, cc_3, cc_4, cc_5, cc_6, cc_7, cc_8, cc_9, cc_{1'}\}$

It can be seen that the cluster Cl_1 (*OP1*) is the focus of the graph having transitions to states Cl_2, Cl_4, Cl_6 and Cl_7 . The graph also shows that the state Cl_5 is only accessible from Cl_4 and that Cl_3 only from Cl_2 . This indicates strong dependencies. From a point of view of supervision these transitions indicate that the faulty states Cl_3 and Cl_5 are a degradation of already degraded states Cl_2 and Cl_4 . The system current state at the end of the simulated scenario is Cl_7 , meaning that the system is currently in *OP2* and no faulty behavior is currently detected. The state $Cl_{1'}$ that has the same continuous behavior as Cl_1 is now diagnosable since it is only accessible from Cl_7 when the system is in *OP2*, as opposed to Cl_1 .

6. CONCLUSIONS

In this paper we show that dynamic clustering results provided by **DyClee** can be used to learn a timed discrete event system that brings valuable temporal information to distinguish behaviors that are non-diagnosable based solely on the clustering itself. This automaton also improves clustering results interpretability for decision-making purposes. This adaptive model, built automatically as new data of the system is gathered, provides a high level abstraction of the system with information about the system past and current states along with the possible transitions identified by their probabilities, the time of transition and time constraints. The \mathcal{A}_t is a very

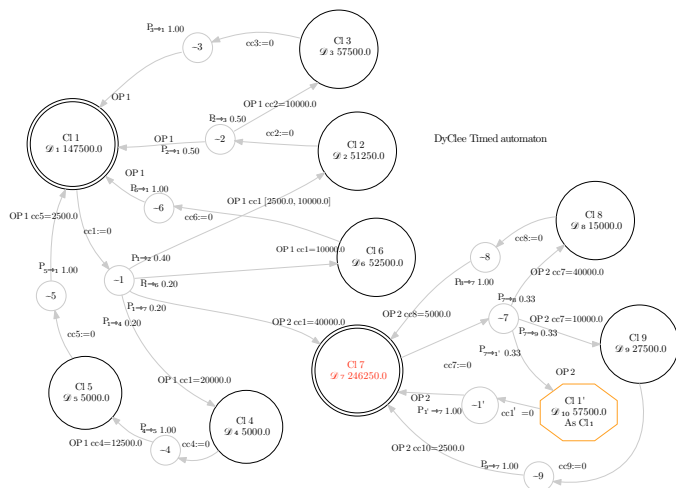


Fig. 8. *DyClee* timed automaton generated from clustering results

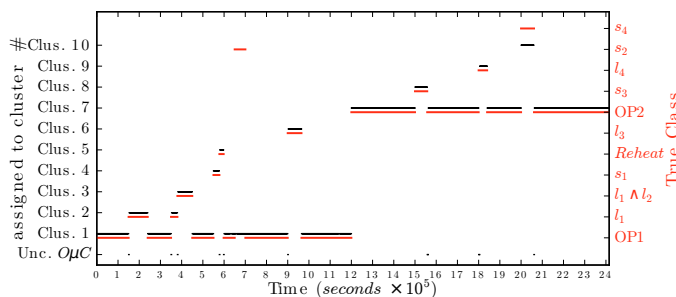


Fig. 9. Final diagnosis achieved by coupling continuous and discrete information

interesting piece of information on its own as it may be used to support other tasks, like control, reachability analysis, etc. Future work will consider to extend the present clustering/ \mathcal{A}_t framework to distributed architectures that are better suited to represent complex power plants in which the dynamics of the different processes interact.

REFERENCES

- Aggarwal, C.C., Han, J., Wang, J., and Yu, P.S. (2003). A framework for clustering evolving data streams. In *Proc. of the 29th Int. Conf. on Very large data bases*, Vol. 29, 81–92.
- Alur, R. and Dill, D.L. (1994). A theory of timed automata. *Theoretical Computer Science*, 126(2), 183–235.
- Angelov, P. (2011). Fuzzily connected multimodel systems evolving autonomously from data streams. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 41(4), 898–910.
- Barbosa Roa, N., Travé-Massuyès, L., and Grisales, V.H. (2015). A data-based dynamic classification technique: A two-stage density approach. In *Proc. of the 9th IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes Safeprocess'15, Paris (France)*, 1224–1231. IFAC.
- Barbosa Roa, N., Travé-Massuyès, L., and Grisales, V.H. (2016). Dynamic clustering as a tool for monitoring evolving systems. In *Proc. of the 27th Int. Workshop on Principles of Diagnosis DX'16, Denver (USA)*. DX. Accepted.
- Bayouduh, M. and Travé-Massuyès, L. (2014). Diagnosability analysis of hybrid systems cast in a discrete-event framework. *Discrete Event Dynamic Systems*, 24(3), 309–338.
- Bayouduh, M., Travé-Massuyès, L., and Olive, X. (2008). Hybrid systems diagnosis by coupling continuous and discrete event techniques. *Proc. of the 17th IFAC World Congress, Seoul (Corea)*, 41(2), 7265–7270.
- Console, L., Picardi, C., and Ribando, M. (2000). Diagnosis and diagnosability analysis using process algebra. In *Proc. of the 11th Int. Workshop on Principles of Diagnosis DX'00, Morelia (Mexico)*, 25–32.
- Cordier, M.O., Grastien, A., et al. (2007). Exploiting independence in a decentralised and incremental approach of diagnosis. In *Proc. of the 20th Int. Joint Conf. on AI IJCAI'07, Hyderabad (India)*, 292–297.
- Gaudel, Q., Chanthery, E., and Ribot, P. (2015). Hybrid particle petri nets for systems health monitoring under uncertainty. *Int. J. of Prognostics and Health Management*, 6.
- Kempowsky, T., Subias, A., and Aguilar-Martín, J. (2006). Process situation assessment: From a fuzzy partition to a finite state machine. *Engineering Applications of Artificial Intelligence*, 19(5), 461–477.
- Kwak, J., Lee, T., and Kim, C.O. (2015). An incremental clustering-based fault detection algorithm for class-imbalanced process data. *IEEE Transactions on Semiconductor Manufacturing*, 28(3), 318–328.
- Lin, F. and Lin, T.W. (1993). Diagnosability of discrete event systems and its applications to circuit testing. In *Proc. of 36th Midwest Symp. on Circuits and Systems, Detroit (USA)*, 344–347 vol.1.
- Pencolé, Y. and Cordier, M.O. (2005). A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence*, 164(1), 121–170.
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on automatic control*, 40(9), 1555–1575.
- Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D.C. (1996). Failure diagnosis using discrete-event models. *IEEE transactions on control systems technology*, 4(2), 105–124.
- Thornhill, N.F., Patwardhan, S.C., and Shah, S.L. (2008). A continuous stirred tank heater simulation model with applications. *Journal of Process Control*, 18(3), 347–360.
- Travé-Massuyès, L., Cordier, M.O., and Pucel, X. (2006). Comparing diagnosability in CS and DES. *Proc. of the 6th IFAC Symp. on Fault Detection, Supervision and Safety for Technical Processes Safeprocess'06, Beijing (China)*, 39(13), 1163–1168.
- Travé-Massuyès, L., Escobet, T., and Milne, R. (2001). Model-based diagnosability and sensor placement application to a frame 6 gas turbine subsystem. In *Proc. of the 17th Int. Joint Conf. on AI IJCAI'01, Vol.1*, 551–556. Morgan Kaufmann Publishers Inc.
- Travé-Massuyès, L., Escobet, T., and Olive, X. (2006). Diagnosability analysis based on component-supported analytical redundancy relations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(6), 1146–1160.