Discriminating qualitative model generation from classified data

A. Subias, L. Travé-Massuyès[1]

LAAS-CNRS, Université de Toulouse, 7 avenue Colonel Roche 31077 Toulouse Cédex 4 FRANCE
{subias,louise}@laas.fr

Abstract

Modeling is quite critical and remains a bottleneck for model-based diagnosis in many application domains. Quantitative models that are developed during the design stage are not applicable as so to model-based diagnosis engines. This paper proposes to take advantage of discretization algorithms used by the machine learning community to discretize the domain value of continuous variables and generate a behavioral qualitative model from the data clusters corresponding to classified data. The results of this approach are illustrated and discussed with the two tanks benchmark example.

Introduction

Nowadays, model-based diagnosis methods are generally opposed to data-based methods. These later rely on machine learning methods which are able to exhibit classes of different behavior directly from the data. The former assume the availability of a (behavioral) model of the system that can be used to predict what should be observed. The model is hence used as a reference: any discrepancy indicates a problem and the knowledge of the components involved in the prediction leading to discrepancies can be used to retrieve the cause of the problem. The advantages of the model-based approaches are several, in particular the model provides a representation of underlying knowledge that can support different tasks along the system's life cycle.

However, modeling is still a bottleneck in many application domains. In the last few years, the qualitative reasoning and model based communities have been working to find solutions to generate automatically models that could be applied to model-based diagnosis engines. But little has been done starting with the idea that machine learning methods could extract the model knowledge from the data.

This paper proposes to take advantage of discretization algorithms used by the machine learning community to discretize the domain value of continuous variables and generate a behavioral qualitative model from the data sets corresponding to classified data. This model must be discriminating in the sense that the data belonging to different behavioral modes, e.g. normal and faulty, must be distinguishable. The discretization algorithm should hence allow us to generate a set of relevant "landmarks", i.e. thresholds defining the variable value domains discretizations, for the different continuous variables involved in the relations exhibited by the classifier. The classifier that has being used for this purpose is the LAMDA classifier (Aguilar *et al.* 1982). The results of our experimental study have been obtained with the two tanks benchmark example.

The next section of this article gives the problem formulation and presents related works. Section 3 focuses on discretization algorithms. Then, Section 4 describes the proposed approach to generate a qualitative model for diagnosis. An application example is presented in Section 5. Concluding remarks are drawn in the last section.

Problem formulation and related work

Previous works of the qualitative reasoning community about qualitative model generation were mainly performed in the framework of the IDD (Integrated Design process for on-board Diagnosis) European project (Brignolo *et al.* 2001). They start with the idea that most engineering models developed during the design stage are in the form of MATLAB/Simulink models. Hence these models are available and different runs can be performed to exhibit the qualitative input-output relationships of a qualitative model.

Most of the proposed methods rely on monotonicity properties of the output functions and require to be able to identify the monotonicity domains, which is far from trivial for non linear systems. The results hence seem restricted to particular cases.

(Console *et al.* 2003) proposes to derive qualitative deviation models. Deviation models are a special type of qualitative model that state the relationships between the variable deviations. Deviations represent the difference between the value of a variable an its expected normal value, e.g. $\Delta x = x - x_{ref}$ for a variable x. Qualitative deviations proceed to a sign abstraction and hence represent whether the value is equal, lower or higher than expected, i.e. $[\Delta x] = sign(x - x_{ref})$. This approach requires to identify the monotonicity domains of the output function, then proceeds to a series of simulations according to a fixed number of sampling points. The approach is not able to always conclude about the deviation sign and it does not scale up nicely to spaces of dimension higher than 2.

(Yan, 2003) (Yan *et al.* 2004) propose to derive input-output mapping qualitative models. Consider a system with a 2 dimensional observable space and assume that x and y are the input and output observable variables, then the system can be modelled by a function of type $y = f(x)$.

[1] Authors are listed in alphabetical order.

Behavioral modes are characterized by a set of input-output tuples $((x_i,x_{i+1}),(y_j, y_{j+1}))$, where x_i, i=1, …, n and y_j, j=1, …, m are the landmarks defining the discretization of the value domains of x and y respectively (cf. figure 1). This description easily extends to the n-dimensional case $Y = f(X)$, where X and Y are then input and output vectors.

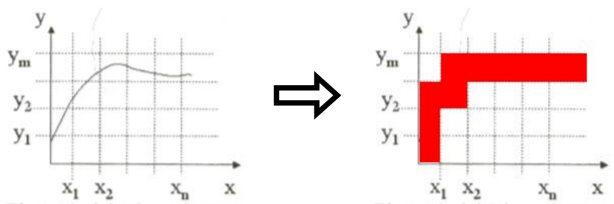

Figure 1: Input-output mapping qualitative model (from (Yan 2003))

For robustness purposes, (Struss 2002) prefers to proceed to the qualitative abstraction for a behavior envelope. However the behavior envelope is still bounded by two functions, as shown in figure 2.

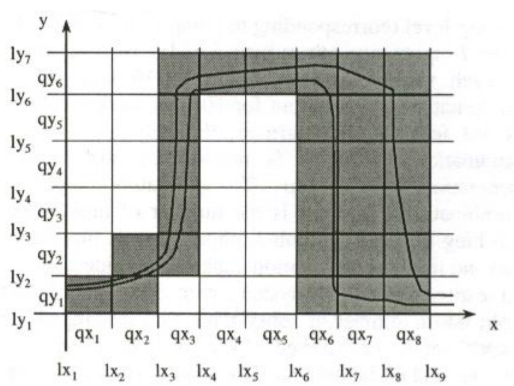

Figure 2: Abstraction of an envelope behavior (from (Struss 2002))

The problem is formulated as the one of determining the set of landmarks for observable variables domains such that a given behavior mode M1 is discriminable from another behavior mode M2, i.e. there exists at least one input interval (x_i,x_{i+1}) for which the sets of corresponding output tuples for M1 and M2 are different. Hence, the partitioning must introduce a minimal number of landmarks while being sufficiently tight to discriminate all behavioral modes. The resulting qualitative model is said to be *discriminating*. The discriminability problem is illustrated in figure 3.

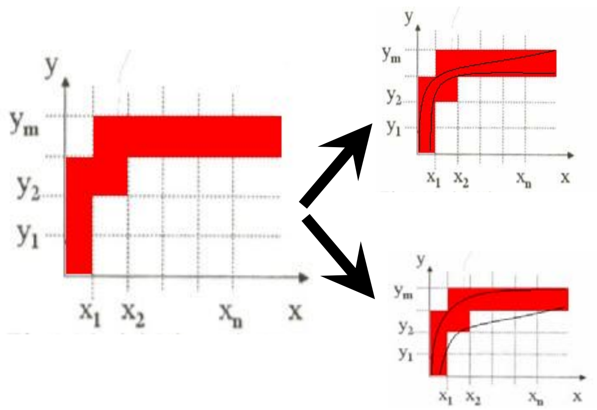

Figure 3: Discriminability problem

In (Yan, 2003) and (Yan *et al.* 2004) the qualitative model generation proceeds in two steps: the value domain of each variable is discretized in an arbitrary way (in general with a predefined number of intervals of equal length) and for each input region, the output regions hit by the output function or behavior beam are determined. These approaches suffer from the same scaling up problem as (Console *et al.* 2003).

Making use of discretization algorithms

Machine learning algorithms are used to extract knowledge from databases. Most algorithms can only be applied to data described by discrete and symbolic attributes. In case of continuous attributes, this community has proposed several so-called *discretization algorithms* that transform continuous attributes into ordered discrete ones. These algorithms aim at speeding up the generation of decision trees, resulting in smaller and hence more easily interpretable decision trees, by finding the discretization with minimum number of intervals that minimises information loss.

The above considerations reveal that discretization algorithms can as well be applied to our problem, provided data is available. Unlike the approaches presented in the previous section, we do not require the availability of a MATLAB/Simulink type simulation model but propose to extract the knowledge directly from historical recorded data. These data is first applied to a classification algorithm that is able to distinguish relevant classes corresponding to different behavior patterns of the system. The classified data is then applied to a discretization algorithm.

We paid particular attention to two discretization algorithms, namely the CADD (Class Attribute Dependant Discretizer) algorithm (Ching *et al.* 1995), and the CAIM (Class Attribute Interdependence Maximization) algorithm (Kurgan and Cios 2004). These algorithms apply without modification to problems of any dimension.

CADD is based on a criterion called CAIR (Class Attribute Interdependence Redundancy), which is built on the notion of class-attribute joint entropy and class-attribute mutual information. On the other hand, CAIM is based on a more intuitive criterion. Both algorithms make use of heuristics to cut the computational cost.

Building a decision tree starts with classified data, i.e. each data sample belong to a similarity class. Let us consider a population of M data samples distributed in S classes C_i and consider that the value domain of an attribute A_j is partitioned into n discrete intervals $[d_0 ; d_1]$,$]d_1 ; d_2]$,…,$]d_{n-1} ; d_n]$. For this attribute, the following table, which associates the number of samples of the population representing the attribute (noted q_{ij}) to every class-interval couple, is generated.

	$[d_0 ;d_1]$	$]d_{r-1} ;d_r]$	$]d_{n-1} ;d_n]$	
C_1	q_{11} . . .	q_{1r} . . .	q_{1n}	M_{1+}
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
C_i	q_{i1} . . .	q_{ir} . . .	q_{in}	M_{i+}
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
C_s	q_{s1} . . .	q_{sr} . . .	q_{sn}	M_{s+}
	M_{+1}	M_{+r}	M_{+n}	M

In the table the following notations are used:
M_{i+}: number of samples belonging to class C_i
M_{+r}: number of samples whose attribute A_j has its value in $]d_{r-1} ; d_r]$

The following « probabilities » are then defined:
Probability of a sample to belong to class C_i :

$p_{i+} = p(C_i) = M_{i+} / M$

Probability of a sample to belong to interval $]d_{r-1} ; d_r]$:
$p_{+r} = p(]d_{r-1} ; d_r]) = M_{+r} / M$

Probability of a sample to belong to class C_i and interval $]d_{r-1} ; d_r]$:
$p_{ir} = p(C_i,]d_{r-1} ; d_r]) = q_{ir} / M$

The CADD Algorithm

In the CADD algorithm (Ching, Wong, and Chan 1995), the criterion to be maximized (CAIR) is the quotient of the class-attribute mutual information and the Shannon entropy:

Class-Attribute Interdependence Redundancy (CAIR) :

$$R(C, A_j) = \frac{I(C, A_j)}{H(C, A_j)}$$

$$I(C, A_j) = \sum_{i=1}^{S} \sum_{r=1}^{n} p_{ir} \log_2 \left(\frac{p_{ir}}{p_{i+} \cdot p_{+r}} \right) \quad \text{Class-Attribute mutual information}$$

$$H(C, A_j) = \sum_{i=1}^{S} \sum_{r=1}^{n} p_{ir} \log_2 \frac{1}{p_{ir}} \quad \text{Shannon Entropy}$$

The class-attribute mutual information alone could have been chosen as a criterion but its value depends on the number of intervals. It is hence not suitable for minimising the number of intervals. $I(C, A_j)$ increases with the number of intervals but this increase is compensated by the increase of $H(C,A_j)$ in the CAIR criterion.

The CADD algorithm proceeds in three steps:
1- Initialisation
1.1- Sort the values taken by the attribute in increasing order and distribute them in n intervals (the number of initial intervals is defined by the user or by another criterion)
The initial partition must distribute the values equally to minimise the information loss. A max entropy criterion is generally chosen.
1.2- Build the table
1.3- Calculate the criterion CAIR.

2- Perturbation step
2.1- Improve CAIR by introducing perturbations on the landmarks of the initial partition (increase or decrease the landmarks). Select the perturbation that most improves CAIR. Iterate until no more CAIR improvement.

3- Merge intervals
Merge the intervals that do not contribute significantly enough to the interdependence criterion by proceeding to the following test:

$$R(C, A_j) \geq \frac{c^2_{(s-1)(Lj-1)}}{2 . M . H(C, A_j)}$$

If the test is TRUE: move to the next adjacent intervals
ELSE: merge the two intervals.

The CAIM Algorithm

The CAIM algorithm (Kurgan and Cios 2004) is more recent and aims at reducing the computation time. It is also based on a heuristic but the criterion is simpler:

$$CAIM(C, A_j) = \frac{\sum_{r=1}^{n} \frac{\max_r^2}{M_{+r}}}{n}$$

with $\max_r = \text{Max}(q_{ir} \mid i = 1,…, S)$.

It is easy to notice that when the criterion increases, the class-attribute interdependence (correlation between the interval partitioning and the classes) also increases. Dividing by M_{+r} has two goals: on the one hand it permits to account for the negative influence of the samples belonging to other classes than the leading one (class most represented in a given interval), on the other hand it avoids a possible excess error when evaluating:

$$\frac{\max_r^2}{M_{+r}}$$

in the case of a high number of samples and reduced work memory. Finally, dividing by n is in favor of minimizing the number of intervals of the discretization.

The CAIM algorithm can be summarized as follows:
For each attribute A_j do
Step 1
1.1- Find the max and min values d_n and d_0
1.2- Sort the values of A_j in increasing order and find all the possible landmarks in between d_0 and d_n. Stack them in *B*.
1.3- Define $D = \{[d_0 ; d_n]\}$ and GlobalCAIM = 0.

Step 2
2.1- $k = 1$
2.2- Try to insert a landmark of *B* in *D* and determine the corresponding CAIM ;
2.3- After trying all possibilities, select the one with the highest CAIM ;
2.4- If (CAIM > GlobalCAIM or $k < S$) update D with the selected landmark and do GlobalCAIM = CAIM
Else end
$k = k+1$ and go to 2.2

The algorithm returns a discretization *D* that contains all the selected landmarks.

Qualitative model generation for diagnosis

Our objective is to generate automatically a behavioral model of the process that can be applied to model-diagnosis engines. The challenge is to generate a discriminating model able to distinguish the different behavioral modes of the system. The aim is to clearly distinguish normal from faulty behaviors and to characterize each mode through the generation of relevant landmarks associated to each state variable involved in the process evolution and considered by the classifier.

This approach can be viewed as an attempt to establish a bridge between data-based approaches and model-based approaches for diagnosis (see figure 4).

Indeed, the use of data mining techniques along with expert knowledge is commonly used for the diagnosis of complex processes. In such systems, an increased level of complexity and automation overloads the role of human operators who must perform not only physical tasks but also, and mainly, cognitive tasks.

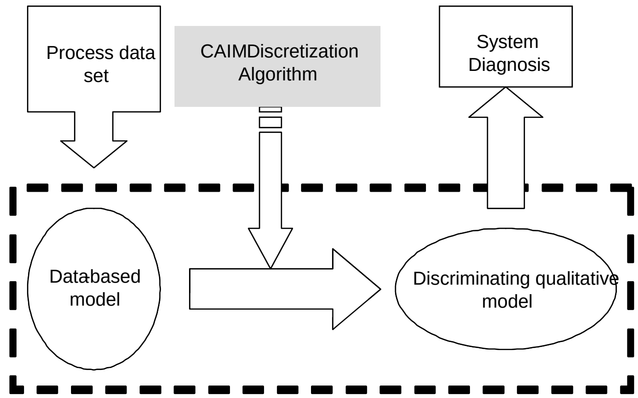

Figure 4: A bridge between Data-Based and Model-Based diagnosis approaches

These techniques permit the identification of the different functional states, normal and/or abnormal, of a complex process. Data mining techniques are applied to extract information from historical data records. The observation space is defined by the selection of a set of representative variables which best characterize the process functional states. The goal is to perform the partition of the observation space to represent and discriminate the classes associated to normal and/or abnormal situations from a series of process data. This partition is defined as a learning stage, where according to the available information of the process, the expert may totally or partially associate the training data set into different classes. Once the known situations have been characterized, the monitoring task is to associate a new observation with one of the expected process behaviors, that is equivalent to assign the element obtained by the new observations to a learned class. Any unrecognized observation corresponds to a deviation from the expected behavior and leads to a symptom detection. This detection procedure based on such a discrepancy principle allows taking into account abnormal situations due to real failures on the process sensors and/or actuators but also unexpected situations which correspond to a normal operating of the process not considered in the elaboration of the behavior pattern. The capability to detect abnormal states must be completed with the cause's identification of these abnormal situations which corresponds to fault diagnosis.

Therefore, in such approaches, the diagnosis is mainly performed by the process operators to which meaningful information must be presented in a convenient way such that it is easily interpretable. The automatic generation of a behavioral model issued from classified data that can be applied to model-diagnosis engines is then an interesting issue. At this preliminary stage the focus was put on showing the feasibility of our approach rather than on a deep analysis of discretization algorithms and available classification/clustering methods. We hence decided to use a fuzzy method for conceptual clustering and classification developed in our research team. This method, called LAMDA, is based on finding the global membership degree of a data sample to an existing class, considering all the contributions of each of its attributes. A software tool called SALSA "Situation Assessment using LAMDA claSsification Algorithm" provides different learning strategies (supervised or not) and the user can obtain different partitions with the same data set, by changing various parameters to improve the quality of the final classification (Kempowsky *et al.* 2003).

The CAIM algorithm has been used for the discretization step having in mind the minimization of the number of intervals (Kurgan 2004).

Application to a two-tanks system

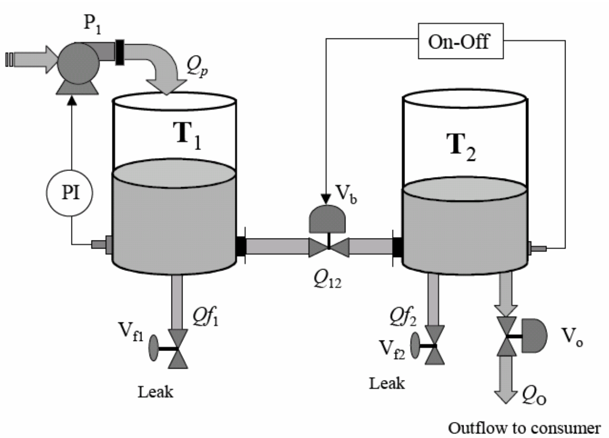

Figure 5: The two-tanks system
(from (Bouamana 2001))

The proposed approach has been applied on a simple system composed by two coupled tanks developed at the University of Lille (see figure 5). The two tanks are connected by a pipe and are controlled to provide a constant water flow Q_0. The water level in the tank T_1 (nominal value h_1=0,5m) is controlled by a PI controller via a pump P_1 providing the inlet flow Q_p. The water flow between the two tanks can be controlled by an on/off controller acting on a valve (V_b). The water level of tank T_2 must be kept at a medium level (0,09m≤$h2$≤0,11m). The valve V_0 is opened in nominal operating and simulates the water outflow to a consumer. Finally, the two valves V_{f1} et V_{f2} can be used to simulate failure situations (leakage in the tanks), these two valves are closed in faultless mode. A Matlab/Simulink model of the benchmark has been used to obtain the relevant data.

First, the normal behavior (faultless mode) has been simulated and observed through the two levels h_1 et h_2 (see figure 6).

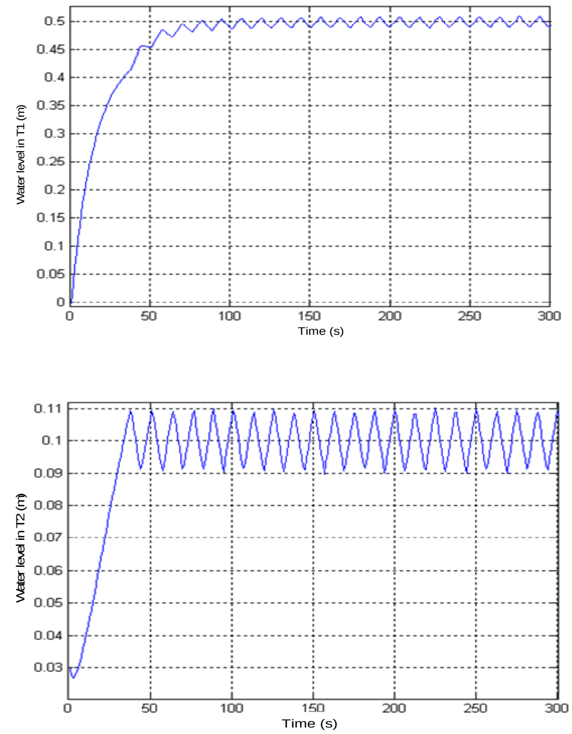

Figure 6: Faultless behavior of the system

Then, a failure scenario corresponding to a pump blockage (pump off: Q_p=0*)* has been simulated. In order to obtain relevant classification results, several blockages are sequentially simulated and used to generate a relevant data set for training.

In our case study, a training data set of 423 samples (individuals) representing three successive pump blockages has been used. Unsupervised learning is chosen since no prior knowledge about the possible situations is given. Because all descriptors are quantitative SALSA needs to normalize them to the unit interval [0,1] in such a way that they can be treated simultaneously. Figure 7b shows the behavior of the different variables (descriptors) chosen for the construction of the behavior pattern: h_1 (level in the tank T_1) , Q_p (flow of the pump P_1*)*and U_p (the output of the PI controller). SALSA initially identifies six different classes (one class per line) (figure 7a) that best represent the operation of the system. Each sample (each abscissa) is associated to one corresponding class. Classes C_1 and C2 respectively correspond to the faultless behavior of the system and the pump blockage, whereas classes C3 to C6 correspond to transitions states to normal operating.

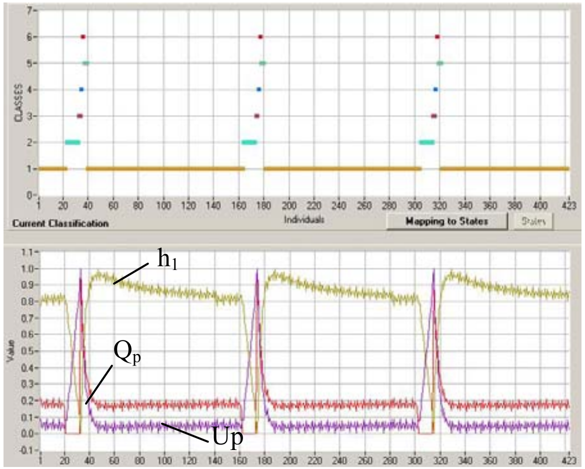

Figure 7. a : Training data: failure scenario: pump blockages b: Classification results.

Figure 8 gives a visualization of the classification results (classes C_1 to C_6) in the description space through group of dots. To a better visualization of the state space partition only the descriptors h_1 and Q_p are considered.

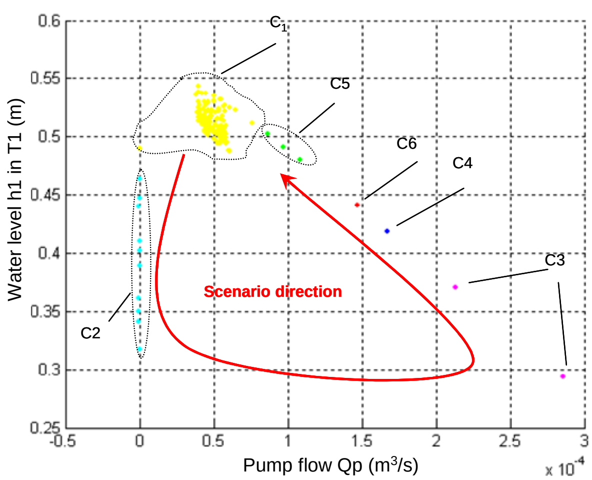

Figure 8: 2-D visualization of the classification results

The next step consists in using the CAIM discretization algorithm on this classification to generate a qualitative model. For a better readability, once again only a 2-D representation is shown in figure 9.

The result is questionable: the discretization of the pump flow Q_p variation space is relevant but the one of the level h_1 could be improved: the obtained intervals do not ensure the discrimination of the classes C_2 to C_6. Therefore, the behavioral modes of the system i.e. normal or faulty are not clearly distinguishable. This can be explained by the fact that the data set used for the classification mainly includes samples corresponding to the normal operating of the system (class C_1). As the criterion used in the discretization algorithm is quite similar to the entropy concept, the classes with few elements have less influence than the others.

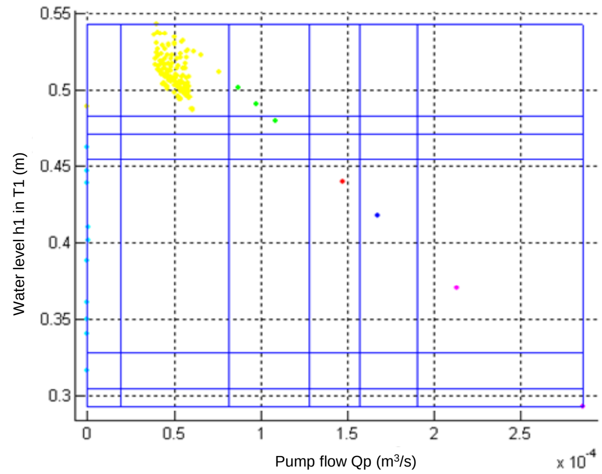

Figure 9: Discretization results

To deal with this discrimination aspect, another test has been performed based on a new failure scenario. This time, the failure scenario includes first a leakage of the tank T_1 and then a pump blockage. From the previous classification, a recognition procedure using the same descriptors has been launched via SALSA to obtain the resulting classification (figure 10).

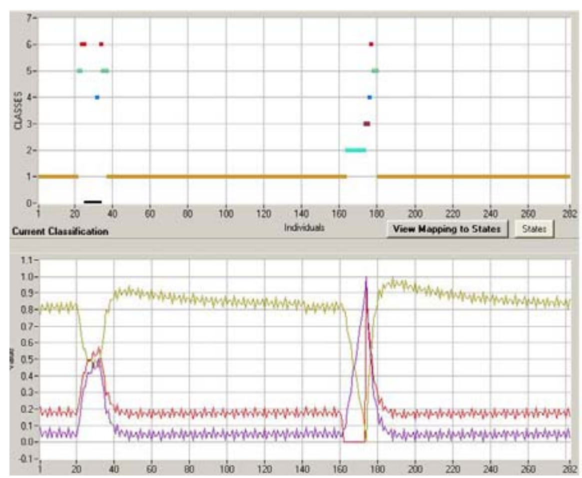

Figure 10: Classification results for scenario 2

The recognition mode of SALSA allows to reject the elements that cannot be assigned to any existing classes i.e. elements that do not correspond to any of the expected behaviors. These elements are assigned to the class C0 called the NIC (Non Informative Class) class without modifying the existing ones.

For the discretization step, the NIC is considered as any class. The results have been improved (figure 12) as the discrimination of the classes is now ensured.

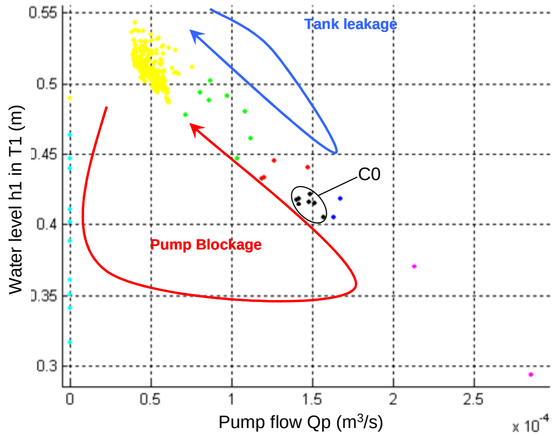

Figure 11: Classification results for scenario 2

By adding the leakage simulation in the failure scenario, new elements have been added to the classes with consequence to enhance the influence of these classes on the discretization criteria and therefore to obtain a more suitable interval decomposition (figure 12). The resulting qualitative model is shown in figure 13.

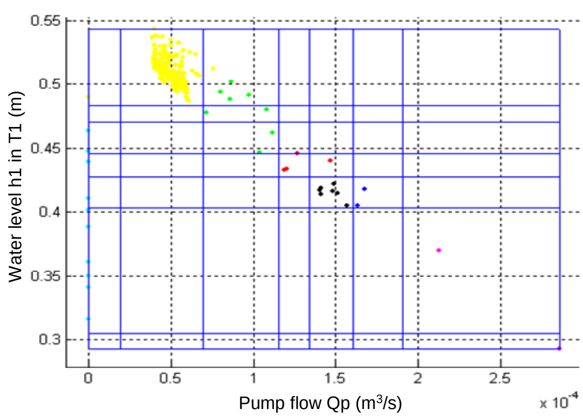

Figure 12: Discretization results for scenario 2

Conclusion and future work

This paper proposes a method to derive a discriminating qualitative model that can be use for model-based diagnosis. Unlike the other approaches of the Qualitative Reasoning community, the method does not assume the availability of a MATLAB/Simulink simulation model to generate the data but is able to extract the knowledge directly from historical data recorded on the real process.

These data is first applied to a classification algorithm that is able to distinguish relevant classes corresponding to different behavior patterns of the system. The classified data is then applied to a discretization algorithm.

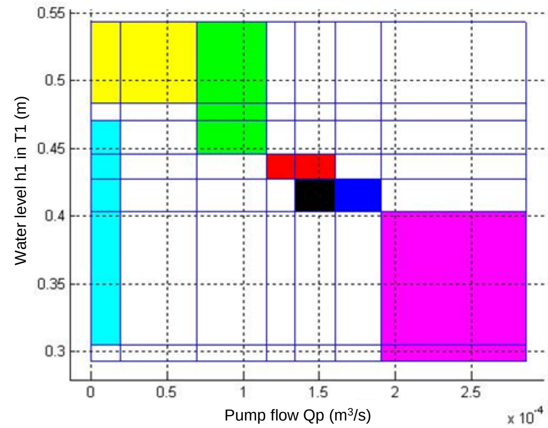

Figure 13: Resulting qualitative model

The experiments with the LAMDA classifier and the discretization algorithm CAIM performed on the two-tanks benchmark example show the feasibility of the approach.

Some work remains to be done to provide the qualitative model in an explicit form. Indeed, in the case of the second two-tanks scenario, the qualitative model is directly given by the obtained discretization, as illustrated by figure 13. However, in other cases, like the first two-tanks scenario, some landmarks indicated by the discretization are questionable and it may be necessary to proceed to a further filtering step for deriving the final qualitative model.

Another interesting issue is to establish the link between the obtained qualitative models and consistency relations as used in the model-based diagnosis approach. Consistency relations are relations that only involve observable variables and can hence be evaluated from the sensored observations. Consistency relations are also known as analytical redundancy relations (ARRs) when derived from a quantitative model (Cordier *et al.* 2004). From the hypothesis supporting the existence of the consistency relations, it is possible to establish the operating modes signatures, or in other words to determine the subset of consistency relations that are satisfied or violated in such and such operating mode. Consistency relations hence characterize the observable subspaces corresponding to the projection of the different operating modes of the system in the output space. They play a crucial part in the diagnosability properties of a system and possible discrimination of its different operating modes (Travé-Massuyès *et al.*2004) (Travé-Massuyès *et al.* 2003). The discriminating qualitative relationships exhibited by the qualitative models obtained by our method are hence closely related to what could be called *qualitative ARRs* and this needs further analysis.

References

Aguilar, J.; López de Mántaras R. 1982. The process of classification and learning the meaning of linguistic descriptors of concepts. *Approximate Reasoning in Decision Analysis,*165-175, North Holland Publishing Company, Amsterdam (The Netherlands).

Bouamana, B.O.; 2001. Model builder for thermofluid systems using a bond graph and functional modelling. In *Proceedings of 13th European Simulation Symposium, ESSO1,* 822-826, Marseille (France).

Brignolo, R.; Cascio, F.; Console, L.; Dubois, P.; Dressler, O.; Millet, D.; Rehfus, B.; Struss, P. 2001. Integration of Design and Diagnosis into a Common Process, *In Electronic Systems for Vehicles*, 53-73, VDI Verlag, Duesseldorf (Germany).

Ching, J. Y.; Wong, A. K. C.; Chan, K. C. C. 1995. Class-Dependant Discretization for Inductive Learning from Continuous and Mixed-Mode Data, *IEEE Transactions on Pattern Analysis and Machine Intelligence,*7 (17), 641-651.

Console, L.; Correndo, G.; Picardi, C. 2003. Deriving Qualitative Deviations from MATLAB Models, *In Proceedings of the Fourteenth International Workshop on Principles of Diagnosis*, DX'03, 87 – 92, Washington DC (USA).

Cordier, M.O.; Dague, P.; Levy, F.; Montmain, J. Staroswiecki, M.; Travé-Massuyès, L. 2004. Conflicts versus analytical redundancy relations: a comparative analysis of the model-based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions. on Systems, Man, and Cybernetics. Part B: Cybernetics*, 34(5), 2163-2177.

Kempowsky T.; Aguilar-Martin J.; Subias, A.; Le Lann M.V. 2003. Classification tool based on interactivity between expertise and self learning techniques. *In Proceedings of the 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'2003)*, Washington DC (USA).

Kurgan, L. A. ; Cios, K. J. 2004. CAIM Discretization Algorithm, *IEEE Transactions on Knowledge and Data Engineering,*16(2), 145-153.

Struss, P. 2002. Automated Abstraction of Numerical Simulation Models. *In Proceedings of the 16th International Workshop on Qualitative Reasoning QR'02*, May 2-4, Semmering (Austria), 161-168.

Travé-Massuyès, L.; Escobet, T.; Olive, X. 2004. Diagnosability Analysis based on Component Supported Analytical Redundancy Relations, *Internal Report LAAS N°04080* (to appear in *IEEE Transactions. on Systems, Man, and Cybernetics Part* A).

Travé-Massuyès, L.; Escobet, T.; Spanache, S. 2003. Diagnosability analysis based on component supported analytical redundancy relations. *In Proceedings of the 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS'2003)*, 897-902, Washington DC (USA).

Yan, Y. 2003. Automatic Qualitative Model Abstraction from Numerica Simulation Model. *In Proceedings of the Fourteenth International Workshop on Principles of Diagnosis DX'03*, 199-206, Washington DC (USA).

Yan, Y.; Lemire, D.; Brooks, M. 2004. Monotone Pieces Analysis for Qualitative Modeling, *In Proceedings of ECAI 04 MONET Workshop on Model Based Systems and Qualitative reasoning*, August 23-27, Valencia (Spain).