International Conference on Control, Engineering & Information Technology (CEIT'13)
Proceedings Engineering & Technology - Vol.3, pp. 100-105, 2013

Multi-model approach for systems diagnosis based on SVDD classification

Foued Theljani, Kaouther Laabidi, Salah Zidi, Moufida Ksouri
University of Tunis El Manar, National Engineering School of Tunis
Analysis, Conception and Control of Systems Laboratory (LR-11-ES20)
BP, 37, Le BELVEDERE, 1002 Tunis,Tunisia
Email : {foued_theljani, labidi_kaouther, salah_zidi, moufida_ksouri}@yahoo.fr

***Abstract*—This work focuses on the problem of systems diagnosis using classification-based method. We employ for this purpose the well known Support Vector Domain Description, abbreviated as SVDD, to make our decision about the functioning state. Because of their complexity, industrial systems are able to operate in various modes. Some of these modes correspond to normal functioning and some others represent faulty functioning. To properly discriminate between modes, we deal with such systems according a multi-model description based on multi-class version of the SVDD technique. To well assess the proposed approach, simulations and experimental tests are done on real hydraulic system consisting of three interconnected tanks.**

***Index Terms*—Classification, Diagnosis, Multi-model, SVDD.**

I. Introduction

Engineering systems, because of their increasing complexity, are able to operate in various states. Some of these states correspond to normal functioning and some others represent faulty functioning. Therefore, the diagnosis is a major task in industrial applications. It consists in tracking the operating state of the system in the course of time and detecting any abnormal event as it occurs. In this paradigm, many methods can be useful and applicable. We categorize them into analytical methods based on mathematical models, relational methods based on expert's knowledge and methods founded on pattern recognition and data classification. In pattern recognition, the functioning modes are represented by a set of similar patterns, called classes. These patterns are obtained by the observation of the most informative parameters of the system. Among the classification algorithms, we cite the Support Vector Domain Description known as SVDD.

The SVDD technique is developed at first by [1], [2]. It is an efficient technique employed to solve one-class classification problems (named also novelty detection problems). The fundamental goal of one-class learning is to generate a rule that distinguishes between a set of target objects called the target class and unseen-novel objects designated as outlier class. In some diagnosis cases, classes are able to evolve over the time and progress in their projection space. For that reason, any diagnosis approach must be endowed with useful tools which allow following online these evolutions and detecting changes. To do so, we use a dynamic version of the SVDD technique proposed in previous works [3], [4]. Furthermore, to fall into the optimum, the SVDD training process assumes dense sampling and requires that all training examples be available at once for a single "batch" learning: if a new sample is presented, the classifier must be retrained from scratch. The concern arises when the cardinality of data increases insofar as the process becomes embarrassing in terms of memory and training time. To overcome this dilemma, we adopt some approximations on the cardinality of the training dataset in order to ensure the rapidity of convergence and reduce the training time [3], [5]–[7], [9], [12]. In the other hand, because of their increasing complexity, industrial systems are often described according a multi-model architecture for purposes of simplification. In this case, the diagnosis task can be perceived as a multi-class classification problem since there are many functioning models, where each model designates one class of data. In view of that, the SVDD technique is as well adapted for multi-class problems in order to support the multi-model architecture of complex systems.

The rest of paper is organized into four sections. In Section 2, we introduce a global overview of the SVDD technique and its theoretical foundation. As for Section 3, it is reserved to describe the dynamic version of SVDD proposed in previous works. The multi-model approach for system diagnosis based on SVDD is well explained in Section 4. We show also here some experimental tests performed on real hydraulic system. In the conclusion, we summarize the presented work and we discuss the interest that yields.

II. SVDD : Theoretical overview

Let $\chi = \{x_1, ..., x_i, ..., x_N\}$ be a target training set, with $\chi \subseteq R^d$. The SVDD technique is a well known data description based on Kernel machines. It aims to find the smallest hyper-sphere containing the most of training instances of the target class with some relaxation defined by slack variables. The original problem is formulated as a constrained optimization as follows:

$$\min\ R^2 + C\sum_{i=1}^{N}\xi_i \tag{1}$$

$$\begin{aligned} s.t \quad & (x_i - a)^T (x_i - a) \le R^2 + \xi_i\,, \\ & \xi_i \ge 0 \ \ \forall i = 1, ..., N, \end{aligned} \tag{2}$$

where R and a denote respectively the radius and the center of the hyper-sphere and C is the regularization parameter which gives the trade-off between the volume of the sphere and the misclassified instances. The variable ξ_i is a slack variable designating the distance of i^{th} data point from the sphere boundary. It represents a quadratic optimization problem that can be solved efficiently by introducing Lagrange multipliers for constraints. The Lagrangian formulation of the problem is given thus by the following formula:

$$L = R^2 + C\sum_i \xi_i - \sum_i \alpha_i \left(R^2 + \xi_i - (x_i - a)^T (x_i - a) \right) - \sum_i \gamma_i \xi_i \tag{3}$$

where α_i and γ_i are the Lagrange multipliers, so that $\alpha_i \geq 0$, $\gamma_i \geq 0$. Note that for each training data point x_i, a corresponding α_i and γ_i are defined. The expression of L has to be minimized with respect to R, a, and ξ_i and maximized with respect to α_i and γ_i. Taking the derivatives of (3) by setting $\partial L/\partial R = 0$, $\partial L/\partial a = 0$ and $\partial L/\partial \xi_i = 0$, we obtain the Karush-Kuhn-Tucker (KKT) conditions [8] given by the following relations:

$$\begin{aligned} \frac{\partial L}{\partial R} &= 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i = 1, \\ \frac{\partial L}{\partial a} &= 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i x_i = a, \\ \frac{\partial L}{\partial \xi_i} &= 0 \quad \Rightarrow \quad 0 < \alpha_i < C. \end{aligned} \tag{4}$$

The QP equations are obtained by substituting the above KKT conditions in (3). We obtain a dual problem expressed by:

$$\max \ \tfrac{1}{2}\sum_{i=1}^{N} \alpha_i (x_i{}^T x_j) - \ \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j (x_i{}^T x_j), \tag{5}$$

$$s.t \ \sum_{i=1}^{N} \alpha_i = 1 \ , \quad 0 \leq \alpha_i \leq C \ \ \forall i = 1, ..., N. \tag{6}$$

After solving such a standard Quadratic Programming (QP), we obtain the solution $\alpha_i = \alpha^*$, whose corresponding training instances can be classified as Boundary Support Vectors $(BSVs)$ outside the hyper-sphere, if $\alpha_i = C$, Non-Support Vectors $(NSVs)$ inside the hyper-sphere, if $\alpha_i = 0$ and Non-Boundary Support Vectors $(NBSVs)$ just at the hyper-sphere, if $0 < \alpha_i < C$. Any data satisfying one of the above KKT conditions is designated as target class. Otherwise, it is an outlier. Furthermore, all points with $\alpha_i > 0$ are called Support Vectors (SVs) which restrict the data domain and can fully describe the one-class boundary. We can write thus $SVs = BSVs \cup NBSVs$. According to (4), the center a can be easily calculated as:

$$a = \sum_{i=1}^{N} \alpha_i x_i. \tag{7}$$

To make prediction on the membership of an unknown instance z, the squared distance to the center of the sphere must be calculated using the following formula:

$$F(z,a) = (z^T z) - 2\sum_{i=1}^{N} \alpha_i (z^T x_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j (x_i{}^T x_j) \tag{8}$$

Afterward, we define the decision function as:

$$G(z) = R^2 - F(z,a) \tag{9}$$

Now we say that the test instance z belongs to the target class or lies inside the hyper-sphere if $G(z) \geq 0$. Otherwise, it is an outlier lying outside the hyper-sphere. Similarly, in traditional support vector machines and other kernel machines, the inner product between two vectors in (7) and (8) can be replaced by various kernels satisfying the Mercer theorem [10]. By introducing a kernel function, the formula (8) becomes as follow :

$$F(z,a) = K(z^T z) - 2\sum_{i=1}^{N} \alpha_i K(z^T x_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(x_i{}^T x_j) \tag{10}$$

As we see, the methodology above gives efficiently an optimized domain description particularly when there is no confrontation with evolutionary data. Nevertheless, the challenge becomes valuable when dealing with non stationary data likely to progress in their space. In the next section, we address this paradigm.

III. Previous work : fast dynamic SVDD

A. Initial formulation

The main challenge is how to deal with the newly added samples and on which criteria we rely to discard irrelevant data. At each iteration, the optimization process is performed on training set denoted $S_{training}$. As initialization, let $S_{training} = SVs$. As previously mentioned, two main aspects will be addressed; the incremental classification and the dynamic adaptation. The adopted approach is structured around basic proprieties that we analyze in the sequel.

Propriety 1: In the classification learning scheme at each step we add one instance to the training set. If the newly added sample meets the KKT conditions, it has not any effect on the previous data description and can be discarded from the training set. Therefore, the optimization process is useless in this case since the new added sample doesn't minimize the currently minimum objective function and the same result will be produced. This property leads us to reduce considerably the cardinality of training samples and ensure the rapidity of convergence.

Propriety 2: On the other hand, if samples in the newly added training set lie outside the hyper-sphere, some among them will become new support vector surely. This propriety indicates that in the newly added training set the samples violating the above KKT conditions should be taken and integrated within the training working set during the optimization process.

Accordingly, as a first step towards an incremental learning, we should seek the neighborhood of each instance presented online with respect to the current data domain. To do so, for a random sample $z^{(k)}$ coming at iteration k, the distance to the center of the hyper-sphere can be calculated through (8). Thus, two cases are possible here:

Case 1: $G(z_t) \geq 0$
This case means that the query sample z_t lies inside the hyper-sphere (target). According to Propriety 1, this sample has no effect on the current SVs set and don't minimize the objective function. Hence, we preserve the same whole of SVs and we don't carry out any change on it. In this way, we get rid for once of the optimization process which can be expensive in terms of time and memory while guaranteeing the optimal solution.

Case 2: $G(z_t) < 0$
Each sample satisfying this condition is considered as outlier and likely to be probably a new support vector according to Propriety 2. In this case, the SVs set needs to be updated to incorporate samples newly detected. To improve performances, we assign a dynamic aspect to the learning process. Each insertion of a new sample must be accompanied by a deletion procedure applied on irrelevant data. We aim thus a removal/insertion procedure.

B. Dynamic data removal /insertion

1) Removal procedure: The removal procedure, as well the insertion procedure, is performed only when a random sample z_t received at iteration t is an outlier meeting the condition mentioned in case.2. Otherwise, z_t is assigned to $NSVs$ set. The deletion concerns particularly the irrelevant samples that don't follow the evolution of data domain. F. Camci proposed in [11] a weighted support vector novelty detector $(WSVND)$ for non-stationary data. To explicitly support the non-stationary nature of data, the method incorporates the notion of weight or importance of data point based on its age. It forces the support vectors to be as young as possible so as to be able track the non-stationary process. The solution seems biased and insufficient given that we can fall into the case where old support vectors can be more significant than younger some others. In this way, the method is not immune to local optima drawbacks if support vectors are judged upon their oldness. In view of that, we propose in this work a simple formulation to deal with this phenomenon. Instead of judging data with respect to their oldness, we perform a weighted relevance measure based on the neighborhood concept. Indeed, samples lose their significance while moving away from the high-density zone of the data domain. Based on this assumption, a spatial trend prediction needs thus to be realized. As a notation, for each coming z_t, let a_t and a_{t+1} be respectively the current center of the hyper-sphere and the probable future center when z_t becomes a new support vector. Let also d_t and d_{t+1} be the distance between a sample $x_i \in SVs$ and respectively a_t and a_{t+1}. We attempt to extract the element among the whole SVs to be deleted. This removable support vector SV_{del} should satisfy the following expression:

$$\begin{aligned} & SV_{del} = \arg\max \left(sign(\Delta d) \times \|z_t - x_i\| \right), \\ & x_i \in SVs, i = 1, ..., N_{SVs} \end{aligned} \tag{11}$$

The term Δd is the distance variation expressed as:

$$\Delta d \ = \ d_{t+1} - d_t, \tag{12}$$

where

$$\begin{cases} d_t = \sqrt{\left(x_i - a_t\right)^T \left(x_i - a_t\right)} \\ d_{t+1} = \sqrt{\left(x_i - a_{t+1}\right)^T \left(x_i - a_{t+1}\right)} \end{cases} \tag{13}$$

The predicted center a_{t+1} can be simply determined by a recursive relation, so that:

$$a_{t+1} = a_t + \frac{1}{N_{SVs} + 1} \left(z_t - a_t\right) \tag{14}$$

According to expression (11), the rule chooses for deletion the foremost support vector which moves away from center when the last one shifts relative to each sample newly appended. This seems proper as much as the removable support vector becomes increasingly isolated and far-off from the high-density zone of the data domain. Once SV_{del} is determined, the training set $S_{training}$ is afterwards adjusted by deleting SV_{del} and inserting the new coming instance z_t, so that:

$$S_{training} = \{S_{training}\} \setminus \{SV_{del}\} \cup \{z_t\} \tag{15}$$

Evidently, the instance z_t must satisfy the case.2.

2) Insertion procedure: At this stage, to enrich the training set and guarantee an optimal description of the data domain, we aim to find data points among the $NSVs$ set which can be probably new support vectors. Without doubt, these points are those in close proximity to the interior boundary of the hyper-sphere. For each point $x_j \in NSVs$ $(j = 1, ..., N_{NSVs})$,N_{NSVs} is the total number of samples belonging to $NSVs$ set, samples which are most likely to be support vectors can be formulated as:

$$\sqrt{F\left(x_j, a_t\right)} \geq T, \quad j = 1, ..., N_{NSV}. \tag{16}$$

The term T is a decision threshold which is fixed with respect to the data distribution estimated on the target class whole. We denote by $PSVs$ the set grouping the Probable Support Vectors meeting conditions given by (16). Hence, the training set $S_{training}$ is adjusted again to insert the $PSVs$ components, so that:

$$S_{training} = \{S_{training}\} \cup \{PSVs\} \quad (17)$$

As a result, we obtain at the end a compact training set that reduces significantly the complexity of the QP problem in terms of cardinality of data and convergence time. This makes the method useful for applications requiring fast data tracking, even online processing. Moreover, despite their small cardinality, the training set elements are significant and well chosen through selective rules to ensure the optimality in the solution. More experimental tests of proposed improvements can be found in [3].

IV. Multi-model approach for system diagnosis

A. System description

For validation, we consider a hydraulic system illustrated by figure 1 which is available in the Unit of research Analysis, Conception and Control of Systems at National Engineering School of Tunis ACCS-ENIT. It consists of three interconnected tanks $T1$, $T2$ and $T3$. The tanks at the extremity $T1$ and $T3$ have similar sections $S = S1 = S3$. They are connected to $T1$ having a section $S2$ through two solenoid-operated valves V_{12} and V_{23}. All tanks are equipped with three drain valves respectively denoted V_1, V_2 and V_3. The filling of tanks is assured by two pumps supplying the same flow ($Q1 = Q2$). Three ultrasonic sensors are placed on top of each tank to measure water levels. We denote as :

- $Q1$, the filling rate of the pump $P1$,
- $Q2$, the filling rate of the pump $P2$,
- H_1, H_2 and H_3, the water levels of each tank.

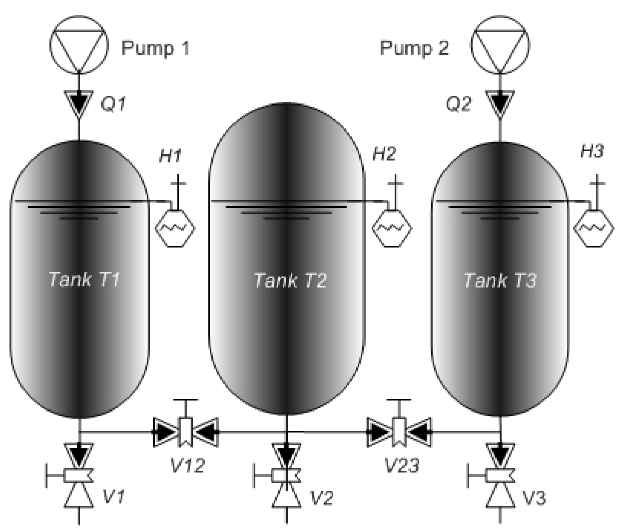

Figure 1. System consisting of three interconnected tanks.

B. Diagnosis methodology

The multi-model approach consists in representing a given system by a set of simple models. Each model represents a required behavior in a considered state. The transition from one model to another is due to the occurrence of an event reflected in the evolution of the functioning zone.

Because of their complexity, industrial systems can operate in several modes, where each one fulfills a required mission. Each mode may correspond to several models. For example, For a given operating mode M, it is possible to consider a first correct model ω_1 for which the system fulfills perfectly its normal behavior, and a second faulty model ω_2 for which the system operates with the presence of a fault.

In this part, we adopt a multi-model approach for the diagnosis of the hydraulic system previously described in the figure 1. Two modes are considered: the first one is denoted M_1 for which the valve V_{12} is closed and the second one is denoted M_2 for which the valve V_{23} is closed. Two operating models are considered for each of these modes: a normal functioning model and a faulty functioning model. This last one corresponds to a progressive failure affecting the pumps. All modes and their corresponding models are described in the table below.

Table I
Functioning modes and their corresponding models.

Mode	Model	Type
Mode M_1 (V_{12} closed)	Model ω_1 (pumps in correct operating)	Normal functioning
	Model ω_2 (pumps in progressive failure)	Faulty functioning
Mode M_2 (V_{23} closed)	Model ω_3 (pumps in correct operating)	Normal functioning
	Model ω_4 (pumps in progressive failure)	Faulty functioning

We assume that the transition from one mode to another is done by switching, while the transition from one model to another is performed with a slow progressive transition, as described in figure 2.

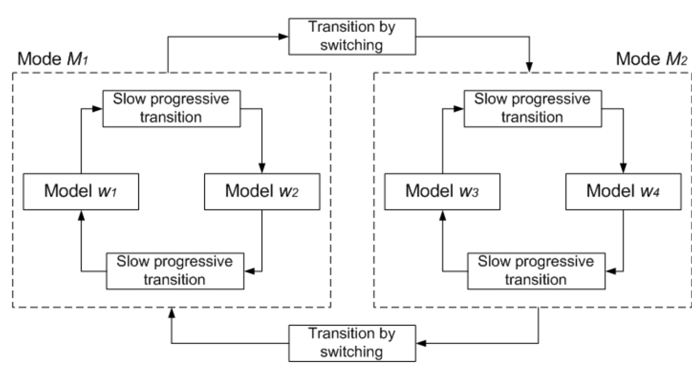

Figure 2. Modes, models and their transitions.

Commonly, the models ω_k ($k = 1, ..., 4$) are obtained by a mathematical modeling of the system behavior in several operating states. In the paradigm of data classification, these models consist of disjoint functioning zones illustrated in the projection space by shapes named "classes". Each class refers to one functioning zone (or functioning condition). In the case of SVDD classifier, these classes are restricted in the set of boundary points that delimit them. These points, called Support Vectors SVs are obtained according to the statistical learning of SVDD formalized in the previous section.

The online classification of observations is made by a recognition procedure which measures the membership degree of a new instance z_t to each class ω_k. This measure is estimated using the membership function $F(\bullet,\bullet)$ given by (8). We get at the end a bench of four membership functions described as follows:

$$F_k(\omega_k, z_t) = K(z_t, z_t) - 2\sum_{i=1}^{N} \alpha_i K(z_z, x_{k,i}) + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(x_{k,i}, x_{k,j}), \quad k = 1, ..., 4 \tag{18}$$

$\{x_{k,i}\}_{i=1}^{N}$ is the training dataset associated respectively to the classes ω_k $(k = 1, ..., 4)$. The term K is the used Kernel function. We adopt for experiments the Gaussian distribution to describe densities. We express the Gaussian Kernel function as:

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma}\right) \quad , \forall\, i \neq j \tag{19}$$

where σ is the scale parameter. The instance z_t is an input vector consisting of three attributes representing the water level at the current iteration in each tank. Hence, z_t is defined as:

$$z_t = \begin{pmatrix} H_1 \\ H_2 \\ H_3 \end{pmatrix} \tag{20}$$

The above functions (18) provide as output some measure of distance which gives the membership degree of a new instance z_t to one of the existing classes. The model in which the system operates corresponds to the maximum output of the associated function membership, as shown in figure (3).

C. Simulations

Referring to the methodology described above, the goal is to follow online the process, detect any change in its functioning as soon as it occurs and identify the model in which the process operates. To do so, we simulate the process according to the following sequence: ω_1, ω_2, ω_3 and ω_4. The transition from ω_1 to ω_2 and from ω_3 to ω_4 is a slow progressive transition. For cons, the transition from model ω_2 to model ω_3 is performed by switching.

For each coming instance z_t, we evaluate the output provided by the functions $F_k(\omega_k, z_t)$ defined above. To each function $F_k(\omega_k, z_t)$, we attribute some binary validity v_k. This last one takes the score "1" if the output of $F_k(\omega_k, z_t)$ corresponds to the maximum output among all, otherwise v_k takes "0". v_k can be expressed thus as:

$$\begin{cases} if \quad F_k(w_k, z_t) = \max\{F_i(w_i, z_t)\}_{i=1}^{4} \quad , \quad v_k = 1 \\ else \quad v_k = 0 \end{cases} \tag{21}$$

where $k = 1, ..., 4$.

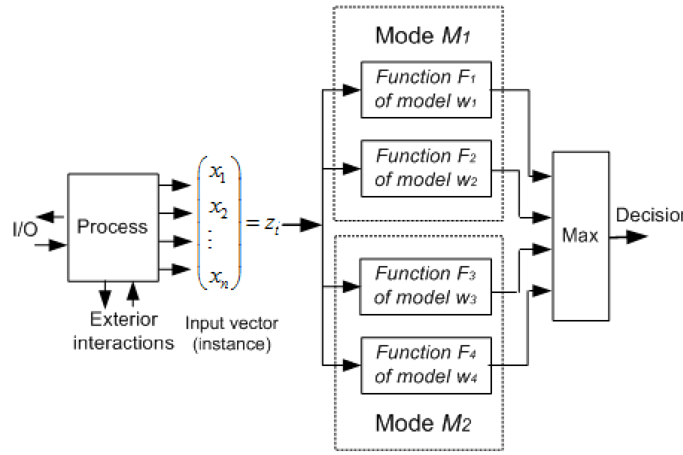

Figure 3. Modes, models and their transitions.

Figure (5) shows the function outputs $F_k(\omega_k, z_t)$ and their validities. Referring to this figure, we can extract four functioning zones:

- Zone 1, in which the system operates according to model ω_1, given that membership function of ω_1 corresponds to the maximum value in this range of time.
- Zone 2, in which the system operates according to model ω_2. Similarly, this zone corresponds to the maximum output of the membership function of model ω_2.
- Zone 3, in which the system operates with respect to the third model ω_3.
- Zone 4, in which the system operates with respect to the fourth model ω_4.

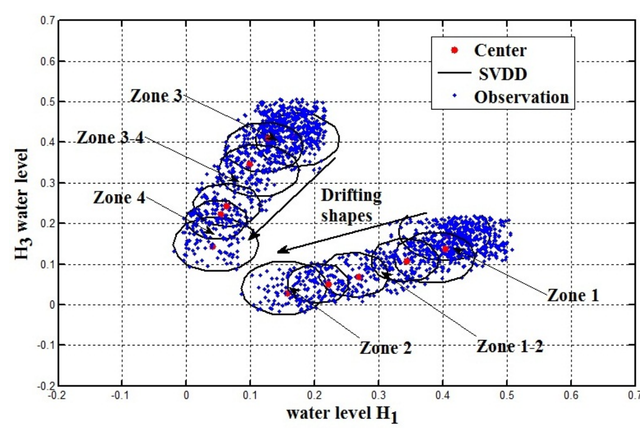

Figure 4. The SVDD estimation in the space of H_1 and H_3 variables.

As well, there two transition zones which are Zone 1-2 and Zone 3-4. These zones are due to the fact the system

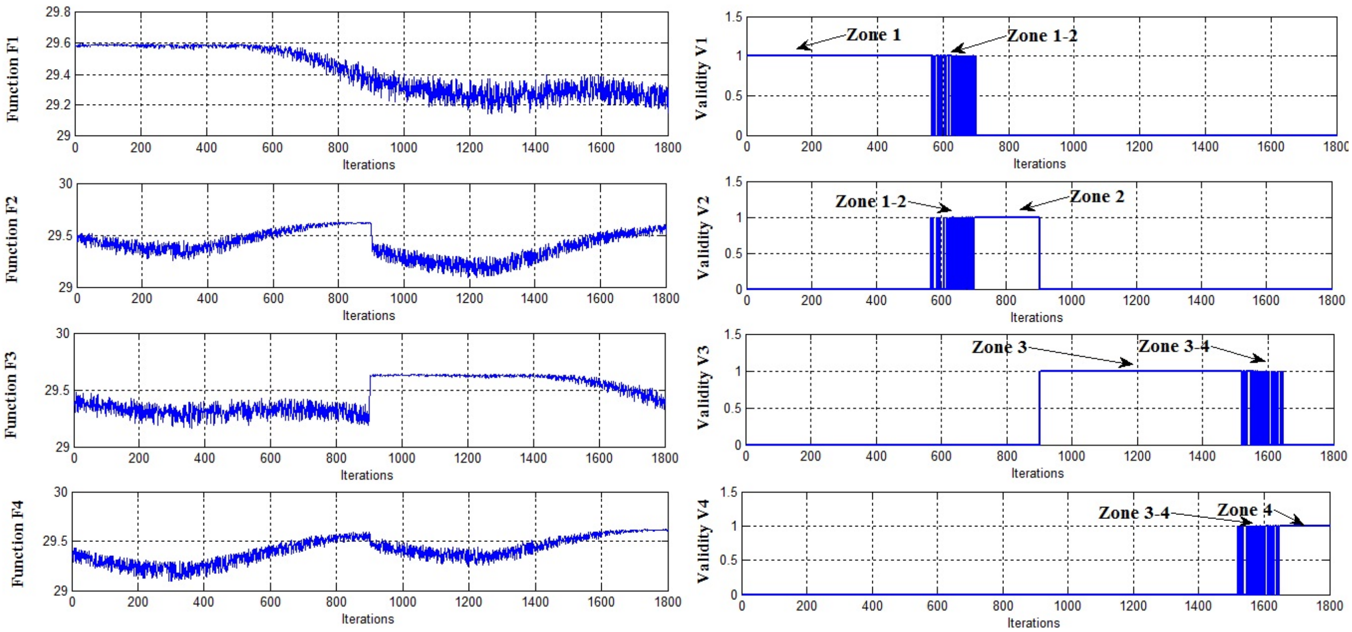

Figure 5. Outputs of membership functions and validities.

shifts from model ω_1 to model ω_2 and from model ω_3 to model ω_4 slowly and progressively. This gives arise to drifting shapes that evolves in the projection space and acts as a sliding data window. As shown in figure (4), despite these dynamic phenomena, the SVDD is dynamically updated and maintained in order to track these evolutions.

V. Conclusion

In this paper, we proposed a robust methodology for diagnosis and system safety based on classification with an improvement version of SVDD technique. This last one is adapted to deal with three main issues. The first one is related to the QP complexity which grows with respect to the number of training samples. Based on KKT conditions, some approximations on the size of the training dataset are adopted to optimize the convergence time. The second problem concerns the evolving data and drifting objects. To cope with these phenomena, the proposed algorithm is endowed with useful tools which maintain dynamically the described data domain by inserting new samples and removing the most irrelevant ones according to adequate rules. Based on such SVDD classifier, the goal is to ensure the online diagnosis of system and distinguish between different functioning models. Hence, we adopted a multi-model approach to describe the diagnosed system, since it can operate in various models. We considered also the situation in which the transition between models is progressive and slow to prove the effectiveness of the SVDD algorithm against degraded data. The approach was assessed afterwards on a hydraulic system consisting of three interconnected tanks.

References

[1] D. Tax and R. Duin, *Support vector domain description*, Pattern Recognition Letters, Vol. 20, pp. 1191-1199, November 1999.

[2] D. Tax and R. Duin, *Support vector data description*, Machine Learning, Vol. 54, pp. 45-66, January 2004.

[3] F. Theljani, K. Laabidi, S. Zidi and M. Ksouri, *Support Vector Domain Description for non-stationary data*, The 6th International Conference on Application of Information and Communication Technologies (AICT'12), Georgia, Tbilisi, October 2012.

[4] F. Theljani, K. Laabidi, S. Zidi and M. Ksouri, *Systems monitoring based on dynamic classification with SVDD*, International Multi-Conference on Systems Signals & Devices (SSD'13), Tunisia, Hammamet, Mars 2013.

[5] X. Hua and S. Ding, *Incremental Learning Algorithm for Support Vector Data Description*, Journal of Software, Vol. 6, No. 7, pp. 1166-1173, July 2011.

[6] R. R. Sillito and R. B. Fisher, *Incremental one-class learning with bounded computational complexity*, Proceedings of the 17th international conference on artificial neural networks (ICANN'07), Berlin 2007.

[7] A. Tavakkoli, M. Nicolescu, M. Nicolescu, G. Bebis, *Incremental SVDD Training: Improving Efficiency of Background Modeling in Videos*, Proceedings of the International Conference on Signal and Image Processing, Kailua-Kona, Hawaii, August 2008.

[8] E. Osuna, R. Freund, F. Girosi, *Improved Train-ing Algorithm for Support Vector Machines*, Proceeding of Neural Networks in Signal Processing (IEEE NNSP'97), Amelia Island, September 1997.

[9] X. Jianhua, *Constructing a Fast Algorithm for Multi-label Classification with Support Vector Data Description*, Proceedings of the 2010 IEEE International Conference on Granular Computing (GRC '10), IEEE Computer Society Washington, DC, USA 2010.

[10] V. N. Vapnik, *Statistical Learning Theory*, New York: John Wiley & Sons, 1998.

[11] F. Camci and R. B. Chinnam, *General support vector representation machine for one-class classification of non-stationary classes*, Pattern Recognition, Vol. 41, Issue 10, pp. 3021-3034, October 2008.

[12] F. Bellocchio, S. Ferrari, V. Piuri, N.A. Borghese, *Hierarchical approach for multiscale Support Vector Regression*, IEEE Trans. on Neural Networks and Learning Systems, Vol. 23, No. 9, pp. 1448-1460, September 2012.