Research Article

Hierarchical Fault Diagnosis for a Hybrid System Based on a Multidomain Model

Jiming Ma¹ and Jianbin Guo²

¹Sino-French Engineer School, Beihang University, No. 37 XueYuan Road, Beijing 1001091, China

²School of Reliability and System Engineering, Beihang University, No. 37 XueYuan Road, Beijing 1001091, China

Correspondence should be addressed to Jianbin Guo; guojianbin@buaa.edu.cn

Received 20 May 2014; Revised 11 August 2014; Accepted 31 August 2014

Academic Editor: Hamid R. Karimi

Copyright © 2015 J. Ma and J. Guo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The diagnosis procedure is performed by integrating three steps: multidomain modeling, event identification, and failure event classification. Multidomain model can describe the normal and fault behaviors of hybrid systems efficiently and can meet the diagnosis requirements of hybrid systems. Then the multidomain model is used to simulate and obtain responses under different failure events; the responses are further utilized as a priori information when training the event identification library. Finally, a brushless DC motor is selected as the study case. The experimental result indicates that the proposed method could identify the known and unknown failure events of the studied system. In particular, for a system with less response information under a failure event, the accuracy of diagnosis seems to be higher. The presented method integrates the advantages of current quantitative and qualitative diagnostic procedures and can distinguish between failures caused by parametric and abrupt structure faults. Another advantage of our method is that it can remember unknown failure types and automatically extend the adaptive resonance theory neural network library, which is extremely useful for complex hybrid systems.

1. Introduction

Hybrid systems include both continuous variable dynamic systems (CVDSs) and discrete event dynamic systems (DEDS) [1] and consist of components with continuous and discrete behavior. Because industrial systems have become more complex, most have the continuous and discrete features of hybrid systems and should be described simultaneously by differential and difference equations. There has been much research into the modeling, analysis, and design of hybrid systems, and there have been many achievements in this field [2].

Generally, procedures for diagnosing hybrid systems are designed using extended observers that estimate the system state under normal and fault conditions [3–5]. Researchers describe systems using hybrid automata [6–8], Petri nets [9–12], or switched systems [13, 14] and make a diagnosis according to values measured from a real system. Alternatively, they design an observer [15–19] to describe a system [19],

estimate a system's status (failure or not) [18], or improve the stability [14]. Fault detection is useful and necessary for normal linear and nonlinear system when designing system controller with high robust ability [20, 21]. However, it is difficult to develop a procedure for diagnosing hybrid systems with complex continuous and discrete dynamics, because quantizing the complex behaviors using a finite set of states and events results in large nondeterministic models, degrades the performance, and increases the computational requirements of the diagnostic algorithms.

Daigle et al. [22] presented a novel approach to construct diagnostic procedures for isolating single, abrupt faults in hybrid systems. Their technique was based on a qualitative abstraction of measured deviations from normal behavior, whereas the procedure is not applicable to the hybrid system with several different CVDS. A fault classification framework for a hybrid system has been created by Tromp et al. [9], the hybrid system was modeled based on a Petri net, and the system normal behavior and fault behavior were described as

certain discrete event and random discrete event separately. The author further performed the fault diagnosis by obtaining a system event transition sequence based on the measured values from the real system. Philippot et al. [15] developed a discrete event model-based approach for fault detection and isolation of manufacturing systems. Alexandre considers a system as a set of independent plant elements (PEs), each being composed of a set of interrelated parts of plant (PoPs) modeled by a Moore automaton. The PoP model was affected only by its local behavior. Degraded and faulty behaviors were added to each PoP model in order to obtain extended behavior.

Zhao et al. [14] established a diagnostic system architecture that integrated timed Petri net modeling and diagnostic components. The author stated that the advantages of model-driven signature analysis are that it can significantly reduce the computational cost of mode estimation and enabled the diagnostic system to operate online. Bhowal et al. [23] proposed an automaton modeling method for discrete time hybrid system. They utilized a variable-based process model where each activity state represented a particular dynamic of the system. A diagnostic construction method was then formulated based on both the measurable and unmeasurable partitions of the variables and relationships with event transitions.

When designing a model-based diagnostic procedure for a hybrid system, it is important to integrate failure behavior modeling, the acquisition and processing of sensor signals, and fault mode. Researchers need to address diagnostic problems at the interface between model-based diagnostic techniques and signature analysis, so that incipient and abrupt faults can be efficiently detected and isolated. There are, however, few specific reports in the literature distinguishing between the different CVDSs of hybrid systems. Most of the current diagnostic algorithms consider variables which have little impact on the hybrid system; so the algorithms are too complex and that their applications are limited [24, 25].

Diagnosis-oriented modeling is essential for a hybrid system, because it is essential for describing all the important factors, such as continuous variable dynamic processes, discrete event dynamic processes, fault behaviors, and uncertain parameters. The rules of hybrid automata modeling are simple but result in an oversized model when describing a complex system. Petri net model design requires an in-depth understanding of the fault mechanisms of components and their transmission characteristics and requires artificially set failure probabilities. Particle filtering will consume considerable computing resources when making diagnostic procedures for a complex system. As with Petri nets, the transition probabilities in Markov chain models should be set manually.

Modelica is one of the most advanced languages for hybrid system modeling. The equation-based object-oriented methodology supported by Modelica provides several advantages when compared with the block diagram modeling approach. Noncausal description is one advantage of models that use differential, algebraic, and discrete equations. Modelica is an object-oriented multidomain modeling language, which can effectively describe hybrid systems [26–30], and

CVDS and DEDS simultaneously. Sanz et al. [26, 29] used a free Modelica library (DEVSLib) that supports the parallel DEVS formalism and is mainly designed to model discrete event systems. They developed an object-oriented modeling methodology, supported by Modelica, for multidomain and multiformalism hybrid control systems. Modelica has also been used successfully in multifield modeling for dynamic systems [31, 32].

The adaptive resonance theory neural network 2 (ART2) [33, 34] has various network structures and strong self-learning ability. ART2 can memorize unknown modes through nonsupervised learning ability. In addition, ART2 can be used to create a self-learning function for a diagnostic system when applied to a poor fault a priori information system.

The Brushless DC motor (BLDCM) has the advantages of high efficiency, high starting torque, and high power density and has been widely used in industrial systems with high performance requirements. The BLDCM switches its three-phase armatures on-off based on position feedback signals. The switching changes the topological structure of the power unit of the BLDCM. Besides this, some continuous variables, such as the current of the armature, torque, and rotation speed, vary in real time. Therefore, it is clear that the BLDCM has both continuous variables (such as phase current, output torque, and output rotation speed) and discrete variables (such as the on-off state of the power unit), and it is a typical hybrid system. Moseler and Isermann [35] proposed a model-based method for the detection of faults in BLDCM. The model was derived from the bridge supply voltage/current and the rotor velocity, and the operating procedure was simplified as a single continuous process, whereas the diagnostic procedure needs to be executed with the parameter estimation results and did not take the six operation stages difference into account. Yu et al. [36] presented a diagnostic approach for a BLDCM based on wavelet transforms and artificial neural networks. The difference with Moseler's method is that the model considers the effects of six different operating sequences through three fault models, the stator winding interturn short circuit fault model, open-switch fault model, and open-winding fault model.

The aim of the present paper is to provide a diagnostic method for hybrid systems. First, multidomain modeling methods are studied to meet diagnostic requirements of hybrid system. Furthermore, the simulation is executed to obtain the response under failure, and the response would be a priori information for training the ART2 library. A hierarchical fault diagnostic engine is then developed based on the characteristics of the hybrid system. Using the event identification results and estimated parameters, the diagnostic engine is designed to achieve the event transition sequence of the dynamic processes of the system. Both the event transition sequence and system estimated parameters are foundation of primary and detailed classification. Primary classification is performed to distinguish the failure classes of the hybrid system, and the detailed classification is used to achieve the exact fault location; an ART2 structure is designed throughout both primary and detailed classification.

FIGURE 1: Scheme of the fault diagnostic system.

2. Fault Diagnostic Methods for a Hybrid System

A hybrid system fault diagnostic scheme is shown in Figure 1. The scheme consists of three parts: multidomain modeling of the hybrid system, fault diagnostic system training (neural network training), and a fault diagnostic procedure.

Multidomain modeling is the first step and needs an in-depth knowledge of the topological structure of the hybrid system. A component library is necessary for integrating multidisciplinary components in a complex system. Fortunately, abundant freely available libraries for Modelica have been developed by early researchers, which make it easier to model hybrid systems. Modeling, simulation, and analysis are fundamental to the diagnosis. Simulation results are used to train the neural network on the diagnostic procedure.

The neural network is trained as follows. Firstly, the system topology and information regarding the system components are used to create a multidomain model of the hybrid. After establishing a clear understanding of all normal and failure behaviors of the system, simulations are used to determine the systems responses under normal and all known fault behaviors. The simulation results are a priori information for subsequent fault diagnosis. The event transition sequences of the system's dynamic processes are determined using event identification, and some of the parameters of the system's characteristics are obtained using parameter estimation. The ART2 structure for primary classification is trained according to the results of the transition sequences and parameter estimations. All the failure events are primarily classified using ART2, and every primary classification includes one

or more failure events. Finally, we design the ART2 for each class. During the detailed classification process, some further parameters (extracted using wavelet transforms) are entered into the ART2 structure. When integrated with the system failure classification information, these parameters are used to train the neural network to memorize the corresponding characteristics of each event of a failure class. The result of this training is an ART2 library for detailed classification.

The fault diagnostic processes are as follows. Firstly, based on the system running results, the event transition sequence is obtained using the event identification method, and several characteristic parameters are obtained through parameter estimation. The event transition sequence and characteristic parameters are entered into ART2 for primary classification, and the failure classification number is then known. Once we have extracted the ART2 for this class, the system event transition and further characteristic parameters are entered into the detailed classification network. Classification results should be identified using the current network library to distinguish whether the failure event has been previously memorized. A new failure event is then saved into the current ART2 structure for future detailed classification.

2.1. Multidomain Modeling. Hybrid systems include characteristics of CVDS and DEDS; so it is important that model-based fault diagnostic technique can exactly describe hybrid systems. A hybrid system model must be established and then applied into an actual diagnostic procedure. Modeling should consider a system's normal behavior, fault behavior, random fluctuation of parameters, and environmental perturbations.

FIGURE 2: Hybrid system model structure based on Modelica.

The behavioral modes of the components in a hybrid system can be expressed as follows.

Assuming the system has z components, the i th ($i = 1, 2, \dots, z$) component has several behavioral modes and the j th behavioral mode is described as the following dynamic system with m input and q output variables:

$$\begin{aligned}\dot{\mathbf{x}}_i(t) &= \mathbf{a}_i(t) \cdot \mathbf{x}_i(t) + \mathbf{b}_i(t) \cdot \mathbf{u}_i(t), \\ \mathbf{y}_i(t) &= \mathbf{c}_i(t) \cdot \mathbf{x}_i(t),\end{aligned}\quad (1)$$

where $\mathbf{x}_i(t) \{x_1(t), \dots, x_n(t)\} \in R^{n \times 1}$ is the state variable vector, $\mathbf{u}_i(t) \in R^{m \times 1}$ is the input variable vector, $\mathbf{y}_i(t) \in R^{q \times 1}$ is the output variable vector, and $\mathbf{a}_i(t) \in R^{n \times n}$, $\mathbf{b}_i(t) \in R^{n \times m}$, and $\mathbf{c}_i(t) \in R^{q \times n}$ are the structure parameters of the j th behavior of the i th component.

Behavior modeling based on Modelica is an object-oriented modeling method. First, the hybrid system is divided into several subsystems domain, and interfaces are created between different domains. The behavioral model of the component in each subsystem is then expressed based using (1). The behavior of each component includes normal and fault behaviors. The model structure is shown in Figure 2.

2.2. Theories and Methods for Neural Network Training. This section introduces some theories and methods related to fault diagnostic system training (neural network training), including the event identification method, parameter estimation method, and ART2 structure.

2.2.1. Event Identification. Assuming the hybrid system has k normal events, q_r ($0 \leq r \leq k$) represents the continuous variable dynamic process of the event r . The transition path is

$$\sigma = [(t_0, e_0), (t_1, e_1), \dots, (t_k, e_k)], \quad (2)$$

where $e_r = (q_{r-1}, q_r)$ represents the transition process from event q_{r-1} to event q_r ($r = 1, 2, \dots, k$) and $t_0 \leq t_1 \leq \dots \leq t_k$, $e_0 = q_0$ is the initial event.

According to (2), when $t_{r-1} < t < t_r$, the system will be within the q_r CVDS.

All the failure events can be separated into two categories. One is a transition failure: the ideal CVDS after transition time t_r should be the process q_{r+1} , and a transition failure has occurred when the real CVDS q_{fact} is not the same as process q_{r+1} . The other category is a nontransition failure: the system does not transit as it should do between t_r and t_{r+1} ($0 \leq r \leq k$), which causes the occurrence of some other process that the system should not perform.

The failure class is primarily based on the event transition sequence of the hybrid system. The real event transition sequence of the system is obtained using event identification; then, the real event transition sequence is compared to the ideal one. Failure occurs if they are not identical. Moreover, primary classification can distinguish different failure classes and positions using event transition sequences.

2.2.2. Parameter Estimation Method. The parameter estimation method can be used to evaluate system parameter values for the mathematical model using an optimization method and the input and output data of the system. The results are useful when diagnosing the system and are also necessary when training the neural network for primary classification.

The hybrid system of (1) with a single output $y(t) = x_1(t)$ can be described by a differential equation:

$$\begin{aligned}\frac{dy(t)}{dt} &= a_1 x_1(t) + \dots + a_n x_n(t) \\ &+ b_1 u_1(t) + \dots + b_m u_m(t),\end{aligned}\quad (3)$$

where $\mathbf{u}(t) \{u_0(t), u_1(t), \dots, u_m(t)\}$ is the input vector, $\mathbf{x}(t) \{x_1(t), \dots, x_n(t)\}$ is the state vector, and $\mathbf{a} \{a_1, \dots, a_n\}$ and $\mathbf{b} \{b_1, \dots, b_m\}$ are the system structure parameters (which can be evaluated using the least mean square (LMS) error algorithm).

The single output y in (3) can be solved analytically in a linear time invariant system:

$$y(t) = \boldsymbol{\psi}^T(t) \boldsymbol{\theta}, \quad (4)$$

where $\boldsymbol{\psi}(t) \in R^{n+m}$ is a time variant function relevant to the real-time input $\mathbf{u}$ and state vector $\mathbf{x}$ and $\boldsymbol{\theta}$ is the structure parameters vector

$$\boldsymbol{\theta}^T = [a_1, \dots, a_n, b_1, \dots, b_m]. \quad (5)$$

The equation error $\Delta(t)$ has been introduced for estimating parameters [35, 37]

$$\Delta(t) = y(t) - \psi^T(t) \hat{\theta}. \quad (6)$$

Assuming the sample time is T_s , M is the number of sampling data sets, and $z = t/T_s = 1, 2, \dots, M$, and the LMS algorithm uses $\Delta(z)$ to express the equation error in discrete time z

$$\Delta(z) = y(z) - \psi^T(z) \hat{\theta}(z). \quad (7)$$

Therefore, the structure parameters θ can be estimated using

$$\hat{\theta}(z+1) = \hat{\theta}(z) + 2\mu\Delta(z)\psi(z), \quad (8)$$

where μ is the step size.

A hybrid system has many failures with different characteristics; so failures cannot be diagnosed by only estimating the structural parameters of the system model using parameter estimation. More powerful signal processing technologies should be used to process the output signal and are necessary for training the neural network structure using characteristic parameters of the hybrid system for detailed classification.

The wavelet transform method is used to extract the hidden characteristic parameters of a hybrid system. For any function of time $f(t) \in R^2$, the continuous wavelet transform is defined as

$$(W_\psi f)(a, \tau) = |a|^{-1/2} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-\tau}{a} \right) dt, \quad (9)$$

where $a, \tau \in R$, $a > 0$, a is a scale parameter, τ is a displacement parameter, $(W_\psi f)(a, \tau)$ is the wavelet transform of the function $f(t)$, $\psi(\cdot)$ is the wavelet base function, and $\psi^*(\cdot)$ is the complex conjugate function of $\psi(\cdot)$.

For the wavelet transform, the most important process is to find an appropriate base function $\psi(\cdot)$ according to the characteristics of the signal and other specific requirements.

In this research, the output data of the hybrid system is processed using wavelet transforms. By changing the scale parameter a , we can discover features of the signal. Thus, the characteristic parameters of the output signal can be extracted and used as input to the neural network training for detailed classification.

2.2.3. ART2. The overall structure of the ART2 was created using adaptive resonance theory [33]. Long term memory from down to up (LTM_U) was integrated into the response layer. The filter and orienting neurons were added to identify failures more easily. The ART2 structure is shown in Figure 3.

The ART2 structure includes six parts: sensor layer, response layer, filter neuron, orienting neuron, LTM_D, and output layer. The weighting vectors $\mathbf{W} = [w_1, w_2, \dots, w_l]^T$ in the response layer ($w_1 \sim w_l$ are the corresponding weights of the neuron in the same response layer) and the output vector V_d in LTM_D are the center reference vectors of the from down to up and from up to down memory modes. An angle threshold is set as the classification criterion. By

FIGURE 3: ART2 structure: V_{sl} is the output vector of the sensory layer, V_d is the output vector of the LTM_D (long time memory from up to down), and V_b is the output vector of the orienting neuron and the input vector of the sensory layer.

calculating the angle between the two vectors of a new mode and memorizing the mode in the network, it is possible to judge whether the two modes are in the same class. The angle threshold is expressed using a vigilance parameter, which is saved in a neuron. The problem caused by uncertain a priori information can be solved by adjusting the vigilance parameter. ART2 is able to distinguish a new mode and memorize it; thus a memory function is designed to memorize and save the new failure mode. When diagnosing the hybrid system, the mode can be loaded; therefore, the neural network is continually improved and its diagnostic abilities improve over time.

2.3. Failure Event Classification Based on the Neural Network.

In the framework of presenting hierarchical fault diagnosis engine, event identification results tend to be an important part for obtaining the event transition sequence of hybrid system dynamic processes. Failure event classification is divided to two separate processes, primary and detailed classification. Primary classification is designed to distinguish the failure classes of the hybrid system. Detailed classification is implemented using an ART2 artificial neural network and is able to precisely identify the fault location.

2.3.1. Primary Classification. We assume that the hybrid system has N types of failure modes and define the failure mode set

$$\mathbf{F} = \{F_1, F_2, \dots, F_N\}. \quad (10)$$

We complete the primary classification by integrating the event transition sequence using event identification and the parameter estimation results. Assuming that all failure modes are divided into p ($1 \leq p \leq N$) subclasses, we generate the primary classification rules based on the event

FIGURE 4: Neural network structure of primary classification.

FIGURE 5: ART2 structure for detailed classification.

transition sequence which can be used to diagnose a real system

$$\begin{aligned} F &= \{C_1, C_2, \dots, C_p\}, \\ C_p &= \{f_{p1}, f_{p2}, \dots\}, \end{aligned} \quad (11)$$

where $\bar{C}_i \cap C_j = \emptyset$ and $\alpha C_1 \cup C_2 \cup \dots \cup C_p = F$.

To simplify the diagnostic method, every failure mode subclass, C_i , should be marked accordingly. The corresponding characteristic parameters for every subclass are extracted during the diagnostic process.

We designed an ART2 structure with n inputs and 1 output for the primary classification, as shown in Figure 4. Here the inputs are the characteristic parameters and (or) the event transition sequences of the hybrid system. A simple perceptron is connected to the ART2 output, which translates the output to the failure class number.

2.3.2. Detailed Classification. After the primary classification, each failure class must be diagnosed in detail and memorized by ART2. Each failure subclass has its own ART2 structure for detailed classification. The characteristic parameters of each class are used to train its distinct ART2 structure. The trained neural networks should be able to distinguish different failure events of each class.

We have designed an ART2 structure for detailed classification according to the requirements of hybrid system diagnostics, as shown in Figure 5. The inputs are the characteristic parameters of some failure event, which may be the result of parameter estimation or wavelet transformation (see Section 2.2.1). A simple perceptron is connected to the ART2 output, which translates the output to a failure event number.

3. Study Case: BLDCM Fault Diagnosis

3.1. BLDCM Introduction. We select a three-phase wye-connected BLDCM as the study case. The structure of the three-phase full bridge-driven circuit of the BLDCM is shown in Figure 6. $VT_1, VT_2, VT_3, VT_4, VT_5$, and VT_6 are power units (MOSFET) switched by the output signal of a HALL sensor.

At any moment, only two power units are in the ON state simultaneously: one is in the upper bridge and the other in the lower bridge. Each power switch works continuously within 120 electrical degrees, and the phase on-off state changes every 60 electrical degrees. The on-off shifting rule is shown in Figure 7.

Through analyzing the BLDCM structure, it is obvious that the BLDCM power units have six logic states. Their corresponding equivalent circuits are as shown in Figure 8.

Figure 8 shows that the pairs of states 1 and 4, 2 and 5, and 3 and 6 are similar, except for opposite currents in the armatures. The equivalent mathematical models of State 1 (4), State 2 (5), and State 3 (6) are shown in the following equation:

$$\begin{aligned} U_{d1} &= (R_A + R_B) i_1(t) + (L_A + L_B) \frac{di_1}{dt} + k_{E1} \omega_1(t), \\ U_{d2} &= (R_A + R_C) i_2(t) + (L_A + L_C) \frac{di_2}{dt} + k_{E2} \omega_2(t), \\ U_{d3} &= (R_B + R_C) i_3(t) + (L_B + L_C) \frac{di_3}{dt} + k_{E3} \omega_3(t), \end{aligned} \quad (12)$$

where U_{dj} is the armature voltage of the j th state, $i_j(t)$ the bus current of the j th state, $\omega_j(t)$ the motor rotation speed of the j th ($j = 1, 2, 3$) state, R_A the armature resistance of phase A, R_B the armature resistance of phase B, R_C the armature resistance of phase C, L_A the armature inductance of phase A, L_B the armature inductance of phase B, L_C the armature inductance of phase C, and k_{Ej} the EMF constant of the j th state.

3.2. BLDCM Multidomain Modeling and Simulation. We modeled the BLDCM and its drive circuit model using the Modelica language, as shown on the right in Figure 9. It consists of a three-phase armature, back EMF module, position sensor, and inertia module. The two parts of the system are integrated by an interface between their domains.

The parameters of the BLDCM are bus voltage $U = 28$ V, rated speed $\omega = 2500$ r/min, stator equivalent resistance $R = 1.4 \Omega$, stator equivalent inductance $L - M = 0.00215$ H, back EMF constant $k_E = 0.1$ Nm/A, pole pairs $p = 1$, rotor inertia moment $J = 2.5 \cdot 10^{-4}$ kg·m², and load $T_L = 0.2$ Nm.

The modeling and simulating results for normal behaviors and some common failure events are shown in Figure 10.

FIGURE 6: BLDCM/drive card/circuit structure.

TABLE 1: Failure event information for BLDCM.

Failure class number	Description	Failure position	Failure event number
A	Normal	No failure	F_0
B_1	Positive deflection of armature resistance R $\Delta R > 0$	Armature A	F_1
		Armature B	F_2
		Armature C	F_3
B_2	Negative deflection of armature resistance R $\Delta R < 0$	Armature A	F_4
		Armature B	F_5
		Armature C	F_6
C	Armature break	Armature A	F_7
		Armature B	F_8
		Armature C	F_9
D	MOSFET break	VT_1	F_{10}
		VT_2	F_{11}
		VT_3	F_{12}
		VT_4	F_{13}
		VT_5	F_{14}
		VT_6	F_{15}
E	Interarmatures short	Armature A	F_{16}
		Armature B	F_{17}
		Armature C	F_{18}

Note: ΔR = real resistance value-rated resistance value.

FIGURE 7: BLDCM MOSFET on-off rule.

Figure 10(c) shows that when VT_5 breaks, the armature C can only be ON in a negative direction; so the armature C has only negative current. One power unit fault will affect two of the six states during one cycle. The bus current of these two states is nearly zero (Figure 10(b)), which causes the motor rotation speed to decrease and oscillate simultaneously.

As shown in Figure 10(c), when the armature C breaks, it has zero current, and it is obvious it will affect four of the six states during one cycle. The bus current of these four states is nearly zero. This causes the motor rotation speed to decrease with strong oscillations.

3.3. BLDCM Fault Diagnosis

3.3.1. Failure Event Analysis. BLDCM failure events can be categorized into four main classes: motor power unit (MOSFET) break, armature break, interarmatures shorting, and armature resistance increase. This paper focuses on these four classes of failure events and diagnoses. We consider the eighteen failure events shown in Table 1.

FIGURE 8: Equivalent circuits of BLDCM in one period.

3.3.2. BLDCM Event Analysis and Identification. As described above (Figure 7), only two power units work simultaneously, and the phase condition changes every one-sixth cycle (60 electrical degrees). Six events occur in sequence during one cycle, and the ON sequence of the armatures is A-B, A-C, B-C, B-A, C-A, and C-B. The on-off control of the MOSFETs and armatures is related to the rotator position, as shown in Table 2 (MOSFET: 1 ON, 0 OFF; armature: 1 positive conduction, -1 positive conduction, 0 OFF).

The analysis in Table 2 shows that the BLDCM has six main events, and the event transition sequence is as shown in Figure 11.

One cycle completes within 360 electrical degrees. According to Table 2, a BLDCM system event can be identified by analyzing the control signal of six MOSFETs. Furthermore, one cycle of bus current information can be extracted and be used for further diagnosis by parameter estimation and data preprocessing.

FIGURE 9: Multidomain model of BLDCM. Diode: $D_1, D_2, \dots, D_6$; MOSFET: $VT_1, VT_2, \dots, VT_6$; control interface: $a_1, a_2, b_1, b_2, c_1, c_2$.

FIGURE 10: Simulation results for BLDCM.

TABLE 2: MOSFETs and armatures state at different positions.

Electrical degree	VT ₁	VT ₂	VT ₃	VT ₄	VT ₅	VT ₆	A	B	C	Event number
0~60°	1	0	0	1	0	0	1	-1	0	1
60°~120°	1	0	0	0	0	1	1	0	-1	2
120°~180°	0	0	1	0	0	1	0	1	-1	3
180°~240°	0	1	1	0	0	0	-1	1	0	4
240°~300°	0	1	0	0	1	0	-1	0	1	5
300°~360°	0	0	0	1	1	0	0	-1	1	6

FIGURE 11: BLDCM event transition sequence.

3.3.3. *Parameter Estimation.* The following variables are used for parameter estimation:

$$\begin{aligned}
 R_1 &= R_A + R_B, \\
 R_2 &= R_A + R_C, \\
 R_3 &= R_B + R_C, \\
 L_1 &= L_A + L_B, \\
 L_2 &= L_A + L_C, \\
 L_3 &= L_B + L_C.
 \end{aligned} \tag{13}$$

Combining (12) with (13) results in the following:

$$\begin{aligned}
 U_{d1} &= R_1 i(t) + L_1 \frac{di}{dt} + k_{E1} \omega(t), \\
 U_{d2} &= R_2 i(t) + L_2 \frac{di}{dt} + k_{E2} \omega(t), \\
 U_{d3} &= R_3 i(t) + L_3 \frac{di}{dt} + k_{E3} \omega(t).
 \end{aligned} \tag{14}$$

Equations (14) can be generalized as

$$U_d = Ri(t) + L \frac{di}{dt} + k_E \omega(t). \tag{15}$$

Based on the requirements of the parameter estimation method in Section 2.2.1, (15) can be rewritten as

$$\begin{aligned}
 y(t) &= U_d, \\
 \psi^T(t) &= \left[i(t) \quad \frac{di}{dt} \quad \omega(t) \right], \\
 \theta^T &= [R \quad L \quad k_E].
 \end{aligned} \tag{16}$$

After sampling the data of the bus voltage U_d , bus current i , and rotation speed ω , estimating the values of R_1 , R_2 , R_3 , L_1 , L_2 , L_3 , and k_E using the relationships expressed in (13), we obtain

$$\begin{aligned}
 \hat{R}_A &= \frac{1}{2} (\hat{R}_1 - \hat{R}_2 + \hat{R}_3), \\
 \hat{R}_B &= \frac{1}{2} (\hat{R}_1 + \hat{R}_2 - \hat{R}_3), \\
 \hat{R}_C &= \frac{1}{2} (\hat{R}_2 + \hat{R}_3 - \hat{R}_1), \\
 \hat{L}_A &= \frac{1}{2} (\hat{L}_1 - \hat{L}_2 + \hat{L}_3), \\
 \hat{L}_B &= \frac{1}{2} (\hat{L}_1 + \hat{L}_2 - \hat{L}_3), \\
 \hat{L}_C &= \frac{1}{2} (\hat{L}_2 + \hat{L}_3 - \hat{L}_1),
 \end{aligned} \tag{17}$$

where $\hat{R}_A$, $\hat{R}_B$, and $\hat{R}_C$ are the estimated values of the BLDCM resistances of the three armatures, $\hat{L}_A$, $\hat{L}_B$, and $\hat{L}_C$ are the estimated values of the inductances (self-inductance + mutual inductance) of the armatures, and $\hat{k}_E$ is the estimated value of the back EMF constant. Defining $\hat{R} = (1/3)(\hat{R}_A + \hat{R}_B + \hat{R}_C)$ and $\hat{L} = (1/3)(\hat{L}_A + \hat{L}_B + \hat{L}_C)$, the BLDCM estimated values under different failure events are shown in Table 3.

As shown in Figure 10, both MOSFET and armature failures will cause the bus current to go to zero. Therefore, only two variables are defined: one is the current threshold $i_{\text{ref}} = i_{\text{max}}/100$ and the percentage $\delta\%$ which represents the percentage of bus current i exceeding i_{ref} during one cycle. Using simulations and analyses of all the failure events, we calculate the $\delta\%$ as shown in Table 4.

3.3.4. *Training the Neural Network Structure for Primary Classification.* Based on the value of $\delta\%$, $\hat{R}$, and $\hat{L}$, all 18 failure events can be divided into five primary classes: normal, armature resistance increase, MOSFET break, armature

TABLE 3: Normal value and estimation value under different failure events.

Event number	R/Ω	L/H	k_E	$\hat{R}/\Omega$	$\hat{L}/H$	$\hat{k}_E/(V \cdot \min/r)$
F_1	1.493	0.00215	0.06667	1.499	0.00205	0.0666
F_2	1.493	0.00215	0.06667	1.448	0.00215	0.0669
F_3	1.493	0.00215	0.06667	1.487	0.00220	0.0659
F_4	1.307	0.00215	0.06667	1.336	0.00205	0.0666
F_5	1.307	0.00215	0.06667	1.365	0.00215	0.0669
F_6	1.307	0.00215	0.06667	1.316	0.00220	0.0659
F_7	1.4	0.00215	0.06667	1.4123	0.00204	0.0665
F_8	1.4	0.00215	0.06667	1.3827	0.00213	0.0666
F_9	1.4	0.00215	0.06667	1.3964	0.00218	0.0669
F_{10}	1.4	0.00215	0.06667	1.3976	0.00205	0.0659
F_{11}	1.4	0.00215	0.06667	1.4056	0.00215	0.0663
F_{12}	1.4	0.00215	0.06667	1.4102	0.00220	0.0669
F_{13}	1.4	0.00215	0.06667	1.3970	0.00219	0.0658
F_{14}	1.4	0.00215	0.06667	1.4037	0.00213	0.0671
F_{15}	1.4	0.00215	0.06667	1.4098	0.00217	0.0636
F_{16}	1.307	0.001892	0.06667	1.2943	0.00188	0.0669
F_{17}	1.307	0.001892	0.06667	1.2986	0.00184	0.0659
F_{18}	1.307	0.001892	0.06667	1.3097	0.00186	0.0661

TABLE 4: Bus current duty cycle.

Failure event number	$\delta\%$	Event number	$\delta\%$	Failure event number	$\delta\%$
F_1	100%	F_7	33.3%	F_{13}	66.7%
F_2	100%	F_8	33.3%	F_{14}	66.7%
F_3	100%	F_9	33.3%	F_{15}	66.7%
F_4	100%	F_{10}	66.7%	F_{16}	100%
F_5	100%	F_{11}	66.7%	F_{17}	100%
F_6	100%	F_{12}	66.7%	F_{18}	100%

break, and interarmatures short. We designed an ART2 structure with three inputs and one output for primary classification, as shown in Figure 12. A simple perceptron is connected to the ART2 output, which translates the output to the failure class number. As the inputs to the ART2 structure, parameters $\delta\%$, $\hat{R}$, and $\hat{L}$ are used to train the neural network automatic classifications. In Figure 12,

$$\begin{aligned}
 0 &\leq \delta \leq 100\%, \\
 R^* &= \begin{cases} 1 & \hat{R} > 1.05 \times R \\ 0 & \hat{R} \approx R \\ -1 & \hat{R} < 0.95 \times R, \end{cases} \\
 L^* &= \begin{cases} 1 & \hat{L} > 1.05 \times L \\ 0 & \hat{L} \approx L \\ -1 & \hat{L} < 0.95 \times L. \end{cases}
 \end{aligned} \quad (18)$$

FIGURE 12: Network structure of BLDCM primary classification.

TABLE 5: Input and output of primary classification network.

Failure class number	Input			Output number
	δ	R^*	L^*	
A	100%	0	0	P1
B_1	100%	1	0	P2
B_2	100%	-1	0	P3
C	33.3%	0	0	P4
D	66.7%	0	0	P5
E	100%	-1	-1	P6

Output P1: normal state.

Output P2: armatures resistance increase.

Output P3: armatures resistance decrease.

Output P4: bus current is zero during four of all the six states of one cycle; one MOSFET break is possible.

Output P5: bus current is zero during four of all the six states of one cycle; one armature break is possible.

Output P6: interarmatures short.

The output of the simple perceptron is the failure class number. The input/output relationships of the neural network designed for BLDCM primary classification are shown in Table 5.

3.3.5. Training the Neural Network Structure for Detailed Classification. The primary classification only determines the information of the failure class; so exact information, such as failure event position and mechanism, must be obtained by further analysis of the bus current. In this research, we have selected one cycle of data for the bus current to be processed using the wavelet transform (9). We use this to find the varying adjustments of the wavelet coefficient under different scale parameters, which are then used for detailed classification of the BLDCM.

After the primary classification, three failure classes of P2, P3, and P6 can be determined by using the fluctuations of the estimated value of the resistance of all three armatures. The two failure classes of P4 and P5 can be determined using the wavelet coefficients of the six states during one cycle. Owing to the characteristics of the BLDCM bus current, we determine that the appropriate wavelet base function was the Gauss wavelet (shown in (19))

$$\psi(t) = C_1 e^{-\pi t^2} + C_2 t^2 e^{-\pi t^2}. \quad (19)$$

FIGURE 13: Bus current under the P4 and P5 failure classes.

For the three failure classes of P2, P3, and P6, we designed three ART2 structures with three inputs and one output (Figure 5). The three inputs are R_A^* , R_B^* , and R_C^*

$$R_i^* = \begin{cases} 1 & \hat{R}_i > 1.05 \times R_i \\ 0 & \hat{R}_i \approx R_i \\ -1 & \hat{R}_i < 0.95 \times R_i \end{cases} \quad i = A, B, C. \quad (20)$$

A priori information (input and output) of all three ART2 structures is shown in Table 6.

For the two failure classes of P4 and P5, it is difficult to perform a detailed classification using only the estimated resistance and inductance values. The wavelet coefficients of all six states were obtained by processing the bus current data using the continuous wavelet transform method. Bus currents under failures P4 and P5 are shown in Figure 13.

TABLE 6: A priori information for detailed classification (for P2, P3, and P6).

Failure event number	Input			Output number
	R_A^*	R_B^*	R_C^*	
F_1	1	0	0	2-1
F_2	0	1	0	2-2
F_3	0	0	1	2-3
F_4	-1	0	0	3-1
F_5	0	-1	0	3-2
F_6	0	0	-1	3-3
F_{16}	-1	0	0	6-1
F_{17}	0	-1	0	6-2
F_{18}	0	0	-1	6-3

FIGURE 14: Wavelet coefficients ($a = 200$).

TABLE 7: A priori information for detail classification (for P4 and P5).

Failure event number	Input						Output number
	$\bar{W}_{\psi_1}$	$\bar{W}_{\psi_2}$	$\bar{W}_{\psi_3}$	$\bar{W}_{\psi_4}$	$\bar{W}_{\psi_5}$	$\bar{W}_{\psi_6}$	
F_7	-19.6955	-10.3186	41.8848	-19.6636	-10.2622	41.7777	4-1
F_8	-10.2973	41.8541	-19.6471	-10.2712	41.8822	-19.6094	4-2
F_9	41.9165	-19.6313	-10.3492	41.8752	-19.6655	-10.2772	4-3
F_{10}	-13.3213	-11.4500	15.1863	1.7025	0.7907	12.0575	5-1
F_{11}	1.7013	0.7901	12.0584	-13.3205	-11.4317	15.1808	5-2
F_{12}	0.7925	12.0552	-13.3209	-11.4515	15.1770	1.7104	5-3
F_{13}	-11.4442	15.1574	1.7079	0.7866	12.0656	-13.3237	5-4
F_{14}	15.1883	1.7006	0.7901	12.0585	-13.3206	-11.4352	5-5
F_{15}	12.0627	-13.3219	-11.4540	15.1871	1.7010	0.7897	5-6

The data are processed using the Gauss2 wavelet (see (9)). Figure 14 shows the wavelet coefficient curves obtained after extracting the wavelet coefficient $(W_{\psi}f)(a, \tau)(W_{\psi})$ (scale parameter $a = 200$). The mean value of the wavelet coefficients is extracted as the characteristic parameters (see Table 7), which are then used for further diagnosis.

We designed two ART2 structures with six inputs and one output for the two failure classes of P4 and P5, as shown in Figure 15. The inputs are the wavelet coefficient mean values of the six states $(\bar{W}_{\psi_1}, \bar{W}_{\psi_2}, \dots, \bar{W}_{\psi_6})$. These six ART2 structures were trained using the wavelet coefficients of the six states for detailed classification.

A priori information for training the P4 and P5 class failures is shown in Table 7.

3.3.6. Validation of the Fault Diagnostic System. After training the neural network, the diagnostic engine has the ability to diagnose the BLDCM. We introduced a fault into the BLDCM system: a series connecting resistance ($R_a = 0.33 \Omega$) with the armature B. The measured rotation speed and bus current are shown in Figure 16.

FIGURE 15: Neural network structure of detailed classification (for P4, P5).

By measuring the position signal using a HALL sensor, the event identification can be finished to obtain the event

FIGURE 16: Rotation speed and bus current under failure.

TABLE 8: Estimated parameters.

Parameter estimation	$\hat{R}/\Omega$	$\hat{L}/H$	$\hat{k}_E/(V \cdot \text{min}/r)$
	1.513	0.00212	0.0658
Primary classification	Input parameters		
	$\delta\%$	R^*	L^*
	100%	1	0
	Output number		
	P2		
Detailed classification	Input parameters		
	R_A^*	R_B^*	R_C^*
	0	1	0
	Output number		
	2-2		

transition sequence. Once the system failure occurs, three cycles of bus current are collected and used for parameter estimation. The results of the parameter estimation are shown in Table 8.

Based on the bus current shown in Figure 16, $\delta\%$ is 100%. Combining this with the estimated parameters of $\hat{R}$ and $\hat{L}$, the inputs and output of the ART2 for primary classification are shown in Table 8, the primary classification output number is P2, and the failure is in class B1.

We obtained the detailed classification parameters for class B1 (see Table 6) using parameter estimation (Table 8). These were entered into the detailed ART2 and obtained an output number of 2-2. According to Table 6, the failure event number is F2, which means the phase B armature resistance has increased (see Table 1). This is consistent with the actual failure injected into the system.

4. Conclusions

This research developed a hierarchical fault diagnostic system, which integrated the theories on multidomain modeling, event identification, and ART2s. We demonstrated the validity of the fault diagnostic system using a typical hybrid system: BLDCM.

The three main achievements of this research are as follows.

(1) The normal and fault behaviors of the hybrid system appear to be able to be modeled and simulated conveniently by the Modelica language, which would provide sufficient prior information for the fault diagnostic system.

(2) Using event identification, parameter estimation, wavelet transform, and ART2 technologies, we implemented a hierarchical fault diagnostic engine. Firstly, we obtained the event transition sequence of the system dynamic process using event identification and the estimated parameters of the hybrid system based on the extracted data of some subprocesses. The failure event was then primarily classified by integrating the event transition sequences and estimated parameters. Finally, we designed an ART2 structure for each failure class which tends to efficiently finish the detailed classification of the failure event.

(3) We have demonstrated that our fault diagnostic method was able to identify both known and unknown failure events using a priori information of the hybrid system. Particularly, for a system with less response information under a failure event, the accuracy of diagnosis seems to be higher. Moreover, the presented method integrates the advantages of current quantitative and qualitative diagnostic procedures and can distinguish between failures caused by parametric and abrupt structure faults. Another advantage of our method is that it can remember unknown failure types and automatically extend the ART2 library, which is extremely useful for complex hybrid systems.

The ART2 used in this paper memorizes failure information through the connection weights between neurons. Therefore, diagnosis results marked by meaningless numbers have poor interpretability. A proposed future study is to develop a corresponding expert system to create the relationship between the outputs of the ART2 network and real phenomena.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Research for this paper was partially supported by National Natural Science Foundation of China (Grant no. 51205007) and Ph.D. Foundation of Ministry of Education of China (Grant no. 20121102120046). In addition, the authors would like to thank the anonymous reviewers who have helped to improve the paper.

References

- [1] A. van der Schaft and H. Schumacher, *An Introduction to Hybrid Dynamical Systems*, vol. 251 of *Lecture Notes in Control and Information Sciences*, Springer, London, UK, 2000.
- [2] J. Lunze and F. Lamnabhi-Lagarrigue, *Handbook of Hybrid Systems Control: Theory, Tools, Applications*, Cambridge University Press, Cambridge, UK, 2009.
- [3] G. K. Fourlas, K. J. Kyriakopoulos, and N. J. Krikelis, "Fault diagnosis of hybrid systems," in *Proceedings of the 20th IEEE International Symposium on Intelligent Control (ISIC '05) and the 13th Mediterranean Conference on Control and Automation (MED '05)*, pp. 832–837, June 2005.
- [4] R. Nouailletas, E. Mendes, and D. Koenig, "Hybrid modeling and identification of dry friction systems, application to a clutch actuator," *Control Engineering Practice*, vol. 18, no. 8, pp. 904–917, 2010.
- [5] S. Narasimhan and G. Biswas, "Model-based diagnosis of hybrid systems," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 37, no. 3, pp. 348–361, 2007.
- [6] T. A. Henzinger, "The theory of hybrid automata," in *Proceedings of the 11th IEEE Symposium on Logic in Computer Science (LICS '96)*, pp. 278–292, New Brunswick, NJ, USA, July, 1996.
- [7] M. F. Karoui, H. Alla, and A. Chatti, "Monitoring of dynamic processes by rectangular hybrid automata," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 4, pp. 766–774, 2010.
- [8] R. Mohammadi, S. Hashtrudi-Zad, and K. Khorasani, "Diagnosis of hybrid systems: part 2—residual generator selection and diagnosis in the presence of unreliable residual generators," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 3340–3345, October 2009.
- [9] L. Tromp, A. Benveniste, and M. Basseville, "Fault detection and isolation in hybrid systems: a petri-net approach," in *Proceedings of the 14th IFAC World Congress*, pp. 79–84, Beijing, China, 1999.
- [10] V. Calderaro, C. N. Hadjicostis, A. Piccolo, and P. Siano, "Failure identification in smart grids based on Petri net modeling," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4613–4623, 2011.
- [11] K. Renganathan and V. Bhaskar, "Performance evaluation and model checking in systems modeled as Hybrid Petri nets," *Applied Mathematical Modelling*, vol. 36, no. 8, pp. 3941–3947, 2012.
- [12] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, "Monitoring and fault diagnosis of hybrid systems," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 35, no. 6, pp. 1225–1240, 2005.
- [13] X. D. Zhao, H. Liu, J. F. Zhang, and H. Y. Li, "Multiple-mode observer design for a class of switched linear systems," *IEEE Transactions on Automation Science and Engineering*, 2013.
- [14] X. Zhao, X. Liu, S. Yin, and H. Li, "Improved results on stability of continuous-time switched positive linear systems," *Automatica*, vol. 50, no. 2, pp. 614–621, 2014.
- [15] A. Philippot, M. Sayed-Mouchaweh, V. Carré-Ménétrier, and B. Riera, "Generation of candidates' tree for the fault diagnosis of discrete event systems," *Control Engineering Practice*, vol. 19, no. 9, pp. 1002–1013, 2011.
- [16] S. A. Arogeti, D. Wang, and C. B. Low, "Mode identification of hybrid systems in the presence of fault," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1452–1467, 2010.
- [17] M. Gholami, H. Schioler, and T. Bak, "Active fault diagnosis for hybrid systems based on sensitivity analysis and adaptive filter," in *Proceedings of the 20th IEEE International Conference on Control Applications (CCA '11)*, pp. 1171–1177, September 2011.
- [18] S. Yin, X. Yang, and H. R. Karimi, "Data-driven adaptive observer for fault diagnosis," *Mathematical Problems in Engineering*, vol. 2012, Article ID 832836, 21 pages, 2012.
- [19] J. Y. Yu, G. H. Sun, and H. R. Karimi, "Fault-reconstruction-based cascaded sliding mode observers for descriptor linear systems," *Mathematical Problems in Engineering*, vol. 2012, Article ID 623426, 20 pages, 2012.
- [20] M. Chadli, S. Aouaouda, H. R. Karimi, and P. Shi, "Robust fault tolerant tracking controller design for a VTOL aircraft," *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2627–2645, 2013.
- [21] H. R. Karimi, "Robust synchronization and fault detection of uncertain master-slave systems with mixed time-varying delays and nonlinear perturbations," *International Journal of Control, Automation and Systems*, vol. 9, no. 4, pp. 671–680, 2011.
- [22] M. J. Daigle, X. D. Koutsoukos, and G. Biswas, "A qualitative event-based approach to continuous systems diagnosis," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 780–793, 2009.
- [23] P. Bhowal, D. Sarkar, S. Mukhopadhyay, and A. Basu, "Fault diagnosis in discrete time hybrid systems—a case study," *Information Sciences*, vol. 177, no. 5, pp. 1290–1308, 2007.
- [24] J. Lunze, "Diagnosis of quantized systems based on a timed discrete-event model," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 30, no. 3, pp. 322–335, 2000.
- [25] X. D. Koutsoukos, P. J. Antsaklis, J. A. Stiver, and M. D. Lemmon, "Supervisory control of hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1026–1049, 2000.
- [26] V. Sanz, A. Urquia, F. E. Cellier, and S. Dormido, "Modeling of hybrid control systems using the DEVSLib Modelica library," *Control Engineering Practice*, vol. 20, no. 1, pp. 24–34, 2012.
- [27] P. Fritzson and P. Bunus, "Modelica-A general object-oriented language for continuous and discrete-event system modeling and simulation," in *Proceedings of the 35th Annual Simulation Symposium*, 2002.
- [28] H. Elmqvist, S. E. Mattsson, and M. Otter, "Modelica—a language for physical system modeling, visualization and interaction," in *Proceedings of the IEEE International Symposium on Computer Aided Control System Design*, pp. 630–639, August 1999.
- [29] V. Sanz, A. Urquia, F. E. Cellier, and S. Dormido, "System modeling using the parallel DEVS formalism and the Modelica language," *Simulation Modelling Practice and Theory*, vol. 18, no. 7, pp. 998–1018, 2010.

- [30] C. Nytsch-Geusen, "The use of the UML within the modelling process of modelica-models," in *Proceedings of the 1st International Workshop on Equation-Based Object-Oriented Languages and Tools (EOOLT '07)*, pp. 1–11, Berlin, Germany, July 2007.
- [31] C. Martin-Villalba, A. Urquia, and S. Dormido, "Development of virtual-labs for education in chemical process control using Modelica," *Computers and Chemical Engineering*, vol. 39, pp. 170–178, 2012.
- [32] S. Mortada, A. Zoughaib, D. Clodic, and C. Arzano-Daurelle, "Dynamic modeling of an integrated air-to-air heat pump using Modelica," *International Journal of Refrigeration*, vol. 35, no. 5, pp. 1335–1348, 2012.
- [33] G. A. Carpenter and S. Grossberg, "ART 2: self-organization of stable category recognition codes for analog input patterns," *Applied Optics*, vol. 26, no. 23, pp. 4919–4930, 1987.
- [34] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "ART 2-A: an adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks*, vol. 4, no. 4, pp. 493–504, 1991.
- [35] O. Moseler and R. Isermann, "Application of model-based fault detection to a brushless DC motor," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 1015–1020, 2000.
- [36] K. Yu, F. Yang, H. Guo, and J. Xu, "Fault diagnosis and location of brushless DC motor system based on wavelet transform and artificial neural network," in *Proceedings of the International Conference on Electrical Machines and Systems (ICEMS '10)*, pp. 1048–1052, October 2010.
- [37] R. Isermann, *Fault-Diagnosis Applications: Model Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-Tolerant System*, Springer, New York, NY, USA, 2011.