Model-Based Monitoring of a Train Passenger Access System

Abderraouf Boussif, Mohamed Ghazel

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 14, 2018, accepted July 23, 2018, date of publication July 31, 2018, date of current version August 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2860966

Model-Based Monitoring of a Train Passenger Access System

ABDERRAOUF BOUSSIF [1,2] AND MOHAMED GHAZEL [1,2]
[1]University Lille Nord de France, 59000 Lille, France
[2]COSYS/ESTAS, IFSTTAR, 59650 Villeneuve d'Ascq, France

Corresponding author: Abderraouf Boussif (abderraouf.boussif@ifsttar.fr)

This work was supported by the ELSAT2020 Project. ELSAT2020 is co-financed by the European Union with the European Regional development Fund, the French state, and the Hauts de France Region Council.

ABSTRACT The passenger access system (PAS) is a complex mechatronic train onboard module with high reliability and safety requirements. This module fulfills one of the dozen main onboard functions onboard train. Consequently, any related fault occurrence may have a serious impact on the safety and availability of the whole train operation. In this context, developing effective automated monitoring and diagnostic techniques for the PAS, as early as from the design phase of the system, becomes an essential and challenging task. In this paper, we carry out a monitoring study on this system, while considering a sufficiently high-level abstraction perspective that allows for adapting discrete event models representing the behavior of the system. First, we establish a Petri net behavioral model that includes the nominal operating mode as well as various faulty behaviors. Then, based on the established Petri net models, a fault detection approach is used to investigate the diagnosability property and synthesize the diagnosers regarding different predetermined classes of failures. Finally, we show how the outputs of the diagnosability analysis can help make efficient design choices that allow for improving the safety of the whole system.

INDEX TERMS Railway safety, fault monitoring, passenger access system, Petri nets, modeling.

I. INTRODUCTION

Safety critical systems are systems whereby human safety depends on the correct operation of the system. In such systems, failures may lead to serious human and material damages [1]. Railway control systems are considered as being safety-critical [2], [3] since their failures may cause serious consequences such as loss of human lives, severe injuries, and a large scale of material and environmental damages as well as considerable economic penalties.

From the engineering point of view, to comply with the requirements of the safety-related standards and to achieve the desired safety level (referred to as the Safety Integrity Level or SIL), much attention needs be paid to railway control design and adequate monitoring and diagnosis means needs be developed [4]–[7]. In fact, the purpose of monitoring and fault diagnosis is to *detect*, *isolate* and *locate* failures as early as possible. In the context of railway applications and in order to fulfill the performance, comfort, and safety goals, developing effective monitoring techniques becomes essential from as early as the design phase of the system. In particular, having efficient train onboard diagnosis tools is of a great interest since this minimizes, or even prevents, downtime by effectively detecting and identifying failures.

A. THE PASSENGER ACCESS SYSTEM

The passenger access system (PAS hereafter) is a safety-critical module that fulfills one of the main functions onboard passenger trains. It is a complex mechatronic distributed module that manages the transit of passengers between the station platform and the train [8], [9]. The passenger access function is implemented partly within the train central control (in the locomotive cab, and prevented from being used by unauthorized persons), and partly locally at each train coach.

From the diagnosis point of view, the PAS is considered as one of the high failure-rate subsystems of onboard railway systems. Indeed, although the PAS makes up only 2-3% of the cost of the passenger rail cars, experience with railway systems has shown that the PAS system is responsible for 30 to 40% of the failures in operating trains; moreover, it is estimated that they are responsible for as much as 25% of the maintenance costs [10]. Furthermore, for passenger trains, the malfunctioning of the PAS is one of the main causes of delays, and can be a source of accidents that may involve passengers and/or train crew. Indeed, a failure impacting one single door panel on a passenger train at a station often causes delay on the concerned train and impacts the whole operation

on the line [11]. Whether localized passenger door controls affect Mean-Time-Between- Failure (MTBF) or Mean-Time-To-Failure (MTTF) currently seems to be unknown [12]. Emergency situations continue to occur and, at times, they result in occupant casualties. In certain cases, the delay, difficulty, or inability of passengers and crew to evacuate the train can contribute to the number and the degree of severity of casualties in emergency situations (fire for example).

B. FAULT DIAGNOSIS OF THE PAS SYSTEM

As reported in [13], the existing approaches for fault diagnosis of railway on-board systems can be divided into two categories: *data-driven* and *model-based* methods. In general, model-based diagnosis uses logical and mathematical models of the monitored system, while data-driven diagnosis uses models learned from training available data for nominal and degraded conditions [14]. Regarding the diagnosis of the PAS, both approaches have been investigated.

Pereira *et al.* [15] dealt with the train door fault prediction using data mining techniques, where an efficient low-pass filter is proposed to reduce the false alarm rate. In [16], an estimated mathematical model obtained from a test-rig data set is used to guide the fault diagnosis of some mechanical and electrical failures in electric train doors. The main obstacle of this approach lies in the signal noise and therefore it is more applicable for fault review. A mathematical model is also proposed in [17] by using parameter estimation approach to get the physical parameters of the system on different working conditions. Then, the *principal component analysis* [18] and *rough set theory* are brought into play in order to perform the diagnosis task. In [19], an ontology approach based on the knowledge space of the system is developed to guide the fault detection process and better automated knowledge discovery to improve diagnosis of pneumatic train doors. Lehrasab *et al.* [11] have dealt with the same issue using dynamic neural network fault diagnosis method. Based on data recorded using a test bench, the Han *et al.* [20] design a probabilistic discriminator, which is used to perform the online predictive diagnosis by discriminating between normal signals and suspicious signals of the door system due to malfunctions or exterior events. Recently, Cauffriez *et al.* [21] have investigated the Bond graph formalism to generate a reference model of a mechatronic train door system and then proposed a global model-based diagnosis system for the generation of fault indicators and residual thresholds in presence of door failures (motor failures, superelevation, etc.). One can notice that even-though there are valuable results in the aforementioned works, they particularly dealt with mechanical, electrical, or pneumatic failures based on the (physical) system behavior. However, they do not consider the logical and sequential failures which may affect the (discrete) dynamic of the system or its control unit, such as communication and synchronization failures.

C. FAULT DIAGNOSIS IN DISCRETE EVENT SYSTEMS

From a theoretical point of view and at a high level of abstraction, discrete event systems (DES) are more suitable for monitoring and model-based diagnosis of complex dynamic systems, due to the convenience of their associated means of analysis [22]–[25]. In particular, using DES formalisms to model railway control systems allows for expressing the system specifications, assessing operational requirements and performing re-design [24]–[26]. Besides, the use of DES modeling formalisms is highly recommended in railway standards (IEC 61508-3 and EN 50128) to design SIL3 or SIL4[1] safety-critical systems. In particular, the Petri net formalism (PN) has proved to be a powerful framework for the modeling, control and verification of railway systems [25], [27]. This is mainly due to its mathematical foundation and expressiveness capabilities, its graphical representation, and the existence of a wide range of software tools for simulation and formal analysis of PNs. In addition, the several extensions of PNs, such as (time/timed, stochastic, colored and continuous PNs) allow for handling various classes of complex dynamic systems from different modeling viewpoint (discrete, continuous, and hybrid systems).

Regarding monitoring and fault diagnosis, PNs have been widely investigated in the literature and several approaches have been developed (see the recent overview [28] and references therein). In fact, the aim of PN based diagnosis methods is to use the structure, the analytical capabilities, and the intrinsically distributed nature of PN models to reduce the computational complexity of diagnosis problems by avoiding the exhaustive enumeration of the system's state space, as well as to deal with some classes of infinite state systems [29].

D. CONTRIBUTION

The present study deals with the diagnosis of some main failures which may affect the PAS from the DES point of view. The investigated failures are the output of a preliminary risk analysis (PRA) that has been carried out on the PAS.[2] In a DES framework, fault diagnosis is often a model-driven process which consists in determining the occurrence of faulty events, based only on the observable part of the system behavior. Therefore, model-based diagnosis requires a complete modeling of the system behavior, considering both its nominal and faulty behaviors. In the present study and for the purpose of performing diagnosis analysis on the PAS, we firstly establish discrete models to depict the behavior of the PAS sub-systems, using the Petri net formalism (PN). The elaborated PN models describe the main operational behavior of the global system, including the nominal operating mode and various faulty scenarios. Based on the established PN models, a diagnoser-based approach, which is called *semi-symbolic diagnoser* technique (SSD), is used to investigate

[1] SIL for Safety Integrity Level, which is relative level of risk-reduction provided by a safety function.

[2] The PRA is not within the scope of the present paper.

the diagnosability of the PAS model with respect to a set of predetermined failures. In particular, the SSD technique synthesizes a semi-symbolic diagnoser which is used to verify diagnosability. The symbolic representation of the system state-space allows us to tackle the combinatorial explosion problem, which arises when dealing with such large complex systems.

This article is an extension of a conference paper [30], in which a brief preliminary PAS model and diagnosis mechanisms have been presented. Regarding the conference version:

1) we propose a global PAS modeling, while extending the normal behavior of the PAS, adding the emergency procedure, and integrating a reset mechanism.
2) we deal with three types of failures which may affect the nominal operation of the PAS.
3) In order to analyze the fault diagnosis of the system, we consider an efficient diagnoser-based approach, which performs diagnosability analysis on the PN model without requiring a preliminary generation of the marking graph (as it was the case in [30]).

The remainder of this paper is organized as follows: in Section II, the main components of the PAS are described, while roughly discussing the system operating functions. In Section III, some preliminary notions and notations related to the Petri net and fault diagnosis in the DES framework are presented. Section IV is devoted to the PN modeling of the PAS sub-systems and their dynamics. Section V discusses the diagnosability analysis based on the global elaborated model. Finally, Section VI draws some concluding remarks and points to some future extensions.

II. THE PASSENGER ACCESS SYSTEM: MODELING AND ANALYSIS

In this section, we give a rough description of the PAS and its operation, while sketching out its emergency scenarios and reviewing the main malfunctioning that may impact its nominal behavior.

A. PAS - GENERAL ARCHITECTURE

In a passenger train consisting of a series of connected vehicles, the role of the PAS is to manage the passenger flows (entrance and exit) by controlling the opening/closing of the train doors. The general authorizations (doors opening, unlocking, etc.) are managed by the central control module and operated by the train driver while, locally, passenger-operated open/close control is provided at each individual door. Each train door is equipped with an actuator (using electric or pneumatic motors). This latter is commanded by a local control system which, besides local signals, receives orders and information that are valid for the whole train from the central control located in the driver cab (e.g. train speed, opening authorization from the driver or from a central controller). Depending on the situation, the opening of the door can operate automatically or only if a passenger issues an opening request (e.g., by pushing a button on the train door) [31]. Generally, there exist two types of train door accesses, the first one is set up on the extremity vehicles, equipped with a gap bridge, while the second type is set up on the intermediate vehicles equipped only with a movable step. Indeed, the control of extremity vehicles considers the case of persons with baby strollers and people using wheelchairs.

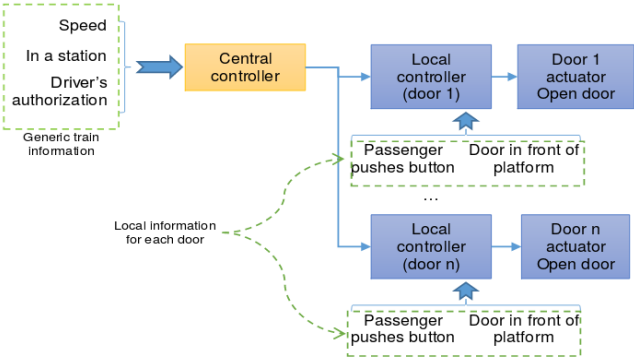

FIGURE 1. Control part of the PAS [31].

In general, the PAS consists in two main parts: (i) the operative part and (ii) the control part. The operative part contains three subsystems:

1) *Door (DR):* a subsystem (often)[3] composed of two opposite-moving sliding leaves which are driven by electric/pneumatic motors;
2) *Movable Step (MS):* it bridges over a wide gap formed between the platform and the train, in order to prevent passengers from falling. Particularly, it facilitates the access for persons with reduced mobility;
3) *Gap Bridge (GB):* it has the same structure and mechanism of MS, and is used to facilitate the access of people in wheelchairs or with baby strollers. The deployment and retraction of the gap bridge are performed by controlling the gap bridge engine adequately, and by reading the filling gap stop point sensors.

These three sub-systems are considered as interd-dependent for two reasons [33]:

- *Mechanical reason:* the three sub-systems are all integrated into a rigid frame and do not interact mechanically.
- *Electronic reason:* the three sub-systems are controlled by the same electronic door control unit, which ensures the interface between the information/commands sent from the driver cabin and each sub-system.

The control part of the PAS is composed of two modules, as illustrated in Figure 1:

1) *Central Controller*: also called *Main Door Control Unit* (MDCU), is an onboard module embedded in the driver cabin, which collects information about the train status (position, speed, etc.) from the sensors onboard

[3]Other types of door exist [32]: sliding, plug, bi-parting, bi-folding, hinged, etc.

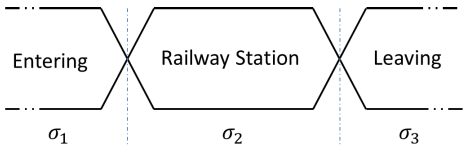

FIGURE 2. The nominal operating cycle.

the train and track-side equipment and sends the controlling authorization (resp. commands) regarding the opening (resp. closing) of train doors to the local controllers.

2) *Local Controller*: also called *Door Control Unit* (DCU), is a module embedded in each train vehicle, which analyzes the information and command requests of the MDCU (opening authorization, closing order, etc.) and the local information (opening request by the passenger, emergency alert, etc.) and then sends the appropriate local (electrical/mechanical) commands to the operative part (door, movable step and gab bridge). The most important task of the DCU is to ensure a safe operation of the door opening and closing operations.

B. THE PAS - GENERAL BEHAVIOR

In this section, we describe the PAS behavior while considering the main commands and (discrete) sensing information used by the PAS components. The scheme depicted in Figure 2 shows the normal operational cycle of the PAS, which is divided into three segments $\sigma_1, \sigma_2, \sigma_3$.

1) *Segment σ_1:* it indicates the train approaching, where the PAS collects information about the train status from the sensors on board the train and track devices (by means of the MDCU and further connected modules);
2) *Segment σ_2:* in this segment, the PAS manages the doors opening/closing according to the information gathered (opening side, platform type, etc.) and the passenger requests (door opening requests, etc.);
3) *Segment σ_3:* it represents the train leaving phase, where the PAS resets its initial status and starts a new operation cycle.

1) MDCU FUNCTIONS

The main information required for the MDCU in order to issue general control orders are:

- *Train Speed (TS)*: this information is generated by the ATESS[4] unit (a system that captures and processes data related to safety) and the WSP unit (wheel slide protection). Two speed thresholds need to be monitored by the PAS:
  - *Speed $v \leq 1\ km/h$:* this is a necessary condition to authorize the doors to unlock in emergency mode;
  - *Speed $v \leq 0,5 km/h$:* this is a necessary condition to authorize the doors to open in nominal mode [31].
- *Station Platform Type (SPT):* this information is provided by the balise reader unit, which captures the signals sent by track-side balises (here at the entrance of a railway station) and indicates whether the platform of the next station is *low* or *high*. In the case of a low platform, the movable step must be deployed in the intermediate train vehicles, in opposition to the case of a high platform, where only the gap bridge is deployed.
- *Station Platform Side (SPS):* this information is also provided by the balise reader unit. It indicates on which side the train must open the doors at the upcoming station (on the right or on the left).

The controlling commands generated by the MDCU and sent to the DCUs are:

- *Opening Permission (OP):* this command is manually ordered by the train driver. It allows the opening of the train doors if the train is stationary at the station (i.e., measured speed $v \leq 0,5\ km/h$);
- *Imminence Closing (IC):* this command is ordered from the driver cabin in order to indicate the imminent closing of all the train doors. A sound alarm signal is then emitted from each DCU until all the train doors are completely closed;
- *General Closing (GC):* this command is issued manually by the train driver and orders the beginning of the door closing process to all DCUs. It is worth noticing that this command has the ultimate priority over all the other commands except the emergency opening command;
- *Gap Bridge Cancellation (GBC):* this command is manually ordered by the train driver. It cancels the deployment of the gap bridge in the extremity vehicles (if there are no people in wheelchairs or with baby strollers present).

2) DCU FUNCTIONS

In addition to the controlling commands received from the MDUC, each DCU also receives (from the passengers) the following local orders:

- *Passengers Request (PR):* each train door is equipped with a push-button that can be pressed by passengers to request the opening of the door;
- *Emergency Door Release (EDR):* this request can be emitted by the passengers (or crew) in case of emergency situations, e.g., fire alarm. It usually requires a cover to be removed or broken and then activated by a hand-pull alarm. It should be noted that the door opening following an EDR request is proceeded only when speed $v \leq 1\ km/h$.

Moreover, the phases of train door opening/closing processes are indicated by some light/sound signals controlled by the DCU. These signals are used to help impaired passengers locate door actuation push-buttons or the doorway opening. Regarding the sound alerts, a unique tone is emitted that maintains a sound level in the approximate range of 2-5 dB above the ambient sound level. This allows the tone

[4]Abbreviation of "*Acquisition et Traitement des Évènements de Sécurité en Statique*".

to stand-out from the environment noises. These light/sound indicators correspond to:

- *Opening Permission Signal (OPS):* this is a continuous light signal located in the interior panel of train doors. It is often associated with the opening push-button, and indicates that the door can be opened.
- *Passenger Request Consideration Signal (PRC):* this is a flashing light signal associated with the opening push-button. It acknowledges passenger requests for door opening;
- *Imminent Closing Signal (IMC):* this is a sound alarm signal that indicates the imminent closing of all the train doors. This signal is emitted from each local door control panel until the complete closing of the doors.

Figure 3 summarizes the functional architecture of the PAS and the distribution of the sensor signals and command flows among the PAS sub-modules.

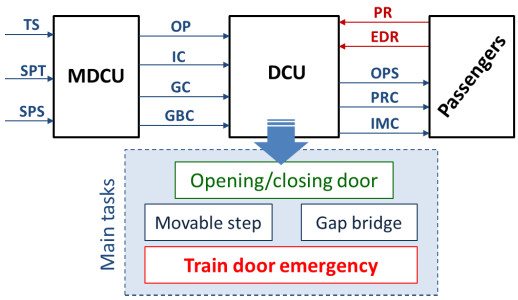

FIGURE 3. Functional architecture of the PAS.

C. THE PAS EMERGENCY EGRESS/ACCESS

One of the most critical functionalities of the PAS is the management of the emergency evacuation. Indeed, emergency situations (e.g., derailments, onboard fires, etc.) require passengers to exit a rail car with (or without) the intervention of the train crew[5] [34].

In fact, emergency egress from a passenger train is a complex process that depends on a number of dynamic factors. Namely, various variables that affect the time necessary for passengers to exit from a passenger train in an unusual or emergency situation must be considered. It should be noted that no methodology exists so far for evaluating the passenger rail car emergency egress system as a whole, or the effects on egress time of failures that may affect this system [35]. Nevertheless, safety standards and guidelines prescribe some general requirements as follows (i) the safest place for passengers is onboard the train, (ii) the doors are locked and passengers are unable to open them even in emergency situations unless the train driver orders the unlocking of the doors, (iii) most emergency situations can be managed without taking passengers off the train, and (iv) passengers are only allowed to open and self evacuate in case of an *extreme* emergency situation.

[5]Besides doors, each train car must be equipped with a combination of emergency windows.

In our study, we only consider extreme emergency situations where passengers are allowed to open and self evacuate provided that the evacuation conditions are fulfilled. In these cases, an emergency brake is launched following the EDR request. Thus, the operational scenario of the PAS is as follows:

1) the DCU receives the EDR signal and transmits it to the MDCU (in order to inform the train driver);
2) the DCU proceeds to unlock the doors once the train speed $v \leq 1km/h$;
3) once the PR button is pushed, the doors open without needing the MS or GB devices to be deployed. Such an operational sequence serves to speed up the evacuation procedure;
4) finally, the doors remain open until the reset action of the PAS, which indicates a return to a normal situation.

Remark: it is important to stress that the preferred means of train evacuation following an emergency would be (as it is currently) for the driver to stop at the closest station, open doors in front of the platform and allow for the train crew to control passenger egress from the train. Passengers opening the doors themselves and self evacuating is a last resort that would only occur in the most extreme situations in which remaining onboard the train could involve a greater danger than that incurred by exiting the train [36].

D. DYSFUNCTIONAL ANALYSIS

Due to its movable components and the frequent interaction with passengers, the PAS is subject to various types of failures and often brings a great proportion of failures that occur onboard trains.

It is commonplace that the frequent failures of the PAS bring much inconvenience to passengers and seriously impact the whole quality of service. Consequently, the safety of the PAS has always been considered as a crucial concern for railway operators, who strive to understand, reduce the risks and improve the reliability of the PAS.

In the current study, we are mainly concerned with technical aspects related to the logical behavior of the PAS. More precisely, our goal is to determine whether the occurrence of some predetermined failures can be detected and identified based on the observable part of the PAS behavior. According to the preliminary risk analysis (PRA) and the failure mode, effect and criticality analysis (FMECA) carried out on the PAS [37], [38], the failures which may affect the system functioning can be roughly classified into two types, regarding their occurrence mode.

1) *permanent Failures:* once such failures occur, they do not disappear (i.e., the system remains indefinitely faulty as long as adequate maintenance operations are not carried out). Often, they induce train immobilization for repair. Examples of permanent failures that my affect the PAS are: the door getting stuck opened/closed, the MS (GB) getting stuck deployed/retracted, the asynchonization between the door opening and the MS deployment, etc.

2) *Intermittent Failures:* such failures occur intermittently. Generally, they do not cause train immobilization; however, they affect its nominal behavior and may reduce its availability. Often, intermittent failures affect the electronic and software devices in the system; for instance, sensor devices, door actuator controller cards, and the spot/sound indicator devices.

III. MODELING AND DIAGNOSIS USING PN

A. PETRI NET MODELING

PNs are a mathematical and graphical notation for modeling parallel and distributed systems. They allow for compact representation of the system state-space and offer good expressiveness, particularly for depicting concurrency and synchronization with non-deterministic behavior [39]. Formally, A PN is *Place/ Transition* structure $N = (P, T, Pre, Post)$, where:

- $P = \{p_1, \ldots, p_i\}$ is a finite set of places;
- $T = \{t_1, \ldots, t_i\}$ is a finite set of transitions;
- *Pre* and *Post* are the pre- and post-incidence mappings.

Below, we discuss some notions and notations related to PNs.

- $C = Post - Pre$ is defined as the *incidence matrix*.
- A *marking* is a vector $m \in \mathbb{N}^{|P|}$ that assigns a non-negative integer to each place. We denote by $m(p_i)$ the marking of place p_i.
- A *marked PN* (N, m_0) is a PN N with a given initial marking m_0. For short, a marked PN will be called PN in the sequel.
- A transition $t_i \in T$ is *enabled* by marking m if $m \geq Pre(\cdot, t_i)$, denoted by $m\,[\,t_i >$. A transition t_i enabled by marking m can fire, yielding to a marking $m' = m + C \cdot \vec{t_i}$, where $\vec{t_i} \in \{0, 1\}^{|T|}$ is a vector in which only the entry associated with transition t_i is equal to 1, the other entries are 0.
- A marking m' is then said to be *reachable* from marking m by firing transition t_i, also denoted by $m\,[\,t_i > m'$.
- A sequence of transitions $s = t_1 t_2 \ldots t_k$ is *executable* at marking m, if $\exists m_1, m_2, m_{k-1}$ s.t. $m\,[\,t_1 > m_1\,[\,t_2 > \cdots m_{k-1}\,[\,t_k >$. This can be denoted as $m\,[\,s >$.
- The *reached marking* m' is computed by $m' = m + C \cdot \pi(s)$, and denoted by $m\,[\,s > m'$, where $\pi(s) = \sum_{i=1}^{k} \vec{t_i}$ is the firing vector corresponding to s.
- A marking m is reachable in (N, m_0) iff there exists a firing sequence s such that $m_0\,[s > m$. The set of all markings reachable from m_0 defines the *reachability set* of (N, m_0) and is denoted $R(N, m_0)$ (may be infinite).
- We denote by L the predix-close language generated by (N, m_0).
- A PN (N, m_0) is *bounded* if the number of tokens in each place does not exceed a finite number $b \in \mathbb{N}$ for any marking reachable from m_0. In this case, the reachability set of the PN is finite.

PARTIAL OBSERVABILITY MODELING

Often, in complex systems, for cost or technical reasons it is not possible to install enough devices for collecting all the information which is needed to monitor the system behavior. This is referred to as partial observability and can be depicted using a partition $T = T_o \uplus T_u$, where T_o is the set of observable transitions (*whose firing can be directly detected by sensor readings or those corresponding to control commands*), and T_u is the set of unobservable transitions (*whose firing depicts some internal activity that can not be captured directly*). Moreover, in order to take into account the faulty behavior of the system in our modeling process, we also consider that the set of unobservable transitions is partitioned into two subsets $T_u = T_f \uplus T_{reg}$ where T_f is the set of fault transitions while T_{reg} corresponds to regular unobservable transitions.

In order to extract the observable behavior, we use a projection mapping: $P_o : T^* \rightarrow T_o^*$, which *erases* the unobservable transitions in any given firing sequence $u \in T^*$. The inverse projection operator P_L^{-1} is defined as $P_L^{-1}(v) = \{u \in T^* \cap L |\, P(u) = v\}$ for $v \in T_o^*$. Simply, P^{-1} re-generates the system executions (sequences of transitions) that correspond to some given observable sequences.

B. FAULT DIAGNOSIS

Fault diagnosis of DESs is often discussed through two main problems: (*offline*) diagnosability analysis and (*online*) diagnosis [40].

DIAGNOSABILITY ANALYSIS

Diagnosability is a property that refers to the ability to detect, isolate and locate the fault occurrences from the observable behavior of the system within a finite delay. The diagnosability property of a PN is defined as follows:

Definition 1 (Diagnosability of PNs): A PN (N, m_0) is said diagnosable w.r.t. fault set T_f and projection mapping P_o if there do not exist two sequences u and v in T^* satisfying the following conditions:

- $\forall t_f \in T_f$, $t_f \notin u$, *i.e., no transition t_f belongs to sequence u;*
- $\exists\ t_f \in T_f$ *such that $t_f \in v$ and the suffix of v starting from t_f can be arbitrarily long;*
- $P_o(u) = P_o(v)$.

According to Definition 1, a PN is diagnosable if after each firing of a faulty transition, one can infer with certainty that the model executes a faulty behavior, within a finite delay after the fault occurrence, based only on the captured observations.

ONLINE DIAGNOSIS

The diagnosis activity is in the main concerned with determining which faulty transition, if any, explains a given observed sequence of transitions. Indeed, online diagnosis consists in determining the current status of the system (faulty or normal) from the online observed behavior.

Formally, the online diagnosis problem can be defined as follows: Given a PN N, T_o and T_f are the sets of observable

and faulty transitions, respectively. For an observed firing sequence $\omega \in P(L \in T^*)$, the diagnosis function $\triangle : P_o(T^*) \times T_f \rightarrow \{N, F, U\}$ assigns a *diagnosis status* to ω, w.r.t. fault set T_f, as follows:

- $\triangle(\omega, T_f) = N$ if $\forall u \in P_L^{-1}(\omega), \forall t_f \in T_f : t_f \notin u$, which means that no fault transitions exist in all the firing sequences, that are consistent with observation ω;
- $\triangle(\omega, T_f) = F$ if $\forall u \in P_L^{-1}(\omega), \exists t_f \in T_f : t_f \in u$, which means that, at least, one fault transition exists in each firing sequences consistent with observation ω;
- $\triangle(\omega, T_f) = U$ if $\exists\, u_1, u_2 \in P_L^{-1}(\omega)$ s.t. $(\forall t_f \in T_f : t_f \notin u_1)$ and $(\exists t_f \in T_f : t_f \in u_2)$, which means that at least two firing sequences consistent with the observation ω exist such that one firing sequence is fault-free while the other one contains at least one faulty transition.

The so-called *diagnoser-based* approaches were the pioneer approaches that deal with both issues (i.e., diagnosability analysis and online diagnosis) [40]–[43]. The main idea behind these approaches is to construct a *diagnoser* automaton, which is a deterministic observer built from the system model itself. A necessary and sufficient condition for analyzing the diagnosability property using the diagnoser-based approach was established in [40]. Moreover, for the systems which are checked to be diagnosable, the diagnoser can be used to perform online diagnosis.

C. THE SEMI-SYMBOLIC DIAGNOSER APPROACH

Recently, we have proposed a diagnoser-based approach, called *the semi-symbolic diagnoser approach* (SSD for short)allow for a subst, which is based on a variant of the classic diagnoser approach introduced in [22] and [40]. The SSD technique has some features which allow for a substantial gain in time and memory consumption [44]. These features consist in:

- explicitly separating the normal and the faulty markings in each diagnoser node. Such a distinction serves to track the faulty and fault-free traces in the diagnoser paths more efficiently [45], [46]. Such a node structure allowed us to formulate simply the necessary and sufficient condition for diagnosability established in [40];
- using an on-the-fly depth-first search procedure, for both synthesizing the diagnoser and checking diagnosability simultaneously;
- practically, the SSD uses a semi-symbolic representation of the diagnoser state-space in order to reduce the memory needed to construct the diagnoser and speed up the verification process. In other words, we combine the enumerative and symbolic representations to encode the diagnoser state-space. The main idea consists in (i) using binary decision diagrams (BDDs) to encode and handle the two sets of markings in each diagnoser node, and (ii) keeping an explicit (enumerative) encoding for the (observable) transitions that link the diagnoser nodes.

It should be noted that the necessary and sufficient condition for diagnosability is stated under the classic assumptions for fault diagnosis of PNs, namely:

- the PN is deadlock free, i.e., every reachable marking enables at least one transition to fire;
- the PN is bounded with an upper bound $b \in \mathbb{N}^*$, i.e., $\forall p \in P, M(p) \leq b$;
- the PN has no cycles composed exclusively of unobservable transitions;
- the faults are considered to be permanent, i.e., when a fault occurs the model remains infinitely faulty.

Theorem 1: A PN model is said to be diagnosable if and only if no F−indeterminate cycle exists in the diagnoser.[6]

The SSD approach is implemented in software tool developed in C++ and called DPN-SOG tool [47]. Using this tool, we have conducted several benchmark-based experimental comparisons using various existing DES diagnosis approaches [41], [44]–[46]. These experimentation have shown that the SSD approach shows a relatively high efficiency and scalability, which allows for dealing with real complex systems (see [44]). The results obtained from the above-mentioned works have motivated our choice of the SSD approach to conduct the fault diagnosis complex systems such as the PAS system investigated in this study.

IV. A PN BEHAVIORAL MODEL

In order to perform a fault diagnosis study on the PAS, we firstly need to establish (PN) behavioral models to describe the dynamics of the PAS subsystems. Such PN models will serve as a basis for the diagnosis analysis and shall satisfy some behavioral properties, (i.e., liveness, boundedness, extendability, reversibility, etc.).

The global model is built progressively by composing the subsystem models while integrating their mutual inter-dependencies. To describe the PN models, let us consider the following notations for all developed models:

- full black colored transitions indicate observable transitions, while the others indicate unobservable ones.
- blue colored places and transitions serve to implement the mutual inter-dependencies between the submodels,[7] namely, command and sensor information exchanged between the system components.

In the subsequent section, we discuss the PN models developed for the various subsystems. A description of the places and transitions meaning in the various models is given in Tables 3, 4, and 5.

A. THE MDCU MODEL

The model for the MDCU behavior is given in Figure 4. This model enables to generate the logical sequences of controlling commands, which are sent to the DCU. We consider that, initially, the train is stopped at a departure station.

[6] We should note that the correctness of SSD approach has been demonstrated in [46].

[7] For instance, blue place To_P_{40} output from T_4, in Figure 4, indicates the existence of an arc outgoes from transition T_4 to place P_{40} in Figure 6.

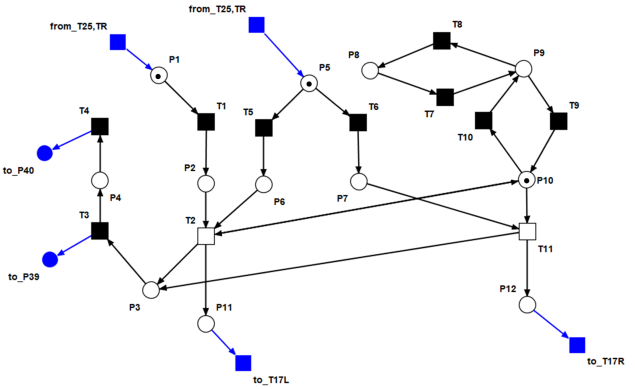

FIGURE 4. PN model for the Main Door Control Unit.

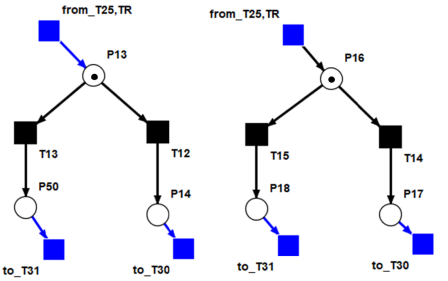

FIGURE 5. PN models for (a) the platform selection and (b) the gap bridge.

The logical controlling sequence is as follows: from the initial position (marked place P_1), the train driver orders the opening permission (T_1) to unlock the doors, on the left (resp. right) side of the train when P_6 (resp. P_7) is marked. This order can be sent to the DCU when the train speed is less than 0.5km/h (i.e., when P_{10} is marked). It should be noticed that the opening permission is sent to each door individually. To close all the doors, the train driver sends two signals: (i) the imminent closing signal (T_3) and then (ii) the general closing signal (T_4). The model gets back to its initial state once it receives the functional cycle ending signals (from transitions T_{25} or T_R).

The PN models for the station platform selection and the gap bridge deployment/cancellation are shown in Figures 5 (a) and (b), respectively. Before reaching the station platform, the balise unit indicates, to the MDCU, whether the station platform is *low* or *high*. This is depicted in the PN model by the XOR-split structure from marked place P_{13} and transitions T_{12} (for the low station platform) and T_{13} (for the high one). It is worth recalling that these information items are used by the DCU to decide whether or not to deploy the movable steps. The same PN structure is used to model the gap bridge deployment/cancellation, with marked place P_{16} and transitions T_{14} and T_{15}. All these information are then sent to the DCU through places P_{14}, P_{15}, P_{17} and P_{18}.

B. THE DCU MODEL

Figure 6 depicts the PN model for the train door opening/closing processes. The door opening can be launched only if the DCU receives the opening permission command from the MDCU (T_{17}) and provided that the passengers request to open the door by pressing the push-buttons in the inner and outer door faces (T_{16}). The door opening process is preceded by the deployment of movable steps in case a low station platform is selected (place P_{34} and transition T_{30} implement this requirement). In the case of a high station platform, the door opening begins without the MS deployment (the set of places/transitions $P_{36}, P_{37}, P_{38}, P_{41}, T_{31}$ and T_{32} implements such a requirement).

The behavior of the MS, in the case of a low station platform, is modeled by the place/transition cycle $P_{30}, T_{26}, P_{31}, T_{27}, P_{32}, T_{28}, P_{33}, T_{29}$, which starts with the MS deployment (represented by transition T_{26}) and terminates with the MS pending (represented by transition T_{29}).

The door closing process begins when the imminence closing command is received from the MDCU, which is represented by transition T_{21}. Sound and light warning signals (places P_{29} and P_{25} respectively) are firstly emitted to indicate the imminent closing of the doors. When the general closing command is received from the MDCU, which is represented by transition T_{22}, the doors move to close (P_{26}). In the case of a low station platform, the door closing is followed by the retraction of the MS (T_{28} and T_{29}). Transition T_{25} indicates the end of the operating cycle and the PN models reach their initial status.

It should be noted that the PN model in Figure 6 depicts the door opening/closing behavior for intermediate train vehicles, since they are equipped with movable steps. The PN model for doors in the extremity vehicles can be obtained by replacing the movable step process by the gap bridge one, since they have a similar operational cycle.

C. THE EMERGENCY PROCEDURE MODEL

Figure 7 depicts the PN model for the emergency procedure which is launched once a passenger pushes the EDR button (modeled by transition T_{33}), which, in turn, activates the emergency brake. It is worth noticing that once the emergency procedure is launched, a mode change is operated and the '*nominal*' mode is deactivated. The door opening can then operate when passengers request to open the door by pressing the push-button in the door (T_{16}), but only when the train speed is $\leq 1km/h$. Once the evacuation procedure is finished, the doors can be closed from the external door side (which is represented by transition T_{36}). As mentioned earlier, in the emergency case, the MS are not deployed in order to speed up the evacuation process.

The emergency procedure is always followed by the reset (re-initialization) process of the PAS components which brings the system to the initial state of its nominal behavior so as to be ready to start a new operation cycle. The set of places/transitions of the reset mechanism is depicted with green color in Figure 8.

The global PN model of the PAS nominal behavior is depicted in Figure 9. Colors are added to differentiate between the various modes. The normal behavior is illustrated with black color, while the emergency mode is with orange color. As shown before, the reset mechanism is

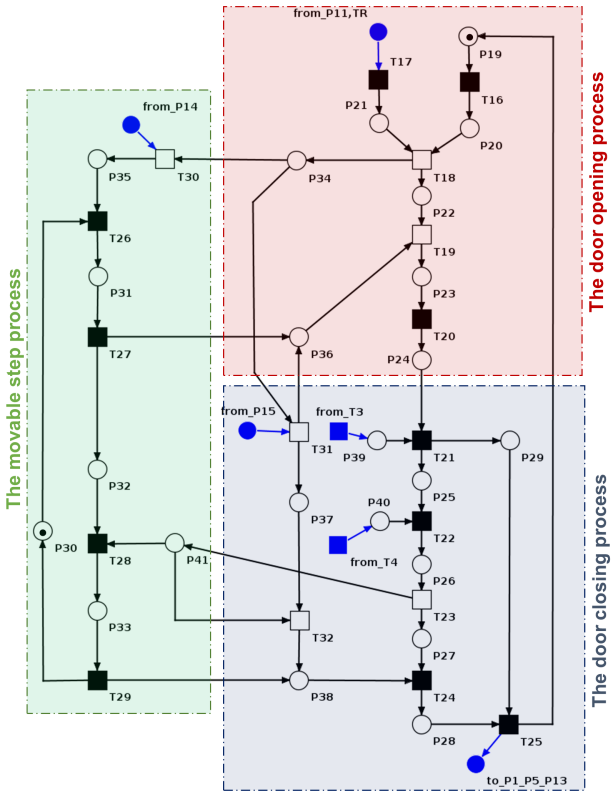

FIGURE 6. PN model for the door opening/closing processes.

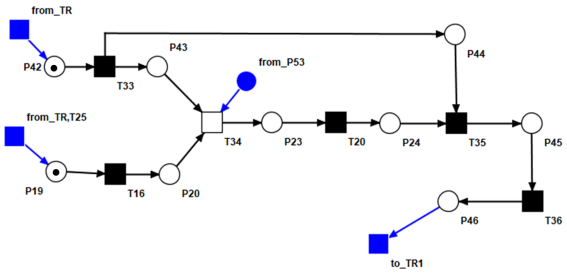

FIGURE 7. PN model for the PAS emergency process.

illustrated with green color and the mutual inter-dependencies places/transitions are in blue. It is worth noting that, for the sake of space and clarity, the PN model in Figure 9 depicts only one (left) side of the PN doors. The right side doors can be connected to the MDCU in the same way; such a connection can be implemented in the model through transition T_{11}. Moreover, the PN models for $n-$doors PAS can be obtained by duplicating n models of the train door (cf. figure 6) and connect each of them to the MDCU PN model (cf. Figure 4). Some inter-dependency place/transition should be added to ensure the synchronization between the doors according to the operational policy adopted for the system. For instance, we consider in our study that in an emergency case, once a passenger activates the EDR button on one side, all the doors on this side are unlocked, while on the doors on the other side remain locked. A different choice can be easily adopted.

D. MODELING FAULTY BEHAVIOR

The above description discusses the nominal behavior of the PAS. In the present study, we consider three types of failures that may affect the system behavior. We recall that these failures have been determined based on a preliminary risk analysis performed on the PAS.

1) OPENING DOOR ASYNCHRONIZATION (F1)

This fault type consists in an asynchronization between the door opening and the movable step deployment. In fact, in the nominal case, the MS deployment precedes the door opening. However, in some cases, the door may open simultaneously or before the MS has been completely deployed. Such a configuration can be due to transmission delays or malfunction of the MS actuator, and corresponds in Figure 10 (a) to the firing of faulty transition F_1. It is obvious that such a failure could cause danger situations related to the falling down of passengers.

2) CLOSING DOOR ASYNCHRONIZATION (F2)

This fault type consists in an asynchronization between the door closing and the movable step pending. In fact, it is similar to the situation above (F_1), where the MS begins pending before the entire closing of the door. Likewise, such a configuration can also be due to transmission delays or malfunction of the door actuators (electric motors), and corresponds in Figure 10 (b) to the firing of faulty transition F_2.

3) UNEXPECTED EMERGENCY DOOR OPENING (F3)

In the emergency procedure, the train speed must be less than $1km/h$ to allow the door opening. In some cases, the door may start opening before the train speed reaches this threshold. Such a configuration can be due to a (speed) sensing problem, or to some transmission delays. It is depicted in Figure 10 (c) by the firing of faulty transition $F3$.

V. PN MODEL ANALYSIS

In this section, we firstly analyze the PN models of the PAS established here-before, and then we discuss the diagnosability of the global model regarding the predetermined failures F_1, F_2, and F_3.

Firstly, it should be noted that the global PN model preserves some interesting properties that allow the model to correctly implement the operational logic of the system. Indeed, the global PN model is *bounded* , *live* , and *reversible* . These properties are checked by means of the TINA tool [48].

The state space of the PN models for $1-$, $2-$, and $4-$doors are enumerated in Table 1 to give a rough idea on the state explosion phenomenon. Both the cases of nominal (fault-free) behavior and the global behavior including failures are considered separately. One can observe that the size of the reachability graph (markings and arcs) grows quickly as the number of doors increases. Moreover, one can see

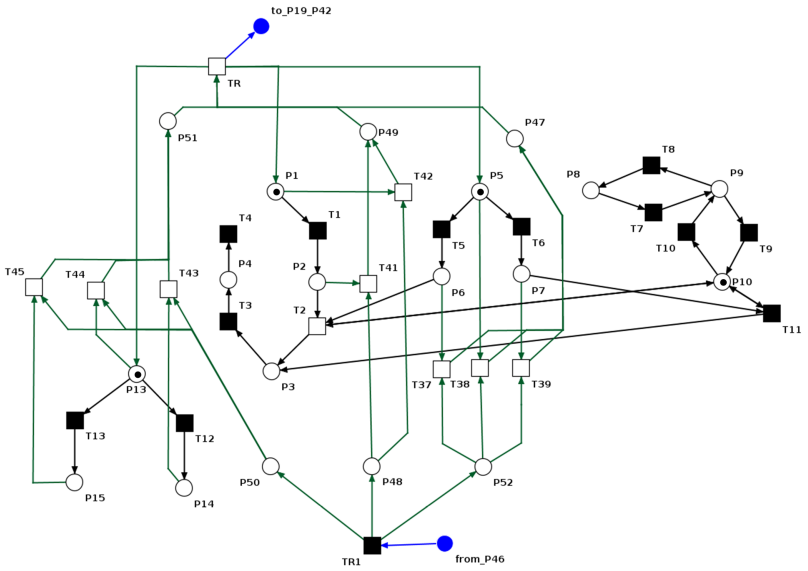

FIGURE 8. PN model for the reset mechanism.

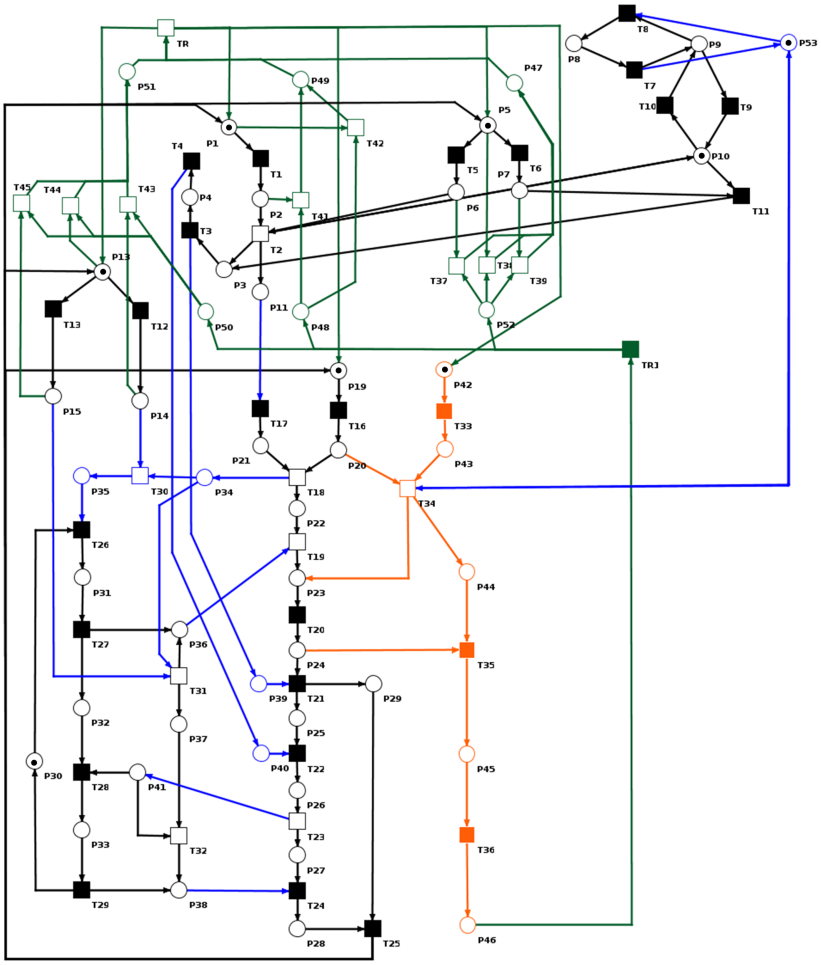

FIGURE 9. The global PN model of the PAS.

that the same number of markings is generated for both normal and global models, with respect to the number of doors, while the number or arcs is different. This is basically due to the fact that the failures mainly disturb the transition sequences without impacting the number of reachable markings.

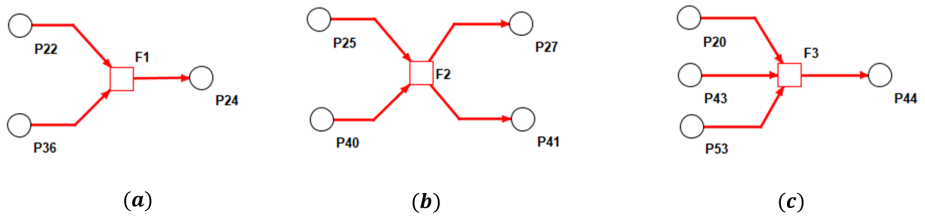

FIGURE 10. The faulty transition modeling.

TABLE 1. Features of the PAS models.

-	Doors	Places	Transi.	Markings	Arcs
Normal	1	47	40	600	2 336
	2	77	64	2 832	13 376
	4	141	104	88 704	518 448
Global	1	47	43	600	2 396
	2	77	64	2 832	13 376
	4	141	116	88 704	538 320

The diagnosability analysis of the developed model is performed using our DPN-SOG tool, on a Ubuntu PC (CPU:2.5GHz, RAM:16GB). The obtained results are summarized in Table 2. Thereafter, we highlight the main observations that can be derived from our analysis.

Firstly, the analysis made with the help of DPN-SOG tool shows that failures F_1 and F_2 are diagnosable, while failure F_3 is non diagnosable, with respect to the various numbers of doors. Therefore, both failures F_1 and F_2 are detectable upon a finite number of observable events, according to Theorem 1. On the contrary, for failure F_3, there is no finite delay that allows us to detect the occurrence of faults. Secondly, we can also observe that the state spaces of the generated diagnosers (State and Trans. columns in Table 2) grow considerably regarding the state spaces of the marking graphs in Table 5, which straightforwardly affects the elapsed time for analyzing diagnosability and the required memory for handling the diagnosers. Besides, DPN-SOG spends more time for analyzing the non-diagnosable models than the diagnosable ones, since the tool generates and analyzes all the critical scenarios which cause non-diagnosability. Such an output is very interesting as it helps determine the sensor map during the design phase and guides the reconfiguration and maintenance operations

After that, based on the DPN-SOG outputs, we discuss the scenarios (transition sequences in the PN models) which better explain the diagnosability verdict in a $1-$door PAS model. We notice that the same reasoning holds for the $n-$doors PAS models ($n > 1$).

- *For Failure F_1:* the system executes a normal behavior if in any transition sequence, the firing of transitions T_{16} and T_{17} is (not necessarily directly) succeeded by observable transition T_{20}. Otherwise, failure F_1 must have occurred, and transition T_{21} succeeds transitions T_{16} and T_{17}, without any firing of T_{20}. It should be noted that the PAS model is diagnosable regarding failure F_1 thanks to the sensors that allow the capturing of the occurrence of T_{20} and T_{21} corresponding events.
- *For Failure F_2:* the MS starts retracting before the closing command is received from the MDCU (transition T_{22}), i.e., before the door is completely closed. In such a scenario, failure F_2 is immediately detected since the observable command corresponding to T_{22} has not appeared in the logic operating sequence, i.e., observable transition T_{21} is preceded by observable transition T_{24}.
- *For Failure F_3:* this failure is related to the emergency procedure. In fact, F_3 may or may not occur right after the activation of the emergency process, i.e., the firing of transitions T_{33} and T_{16}. For both cases and during the entire emergency procedure, the same ensuing firing sequence $T_{35}, T_6, TR1, \ldots TR$ occurs and leads the model into an ambiguous status. Indeed, from a diagnosis point of view, such a critical scenario corresponds to an indeterminate cycle in the diagnoser, which makes the model non-diagnosable and, thus, F_3 non-detectable.

It is worth noting that for diagnosble failures F_1 and F_2, the diagnosers generated using DPN-SOG tool can be used to perform the online diagnosis tasks according to the technique proposed in [40]. In fact, the role of the diagnoser (as a deterministic automaton) is to carry out the state estimation of the PAS system online and emit verdicts regarding its behavior (normal or faulty) based on the observations captured online. The diagnoser is generally implemented as a dedicated module as part of the onboard train supervision and control devices. It can be implemented using Oriented Objects languages (Java, C++, etc.) [49] or Hardware Language (VHDL, Verilog) [50]. To perform the online diagnosis, the diagnoser module takes as input the signals gathered from the sensors and the commands issued from the door control unit, and determines the current state of the targeted system and the verdict regarding its behavior (i.e., if a fault has occurred or not and if so which one).

Regarding the non-diagnosable failure F_3, the corresponding diagnoser cannot be used for the online diagnosis, since it would fail to detect the fault occurrences with certainty. In this case, the set of deployed sensors needs to be endowed by adding new sensors in order to make some non-observable transitions observable (or also by modifying the sensor map).

TABLE 2. Diagnosability analysis for *n*−doors PAS.

–	Faults	States	Trans.	Verdict	Time (s)	Memory (kb)	Critical scenarios
1-door	F_1	1 875	10 381	yes	1	53 278	0
	F_2	1 875	10 385	yes	1	53 206	0
	F_3	1 757	9 114	non	7	84 201	3 468
2-doors	F_1	10 256	73 168	yes	9	1 012 274	0
	F_2	10 256	73 184	yes	8	1 011 360	0
	F_3	9 265	61 532	non	554	1 565 897	24 679
4-doors	F_1	157 176	1 289 040	yes	3 530	47 649 731	0
	F_2	157 176	1 290 670	yes	3518	47 208 460	0
	F_3	144 000	1 135 140	no	$> 24\,h$	81 033 798	209 807

TABLE 3. Meanings of places and transitions of the PN models (1).

-	P / T	Meaning
the MDCU behavior	P_1	MDCU's initial state
	P_2	memorizing the opening permission
	P_3	the process of opening the door is launched
	P_4	waiting the general closing command
	P_5	there is no selection of the platform side
	P_6	the platform is in the train left side
	P_7	the platform is in the train right side
	P_8	the train speed: $v > 1km/h$
	P_9	the train speed: $1km/h \geq v > 0,5km/h$
	P_{10}	the train speed: $v \leq 0,5km/h$
	P_{11}	the command to open the left doors is sent
	P_{12}	the command to open the right doors
	T_1	send the opening permission command (OP)
	T_2	send the signal permission to open the left doors
	T_3	send the imminent closing command (IC)
	T_4	send the general closing command (GC)
	T_5	detection of the left platform
	T_6	detection of the right platform
	T_7	detection of the train speed: $v \geq 1km/h$
	T_8	detection of the train speed: $v \leq 1km/h$
	T_9	detection of the train speed: $v \leq 0,5km/h$
	T_{10}	detection of the train speed: $v \geq 0,5km/h$
	T_{11}	send the signal permission to open the right doors
Platform	P_{13}	the is no platform type detected
	P_{14}	a low platform is detected
	P_{15}	a high platform is detected
	T_{12}	detection of a low platform
	T_{13}	detection of a high platform
Gap Bridge	P_{16}	there is no information regarding the GB
	P_{17}	the GB is authorized
	P_{18}	the GB is canceled
	T_{14}	send the GB authorization
	T_{15}	send the GB cancellation

TABLE 4. Meanings of places and transitions of the PN models (2).

-	P / T	Meaning
Door opening process	P_{19}	there is no request to open the door (PR)
	P_{20}	the PR request is memorized
	P_{21}	the OP command (from the MDCU) is memorized
	P_{22}	the MS/GB are processing
	P_{23}	the door is opening
	P_{24}	the door is completely open
	T_{16}	the passenger request (PR) is activated
	T_{17}	receive the opening permission (OP) command
	T_{18}	send the ordinary signal to open the MS or GB
	T_{19}	start the door opening operation
	T_{20}	end of the door opening operation
Door closing process	P_{25}	a light signal indicates the imminence door closing
	P_{26}	the door is closing
	P_{27}	the door is closed
	P_{28}	the PR waits for the MS/BG to be closed
	P_{29}	emission of a sound warning signal
	T_{21}	reception of the imminence closing command (IC)
	T_{22}	reception of the general closing command (GC)
	T_{23}	send closing the MS/GB command
	T_{24}	MS/GB is closed
	T_{25}	the door closing process is finished
Movable step process	P_{30}	the MS is in closed position
	P_{31}	the MS is deploying
	P_{32}	the MS is completely deployed
	P_{33}	the MS is closing
	P_{34}	waiting the platform selection signal
	P_{35}	waiting the MS deployment to begin
	P_{36}	the MS is deployed or the platform is low
	P_{38}	the MS is closed or the platform is low
	P_{39}	the imminence closing signal is received
	P_{40}	the general closing signal is received
	P_{41}	waiting the MS closing to start
	T_{26}	start the MS deployment
	T_{27}	send the signal indicating the MS deployment
	T_{28}	reception of the signal to close the MS
	T_{29}	send the signal indicating that the MS is closed
	T_{30}	send a signal to start the MS deployment
	T_{31}	receive the information for low platform

Such an operation is known in the fault diagnosis field as the sensor placement's / sensor optimization [51]–[54]. For instance, a preliminary analysis of the critical scenarios generated in the case of failure F_3 shows that it is sufficient to turn transition T_{34} to become observable, by adding a dedicated sensor device, to ensure the diagnosability of failure F_3. In fact, the determination of the unobservable transitions to be turned observable is often a fastidious task and cannot be done by a manual analysis of the critical scenarios. Besides, developing efficient algorithms to deal with critical scenarios is an interesting and emerging research topic in the DES community [29].

Finally, it should be noticed that, in practice, the developed technique can be used to assess the design choices made and shall be used from as sonna s the design phases. Indeed, analyzing diagnosability gives valuable information to guide sensors' placement.

TABLE 5. Meanings of places and transitions of the PN models (3).

-	P / T	Meaning
Emergency procedure	P_{42}	there is no emergency door request (EDR)
	P_{43}	the EDR request is memorized
	P_{19}	there is no request to open the door (PR)
	P_{20}	the PR request is memorized
	P_{23}	the door is opening
	P_{24}	the door is completely open
	P_{45}	passengers evacuation
	P_{46}	Waiting the reset command
	T_{33}	the emergency door release is activated
	T_{16}	the passenger request (PR) is activated
	T_{34}	start the door opening
	T_{20}	end of the door opening operation
	T_{35}	start the evacuation process
	T_{36}	the emergency procedure is ended

VI. CONCLUSION

In this paper, we have investigated the fault diagnosis of the passenger access system, which is a safety-critical main module onboard passenger trains. The system behavior is abstracted as a DES and modeled using PN formalism so as to apply DES fault detection and identification techniques. Firstly, the PN behavioral models for the PAS components, including their normal and faulty modes, are developed and then integrated to obtain a global behavioral model. The diagnosability analysis of the obtained model is then carried out using a diagnoser-based approach, with the help of DPN-SOG tool.

Our future research in this topic will be twofold: regarding the system modeling, we plan to endow our PN models with timed information and use some extensions of PN formalism, namely timed/time PNs; thus, further diagnosis techniques can be brought into play to enhance the diagnosis process. Regarding the diagnosis process, we wish to investigate some issues related to sensor placement/optimization, since this is an essential task in the system design cycle to enhance its reliability and optimize the system engineering as a whole.

APPENDIX
PLACES AND TRANSITIONS MEANINGS

See Tables 3–5.

REFERENCES

[1] J. C. Knight, "Safety critical systems: Challenges and directions," in *Proc. 24th Int. Conf. Softw. Eng. (ICSE)*, 2002, pp. 547–550.

[2] T. Lecomte, T. Servat, and G. Pouzancre, "Formal methods in safety-critical railway systems," in *Proc. 10th Brasilian Symp. Formal Methods*, 2007, pp. 29–31.

[3] J. Bowen and V. Stavridou, "Safety-critical systems, formal methods and standards," *Softw. Eng. J.*, vol. 8, no. 4, pp. 189–209, Jul. 1993.

[4] *Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems*, document IEC 61508-3, European Committee for Electrotechnical Standardization, Brussels, Belgium, 2000.

[5] M. S. Durmuş, S. Takai, and M. T. Söylemez, "Fault diagnosis in fixed-block railway signaling systems: A discrete event systems approach," *IEEJ Trans. Elect. Electron. Eng.*, vol. 9, no. 5, pp. 523–531, 2014.

[6] Q. Liu, H. Zhang, J. Wan, and X. Chen, "An access control model for resource sharing based on the role-based access control intended for multi-domain manufacturing Internet of Things," *IEEE Access*, vol. 5, pp. 7001–7011, 2017.

[7] J. Wan, B. Yin, D. Li, A. Celesti, F. Tao, and Q. Hua, "An ontology-based resource reconfiguration method for manufacturing cyber-physical systems," *IEEE/ASME Trans. Mechatronics*, doi: 10.1109/TMECH.2018.2814784.

[8] Q. An *et al.*, "Fault decoupling method of railway door system based on the extended Petri net," in *Proc. Int. Conf. Elect. Inf. Technol. Rail Transp.* Berlin, Germany: Springer, 2016, pp. 155–165.

[9] Y. Fu, L. Jia, Y. Qin, and X. Cheng, "Rail train door system hidden danger identification based on extended time and probability Petri net," in *Proc. Int. Conf. Elect. Inf. Technol. Rail Transp.* Berlin, Germany: Springer, 2016, pp. 319–328.

[10] F. Turgis, R. Copin, P. Loslever, L. Cauffriez, and N. Caouder, "Design of a testing bench for simulating tightened-up operating conditions of train's passenger access," in *Proc. ESREL*, 2009, pp. 2279–2284.

[11] N. Lehrasab, H. P. B. Dassanayake, C. Roberts, S. Fararooy, and C. J. Goodman, "Industrial fault diagnosis: Pneumatic train door case study," *Proc. Inst. Mech. Eng. F, J. Rail Rapid Transit*, vol. 216, no. 3, pp. 175–183, 2002.

[12] Eao Ag. *Humain Machine Interface Systems for Passenger Access in Rail Applications*. [Online]. Available: www.railway-technology.com

[13] C. Li, S. Luo, C. Cole, and M. Spiryagin, "An overview: Modern techniques for railway vehicle on-board health monitoring systems," *Vehicle Syst. Dyn.*, vol. 55, no. 7, pp. 1045–1070, 2017.

[14] T. Khaoula, C. Nizar, V. Sylvain, and T. Teodor, "Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges," *Ann. Rev. Control*, vol. 42, pp. 63–81, Dec. 2016.

[15] P. Pereira, R. P. Ribeiro, and J. Gama, "Failure prediction—An application in the railway industry," in *Proc. Int. Conf. Discovery Sci.* Cham, Switzerland: Springer, 2014, pp. 264–275.

[16] H. Dassanayake, C. Roberts, C. J. Goodman, and A. M. Tobias, "Use of parameter estimation for the detection and diagnosis of faults on electric train door systems," *Proc. Inst. Mech. Eng. O, J. Risk Rel.*, vol. 223, no. 4, pp. 271–278, 2009.

[17] L. Shuai, J. Limin, Q. Yong, Y. Bo, and W. Yanhui, "Research on urban rail train passenger door system fault diagnosis using PCA and rough set," *Open Mech. Eng. J.*, vol. 8, pp. 340–348, 2014.

[18] J. Yu, "Local and global principal component analysis for process monitoring," *Process Control*, vol. 22, no. 7, pp. 1358–1373, 2012.

[19] E. Miguelanez, K. E. Brown, R. Lewis, C. Roberts, and D. M. Lane, "Fault diagnosis of a train door system based on semantic knowledge representation," in *Proc. 4th IET Int. Conf. Railway Condition Monit.*, 2008, pp. 1–6.

[20] Y. Han *et al.*, "Online predictive diagnosis of electrical train door systems," in *Proc. 10th World Congr. Railway Res. (WCRR)*, 2013, p. 6.

[21] L. Cauffriez, S. Grondel, P. Loslever, and C. Aubrun, "Bond graph modeling for fault detection and isolation of a train door mechatronic system," *Control Eng. Pract.*, vol. 49, pp. 212–224, Apr. 2016.

[22] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer, 2009.

[23] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dyn. Syst.*, vol. 4, no. 2, pp. 197–212, 1994.

[24] A. I. Potekhin, S. A. Branishtov, and S. K. Kuznetsov, "Discrete-event models of a railway network," *Automat. Remote Control*, vol. 77, no. 2, pp. 344–355, 2016.

[25] G. Cavone, M. Dotoli, and C. Seatzu, "A survey on Petri net models for freight logistics and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1795–1813, Jun. 2018.

[26] M. S. Durmus, "Control and fault diagnosis of railway signaling systems: A discrete event systems approach," Ph.D. dissertation, Graduate School Eng., Osaka Univ., Suita, Japan, 2015.

[27] A. Giua and C. Seatzu, "Modeling and supervisory control of railway networks using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 3, pp. 431–445, Jul. 2008.

[28] F. Basile, "Overview of fault diagnosis methods based on Petri net models," in *Proc. IEEE Eur. Control Conf.*, Jun. 2014, pp. 2636–2642.

[29] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annu. Rev. Control*, vol. 37, no. 2, pp. 308–320, 2013.

[30] A. Boussif and M. Ghazel, "A diagnosis study on a train passenger access system using Petri net models," in *Proc. 15th IFAC Symp. Control Transp. Syst. (CTS)*, 2018, pp. 1–6.

[31] F. Jovicic, "Guideline for the application of harmonised design targets (CSM-DT) for technical systems as defined in (EU) regulation 2015/1136 within the risk assessment process of regulation 402/2013," in *Proc. Eur. Union Agency Railways*, 2016, pp. 1–139.

[32] APTA PRESS Task Force, *Standard for Door Systems for New and Rebuilt Passenger Cars*, document PR-CS-S-012-02, APTA Commuter Rail Executive Committee, 2002.
[33] L. Cauffriez, P. Loslever, N. Caouder, F. Turgis, and R. Copin, "Robustness study and reliability growth based on exploratory design of experiments and statistical analysis: A case study using a train door test bench," *Int. J. Adv. Manuf. Technol.*, vol. 66, nos. 1–4, pp. 27–44, 2013.
[34] APTA PRESS Task Force, *Standard for Emergency Evacuation Units for Passenger Rail Cars*, document APTA PR-PS-S-003-98, 1998.
[35] S. H. Markos *et al.*, "Passenger train emergency systems: Review of egress variables and egress simulation models," U.S. Dept. Transp., Federal Railroad Admin., Office Res. Develop., Washington, DC, USA, Final Rep., 2013.
[36] *Train Door Emergency Egress and Access and Emerge Evacuation Procedures*, document 02468, Independent Transport Safety Reliability Regulator, Transport Safety Regulation Division, 2004.
[37] T. K. Park, K. W. Park, and K. Uematsu, "The assessment to achieve safety of the train door system in IEC 61058 (The Case of Busan-Gimhae Light Papid Train)," *Scientiae Mathematicae Japonicae*, vol. 75, pp. 255–266, 2012.
[38] X. Cheng, Z. Xing, Y. Qin, Y. Zhang, S. Pang, and J. Xia, "Reliability analysis of metro door system based on FMECA," *J. Intell. Learn. Syst. Appl.*, vol. 5, no. 4, p. 216, 2013.
[39] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
[40] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.
[41] B. Liu, M. Ghazel, and A. Toguyéni, "Model-based diagnosis of multi-track level crossing plants," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 546–556, Feb. 2016.
[42] T. Ushio, I. Onishi, and K. Okuda, "Fault detection based on Petri net models with faulty behaviors," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 1, Oct. 1998, pp. 113–118.
[43] L. Yin, Z. Li, N. Wu, S. Wang, and T. Qu, "Fault diagnosis in partially observed Petri nets using redundancies," *IEEE Access*, vol. 6, pp. 7541–7556, 2018.
[44] A. Boussif, B. Liu, and M. Ghazel, "An experimental comparison of three diagnosis techniques for discrete event systems," in *Proc. Int. Workshop Principles Diagnosis*, 2017, pp. 1–7.
[45] A. Boussif, M. Ghazel, and K. Klai, "Combining enumerative and symbolic techniques for diagnosis of discrete-event systems," in *Proc. 9th Int. Workshop Verification Eval. Comput. Commun. Syst.*, 2015, pp. 23–34.
[46] A. Boussif, "Contributions to fault diagnosis of discrete-event systems," Ph.D. dissertation, Univ. Lille, Lille, France, 2016, vol. 1.
[47] A. Boussif, M. Ghazel, and K. Klai, "DPN-SOG: A software tool for fault diagnosis of labeled Petri nets using the semi-symbolic diagnoser," in *Proc. 11th Colloque Modélisation Systèmes Réactifs (MSR)*, 2017, pp. 1–7.
[48] B. Berthomieu, P.-O. Ribet, and F. Vernadat, "The tool TINA—Construction of abstract state spaces for Petri nets and time Petri nets," *Int. J. Prod. Res.*, vol. 42, no. 14, pp. 2741–2756, 2004.
[49] J. Van Gurp and J. Bosch, "On the implementation of finite state machines," Licentiate thesis, Dept. Softw. Eng. Comput. Sci., Univ. Groningen, Groningen, The Netherlands, 2000, p. 45.
[50] S. Golson, "State machine design techniques for Verilog and VHDL," *Synop. J. High-Level Des.*, vol. 9, nos. 1–48, p. 12, 1994.
[51] F. Cassez and S. Tripakis, "Fault diagnosis with static and dynamic observers," *Fundamenta Informaticae*, vol. 88, no. 4, pp. 497–540, 2008.
[52] M. Krysander and E. Frisk, "Sensor placement for fault diagnosis," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1398–1410, Nov. 2008.
[53] R. Debouk, S. Lafortune, and D. Teneketzis, "On an optimization problem in sensor selection," *Discrete Event Dyn. Syst.*, vol. 12, no. 4, pp. 417–445, 2002.
[54] M. P. Cabasino, S. Lafortune, and C. Seatzu, "Optimal sensor selection for ensuring diagnosability in labeled bounded Petri nets," in *Proc. 11th IFAC Workshop Discrete Event Syst.*, 2012, pp. 1–6.

ABDERRAOUF BOUSSIF received the B.Eng. degree in control and computer engineering from the Ecole Natioanle Polytechnique, Algiers, in 2012, the master's degree in control and computer engineering from the École Normale Supérieure de Cachan, Paris, in 2013, and the Ph.D. degree in control and computer engineering from the University of Lille, Lille, in 2016. He is currently a Post-Doctoral Researcher with the Metz Engineering School, École Nationale Supérieure D'arts et Métiers. His research interests are mainly in formal methods, model-based safety analysis, and fault diagnosis of safety-critical systems.

MOHAMED GHAZEL received the master's degree in automatic control and industrial computer sciences from the École Centrale de Lille in 2002, the Ph.D. degree in automatic control and industrial computer sciences from the University of Lille in 2005, and the Habilitation à Diriger des Recherches degree from the University Lille Nord de France in 2014. He is currently the Research Director of the COSYS/ESTAS Team, The French Institute of Science and Technology for Transport, Development and Networks. His research mainly focuses on safety and interoperability of transportation systems using discrete models. He is a member of the IFAC TC 7.4 on Transportation Systems. He is involved in several national and European research projects and acts as an Expert for the European Commission in the framework of innovation programs.