HAL
open science

Health Monitoring of a Planetary Rover Using Hybrid Particle Petri Nets

Quentin Gaudel, Pauline Ribot, Elodie Chanthery, Matthew J. Daigle

To cite this version:

Quentin Gaudel, Pauline Ribot, Elodie Chanthery, Matthew J. Daigle. Health Monitoring of a Planetary Rover Using Hybrid Particle Petri Nets. International Conference on Applications and Theory of Petri Nets and Concurrency, Jun 2016, Toruń, Poland. pp.196–215, 10.1007/978-3-319-39086-4_13 . hal-01352011

HAL Id: hal-01352011

https://hal.science/hal-01352011v1

Submitted on 5 Aug 2016

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

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

Health Monitoring of a Planetary Rover Using Hybrid Particle Petri Nets

Quentin Gaudel[1], Pauline Ribot[1], Elodie Chanthery[1], and Matthew J. Daigle[2]

[1] LAAS-CNRS, Université de Toulouse, CNRS, INSA, UPS, Toulouse, France
{quentin.gaudel, pauline.ribot, elodie.chanthery}@laas.fr
[2] NASA Ames Research Center, Moffett Field, CA, 94035, USA
matthew.j.daigle@nasa.gov

Abstract. This paper focuses on the application of a Petri Net-based diagnosis method on a planetary rover prototype. The diagnosis is performed by using a model-based method in the context of health management of hybrid systems. In system health management, the diagnosis task aims at determining the current health state of a system and the fault occurrences that lead to this state. The Hybrid Particle Petri Nets (HPPN) formalism is used to model hybrid systems behavior and degradation, and to define the generation of diagnosers to monitor the health states of such systems under uncertainty. At any time, the HPPN-based diagnoser provides the current diagnosis represented by a distribution of beliefs over the health states. The health monitoring methodology is demonstrated on the K11 rover. A hybrid model of the K11 is proposed and experimental results show that the approach is robust to real system data and constraints.

Keywords: Diagnosis · Hybrid systems · Model-based monitoring · Health management · Uncertainty · Petri Nets · Particle filter

1 Introduction

Real systems have become so complex that it is often impossible for humans to capture and explain their behaviors as a whole, especially when they are exposed to failures. *System health management* or *prognostics and health management* (PHM) aims at developing tools that can support operator tasks, reducing the global costs due to unavailability and repair actions, but also optimizing the mission reward by replanning or reconfiguring the system [23].

An efficient health monitoring technique has to be adopted to determine the health state of the system at any time by using diagnostics and prognostics techniques. A diagnosis method is used to determine the current health state and identify the possible causes of failures that lead to this state by reasoning on observations. Prognosis is used to predict the future health states and the dates of the occurrences of the faults that lead to these states.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-39086-4_13

A system is considered as a *hybrid system* if it exhibits both *discrete and continuous dynamics* [13]. Sensor data and commands are designated as continuous or discrete observations on the system. Hybrid systems are usually described as a multi-mode system composed of an underlying discrete-event system (DES) representing the mode changes and various underlying continuous dynamics associated with each mode [3]. A discrete state of the DES coupled to a continuous evolution (continuous dynamics) represents a mode (or operational condition) of the system. The changes of modes are then associated with occurrences of *events*. The system *discrete state* is the current discrete state of the DES. The evolution of the system *continuous state* depends on continuous dynamics associated with the current system mode.

In most industrial systems, if the degradation is not observable, it is estimated as fault occurrence probabilities. The degradation thus depends on the stress level of the current health mode of the system and, in some cases, also relies on the current continuous state and also on the analysis of the events that occurred on the system [11]. Because of these dependencies, we consider the degradation as a hybrid characteristic. We thus defined the evolution of this hybrid characteristic as *hybrid dynamics* and its current value as the *hybrid state*. We extend the multi-mode system by associating underlying hybrid dynamics (e.g. degradation laws) with each mode. The definition of a *mode* is thus enriched and is a combination of a discrete state of the DES with continuous dynamics and hybrid dynamics [9]. The *state* of the hybrid system is the combination of its discrete, continuous and hybrid states.

Our previous works introduced a framework called *Hybrid Particle Petri Nets* (HPPN). [10] proposed to use HPPN to both model the system, which is hybrid but also uncertain, and track its current health state with a diagnoser representation. The methodology uses information about the system degradation that is a significant advantage to compute a more accurate diagnosis and to perform prognosis. In [11], we tested the proposed approach on a simulated three-tank system.

The main contribution of this paper is to expose results of the implemented HPPN-based health monitoring method on the K11 planetary rover prototype. The K11 is a testbed developed by NASA Ames Research Center and is used for diagnostics and prognostics purposes [1, 7, 8, 23]. A hybrid model of the rover is proposed, based on the discretization of its health evolution. Experimental results are given, illustrating how the methodology is robust to real system data and constraints. The method exposed in [11] have been improved. It is hence recalled and new notions are precised, such as the definition of events, the mode scores or the scale parameters for example.

This paper is organized as follows. Section 2 presents related works on diagnosis of hybrid systems. Section 3 recalls and deepens the health monitoring methodology based on the modeling of the system and the generation of a diagnoser by using HPPN. Section 4 focuses on the application of the proposed methodology on the K11 planetary rover prototype. It provides the K11 hybrid model and exposes the experimental results and performance metrics. Conclusions and future works are discussed in the final section.

2 Related Works

In [5], we extended the diagnosis approach proposed in [3] in order to integrate diagnosis and prognosis for hybrid systems. The approach uses hybrid automata and stochastic models for the system degradation. Diagnosis is performed using a Discrete Event System (DES) approach. The DES-oriented diagnosis framework, however, explodes in the number of states and it does not seem to be the most suited for the incorporation of the prognosis task. Prognosis is indeed a probabilistic prediction process and is highly subject to uncertainty. The health monitoring task usually has to take into account the different sources of uncertainty, such as model approximation, partial observability of the system and measurement noise. Diagnosis should help the decision making process. In case of ambiguity in diagnosis results, the traditional diagnoser fails at providing relevant information. By taking all uncertainty sources into account, the method we propose succeeds in quantifying each diagnosis result.

The diagnoser approach was introduced in [21]. The diagnoser is basically a monitor that is able to process any possible observable event that occurs in the system. It consists in recording these observations and providing the set of possible faults whose occurrence is consistent with the observations. However, this approach is restricted to DES and does not manage uncertainty. Some approaches extend the diagnoser to DES modelled by Petri nets. A distributed version of the diagnoser is proposed in [12]. In [4], the authors study the diagnosability of a system, inspired by the diagnosability approach for finite state automata proposed by [21]. However, none of these approaches take into account continuous aspects, nor consider uncertainty in the system. In [22], an approach for the localization of intermittent faults by dealing with partial observability in the discrete event framework is proposed. The method is based on Petri nets that model the normal functioning of the system observable behavior. A localization mechanism, based on the diagnoser approach, points out the set of events potentially responsible for the faults.

Some works try to take into account uncertainty. In [15], a particle filtering technique is used to estimate the state of a hybrid system modeled as a hybrid automaton. Uncertainty related to discrete events is not taken into account and the system degradation is not considered. The authors of [20] use partially observed Petri nets. Partially observed Petri nets are transformed into an equivalent labelled Petri net and an online monitor is built to diagnose faults and provide beliefs (degrees of confidence) regarding the occurrences of faults. However, this approach is limited because it only takes into account uncertainty in the diagnosis results, not about the model or the event observations. In [2], the authors propose to reduce the explosion of the state space by introducing generalized markings (negative tokens) to take into account uncertainty about the firing of transitions. The stochastic Petri nets are used in [14] to build a formal model of each component of an integrated modular avionics architecture. However, for all these approaches, no continuous aspect in the model is taken into account.

In [24], the Modified Particle Petri Nets (MPPN) formalism is used to get a more compact representation and to capture all uncertainties related to the system, the observations and the diagnosis results. MPPN are an extension of particle Petri nets [17] that combine a discrete event model (Petri net) with a continuous model (differential equations). The main advantage of MPPN is that uncertainties about both discrete and continuous dynamics are taken into account. A particle filter is used to integrate probabilities in the continuous state estimation process. Tokens are duplicated during the online process to model uncertainty on the event occurrences. The duplication, however, disturbs the distribution over the continuous state. In addition, there is no mention of the health state notion for the system. In [9], we apply the MPPN formalism to health monitoring and highlight the inability to capture hybrid characteristics. In [10], we extend MPPN into HPPN in order to monitor hybrid characteristics and solve the continuous distribution issue. HPPN are used to monitor a three-tank system, for which system degradation evolves according to the valves configurations.

This paper focuses on the application of the health monitoring methodology on the K11 rover, that is subject to the inherent uncertainty of real systems. In previous works, health monitoring and diagnosis was applied to the K11 rover. In [18], two diagnosis algorithms were applied, Qualitative Event-based Diagnosis (QED) [6], and the Hybrid Diagnosis Engine (HyDE) [19]. QED performs diagnosis based on reasoning over symbols representing qualitative deviations of the sensor signals with respect to model-predicted values. Sensor and process noise are handled by using an observer to estimate the current system state, however no uncertainty in the symbols computed for diagnosis is considered, and all diagnostic hypotheses are viewed as equally likely. HyDE is a consistency-based diagnosis engine that uses hybrid and stochastic models and reasoning. Reasoning is performed by hypothesizing alternative system trajectories inferred from the transition and behavior models of the system, and considers a priori fault probabilities and mode transition probabilities. Both diagnosis algorithms were used to diagnose parasitic load, motor friction, and voltage sensor faults in simulation. In [23], QED diagnosed parasitic load faults and voltage sensor faults in real-world scenarios.

3 Hybrid System Health Monitoring

This section recalls the methodology proposed in [11] to perform model-based health monitoring of hybrid systems.

We are interested in modeling changes in system dynamics when one or several anticipated faults occur. The *health modes* are the hybrid system modes and represent different health conditions. As long as the system does not encounter any fault, it is in a *nominal mode*. Tracked faults are assumed to be permanent, i.e. once a fault happens, the system moves from a nominal mode to a *degraded mode* or faulty mode. Without repair, the system ends in a *failure mode* in which it is not operational anymore.

The proposed diagnosis solution is a two-step method. The first offline step is to model the considered system using the HPPN framework (see Sect. 3.1) and to generate the HPPN-based diagnoser (see Sect. 3.2). Then the online process initializes the diagnoser marking and uses consecutive observations to update it and compute the diagnosis at any time (see Sect. 3.2).

Example 1. Throughout Sect. 3, an example of a mobile robot, described in Fig. 1, is used to illustrate the definitions and concepts.

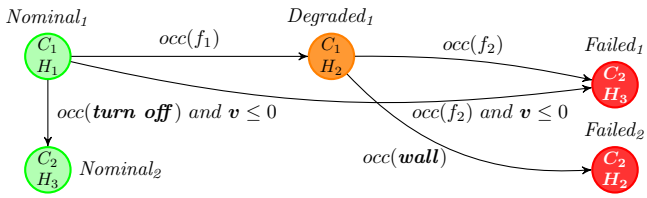

Fig. 1. Mobile robot description.

The system is described with an oriented graph, in which the nodes represent the health modes and the arcs represent the mode changes. Variables that can be observed or estimated with observations are in bold.

The robot mission is to move without encountering an obstacle or failure, until it reaches a specific area and is turned off. The initial mode is $Nominal_1$: the robot is not degraded and is moving in a non-hostile zone. Its velocity v can be estimated with continuous dynamics C_1 and continuous observations, and is positive. Two faults are expected and the robot degradation is estimated as fault occurrence probabilities with hybrid dynamics H_1, in which the probabilities increase with time.

When the (discrete and observable) on-off command *turn off* occurs, the robot stops and its velocity decreasing to 0. The robot enters in mode $Nominal_2$, where its motor is turned off and its velocity thus stays 0 (continuous dynamics C_2). Because the robot is turned off, the fault occurrence probabilities stagnate, following hybrid dynamics H_3.

Fault f_2 represents the disconnection of the robot motor. Its occurrence leads the system to the failure mode $Failed_1$. The occurrence of f_2 implies the robot stops, so its velocity decreases to 0. Once the motor is disconnected, the robot has the same continuous and hybrid dynamics (C_2 and H_3) as if it was turned off.

Fault f_1 represents the entrance in a hostile zone and in mode $Degraded_1$. The robot is still moving at the same velocity (C_1). The physical conditions, however, imply that the probability of occurrence of f_2 increases more significantly than in mode $Nominal_1$. This is defined with hybrid dynamics H_2.

From mode $Degraded_1$, the robot can still enter in mode $Failed_1$ with fault f_2 occurrence but it does not match with any condition on the velocity in that case

(see arc between *Degraded₁* and *Failed₁*). The velocity estimation is considered less accurate in the hostile zone than in the non-hostile zone, indeed.

Finally, the hostile zone contains obstacles. The robot can encounter a wall, that stops the robot but not its motor. In that case, the mission fails and the robot enters in failure mode *Failed₂*. This event *wall* is not predictable (not estimated with probabilities) but is observable with an environmental on-off sensor. Even if the mission is compromised and the robot is not moving anymore (C_2), its motor is still on so the degradation laws remain the same (H_2).

3.1 Hybrid System Modeling

We propose to model the system by using the *Hybrid Particle Petri Nets* (HPPN) formalism, introduced in [10].

Hybrid Particle Petri Nets. The HPPN formalism is an extension of Petri nets.

A HPPN is defined as a tuple $\langle P, T, A, \mathcal{A}, E, X, H, C, \mathcal{F}, \Omega, M_0 \rangle$ where:

- P is the set of places, partitioned into numerical places P^N, symbolic places P^S and hybrid places P^H,
- T is the set of transitions,
- $A \subset P \times T \cup T \times P$ is the set of arcs,
- $\mathcal{A}$ is the set of arc annotations,
- E is the set of event labels,
- $X \subset \mathbb{R}^n$ is the state space of the continuous state vector, with $n \in \mathbb{N}$ the number of continuous state variables,
- $H \subset \mathbb{R}^m$ is the state space of the hybrid state vector, with $m \in \mathbb{N}$ the number of hybrid state variables,
- C is the set of dynamic equation sets associated with numerical places, representing continuous dynamics,
- $\mathcal{F}$ is the set of dynamic equation sets associated with hybrid places, representing hybrid dynamics,
- Ω is the set of conditions associated with transitions,
- M_0 is the initial marking of the Petri net.

The marking M_k of the HPPN at time k is composed of tokens, that can be symbolic, numerical or hybrid tokens:

$$M_k = \{M_k^S, M_k^N, M_k^H\}. \tag{1}$$

Symbolic places model the discrete states of the system and are marked by configurations. Σ is the sets of events of the system. An event $e \in \Sigma$ is a couple (v, k) where $v \in E$ is an event label and k the time of occurrence of e. A *configuration* δ_k^i with $i \in \{1, ..., |M_k^S|\}$ is a symbolic token at time k and represents a possible set of events b_k^i that occurred on the system until time k. $b_k^i = \{e_j\}$ with $j \in \{1, ..., |b_k^i|\}$ and for any event $e_j = (v, \kappa)$, $\kappa \leq k$.

A numerical place $p^N \in P^N$ is associated with a set of dynamic equations $C(p^N)$ modeling system continuous dynamics and its corresponding model noise and measurement noise. They are marked with particles. A *particle* π_k^i, with $i \in \{1, ..., |M_k^N|\}$ is a numerical token at time k and represents a possible continuous state $x_k^i \in X$ of the system at time k.

A hybrid place $p^H \in P^H$ is associated with a set of dynamic equations $H(p^H)$ modeling system hybrid dynamics. They are marked with hybrid tokens. A *hybrid token* h_k^i, with $i \in \{1, ..., |M_k^H|\}$ is linked with a configuration δ_k^j and a particle π_k^l, and represents a possible hybrid state $d_k^i \in H$ of the system at time k.

The initial marking M_0 of a HPPN carries the system initial states b_0, x_0 and d_0.

A condition $\Omega(t)$ associated with a transition $t \in T$ is a Boolean function that combines tests on the values of the tokens in the input places of t. Let $^\circ t$ (t°) designate the set of input (output) places of t. A condition must involves at least one token in each place in $^\circ t$. A condition involving more than one type of tokens can be satisfied only if the tokens are linked with hybrid tokens. If $\Omega(t)$ involves a configuration δ_k, it can deal with the occurrence of an event labeled with $v \in E$ (faults, mission events, interaction with the environment, ...). In that case, it takes the form $occ(b_k, v)$, to test if the set of events b_k of δ_k contains the event (v, k). A condition $\Omega(t)$ that involves a particle π_k can concern the continuous state. For example, $c(x_k) < B$ tests if the constraint equation c on the numerical state vector x_k of π_k is greater than a threshold B. In the same way, a condition involving a hybrid token h_k can deal with the hybrid state by constraining the hybrid state vector d_k of h_k, e.g. $\varsigma(d_k) \geq \beta$. Finally, a condition that involves more than one token can be a Boolean expression combining two or the three kinds of conditions above, e.g. $\Omega(t)(\delta_k, \pi_k, h_k) = occ(b_k, v) \wedge (c(x_k) < B) \vee (\varsigma(d_k) \geq \beta)$.

An annotation $\varrho \in \mathcal{A}$ is associated with any arc $a \in A$ that connects a transition t to a symbolic place p^S. It is an assignment function defined as follows: if $\Omega(t)$ deals with the occurrence of an event labeled with $v \in E$, $\varrho(\delta^i)$ adds the event (v, k) to the event set b^i of δ^i, when δ^i is moved to p^S after the firing of t at time k.

Health Modeling. With the definition of the HPPN above, it is possible to build a health-oriented model of a hybrid system. We consider the system modes as health modes (nominal, degraded and failure modes). Symbolic places represent the different discrete health states of the system. Numerical (resp. Hybrid) places represent various system continuous (resp. hybrid) dynamics. Health modes are thus combinations of discrete states, continuous dynamics and hybrid dynamics. Transitions model changes of health modes, so any transition $t \in T$ must have three places (one of each type) in its sets of input places and three places in its set of output places. Two transitions cannot have both the same set of input places and the same set of output places.

An anticipated fault is represented by an unobservable event $f \in \Sigma_{uo} \subset \Sigma$, where Σ_{uo} is the set of unobservable events. Fault events are abstractions of changes of health mode that might be unobservable or difficult to describe as conditions on the continuous state.

Finally, we use conditions to model the change of health modes and then let the degradation state affect the system evolution. For example, if the degradation is modeled by a fault occurrence probability, a condition on the hybrid state can be a Boolean function satisfied if the probability is higher than a given threshold.

Example 2. The HPPN-based model of the mobile robot is presented in Fig. 2. Symbolic places are represented by places with regular thicknesses, while numerical and hybrid places are represented by places with medium and large thicknesses, respectively. Arcs that connect transitions and symbolic (numerical and hybrid) places are represented by solid (dashed and dotted) arrows.

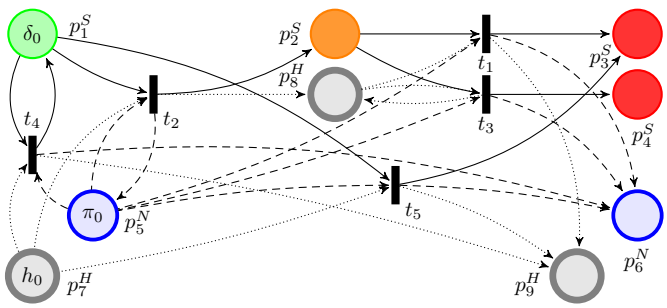

Fig. 2. Health-oriented model of the mobile robot using HPPN.

We decompose the five health modes of the robot into four symbolic places, two numerical places and three hybrid places. Four discrete health states are identified from the robot description (Fig. 1). One nominal state, one degraded state, and two different failure states are represented by the four symbolic places p_1^S, p_2^S, p_3^S and p_4^S, respectively. The two numerical places p_5^N and p_6^N represent the continuous dynamics C_1 and C_2. The three hybrid places p_7^H, p_8^H and p_9^H represent the hybrid dynamics H_1, H_2 and H_3, respectively. Five transitions represent the health mode changes. For example, transition t_4 represents the change from mode $Nominal_1$ to mode $Nominal_2$ so ${}^\circ t_4 = \{p_1^S, p_5^N, p_7^H\}$ and $t_4^\circ = \{p_1^S, p_6^N, p_9^H\}$.

The initial mode is $Nominal_1$ so δ_0, π_0 and h_0 are in p_1^S, p_5^N and p_7^H, respectively. At time $k = 0$, no event has occurred, so $b_0 = \{\}$. The only estimated state is the velocity, so $x_0 = [v_0]^T$ with $v_0 > 0$ because the velocity is initially positive. The initial fault occurrence probabilities $\rho_0^{f_1}$ and $\rho_0^{f_2}$ are very low. Thus, $d_0 = [\rho_0^{f_1}, \rho_0^{f_2}]^T$ with $\rho_0^{f_1} = 0.01$ and $\rho_0^{f_2} = 0.05$.

The condition $\Omega(t_4)(\delta_k, \pi_k, h_k) = occ(b_k, turn\ off) \wedge (x_k^0 \leq 0)$ tests if an event labeled with $turn\ off$ occurred at time k and if v_k is 0. We assume

that a fault occurs if its probability of occurrence is greater than 0.9. Consequently, the condition associated with transition t_2 is $\Omega(t_2)(\delta_k, \pi_k, h_k) = occ(b_k, f_1) \vee (d_k^0 > 0.9)$. With the same reasoning, we obtain $\Omega(t_1)(\delta_k, \pi_k, h_k) = occ(b_k, f_2) \vee (d_k^1 > 0.9)$, $\Omega(t_3)(\delta_k, \pi_k, h_k) = occ(b_k, f_2) \wedge (x_k^0 \leq 0) \vee (d_k^1 > 0.9)$ and $\Omega(t_5)(\delta_k, \pi_k, h_k) = occ(b_k, wall)$.

3.2 Hybrid System Diagnosis

In a health monitoring context, diagnosis aims at tracking the system current *health state*. The system health state is the combination of its discrete, continuous and hybrid states. In earlier work, we proposed to build a *diagnoser* from a HPPN model [9]. The HPPN-based diagnoser is generated based on the HPPN specifying the system model. It is a HPPN that monitors both the system behavior and degradation under uncertainty. Its online process takes as inputs the set of observations on the system. The output of the diagnoser at any time k is an estimation of the system health state that takes the form of a marking of the diagnoser $\Delta_k = \hat{M}_k$.

Uncertainty. Several types of uncertainty are taken into account. Knowledge-based uncertainty must be taken into account because the model does not reflect perfectly reality, as for the symbolic part of the model than the numerical one. Due to the inherent imprecision of sensors, we also consider uncertainty about observations. Regarding the symbolic aspects, the possible observation of an event that has not really occurred and the non observation of an observable event that occurred are taken into account. Symbolic uncertainty is dealt with using *pseudo-firing* (i.e. duplication) of tokens [17, 24]. Numerical uncertainty embodies the fact that the numerical values are imprecise. It is often dealt with through an estimator, that aims at estimating the continuous state according to model noise and measurement noise. We use particle filters to estimate the continuous state through the set of particles of the HPPN. The links between the configurations and the particles, provided by the hybrid tokens, are used to prevent the particle distribution to be disturbed by pseudo-firing.

Diagnoser Generation. Let us suppose that the health-oriented system model is a HPPN given by a tuple $\langle P, T, A, \mathcal{A}, E, X, H, C, \mathcal{F}, \Omega, M_0 \rangle$ as defined in Sect. 3.1.

The set of places of the diagnoser remains the same as the one of the model. Concerning the conditions associated with transitions, two aspects have to be taken into account. First, any Boolean function dealing with an event occurrence that is part of a condition $\Omega(t)$ is removed from it, in order to manage symbolic uncertainty (see Sect. 3.2). Arc annotations, however, are conserved to monitor event occurrences. Secondly, conditions on the hybrid state must also be substituted because a diagnoser works with observations (the degradation is estimated but not corrected with observations).

To improve computational performance, transitions of the HPPN are transformed following several rules defined in [11]. Basically, some transitions are merged and other are created in a way that the HPPN is separated in two levels. The *behavioral level* contains only the symbolic and numerical places, while the *hybrid level* contains the hybrid places. New transitions (called *hybrid transitions* in previous works) connect hybrid places. A hybrid token $h_k \in \hat{M}_k^H$ is moved from one hybrid place to another if it satisfies a condition associated with hybrid transition. Theses conditions are called *hybrid conditions* in previous work. The satisfaction of a hybrid condition depends on the places in which δ_k and π_k belong at time k, where δ_k and π_k are the configuration and the particle associated with h_k.

Example 3. Figure 3 shows the two levels of the HPPN-based diagnoser of the mobile robot example. The hybrid places are isolated and the hybrid transitions $\{t_i^H\}$ with $i \in \{6, ..., 11\}$ are added to the net. The condition associated to t_2 becomes $\Omega(t)(\delta_k, \pi_k) = \top$, a function returning *true* for any δ_k and π_k, because it does not depend on the continuous state. With the same reasoning, $\Omega(t_1)$ and $\Omega(t_3)$ become also $\top$, while $\Omega(t_4)$ and $\Omega(t_5)$ become $x_k^0 \leq 0$. Then transitions t_1 and t_3 (t_4 and t_5) have been merged because they were associated with the same condition, they have the same input places $\{p_2^S, p_5^N\}$ ($\{p_1^S, p_5^N\}$) and the numerical place p_6^N in their set of output places. The merging of t_4 and t_5 into t_{45} is useful to monitor at time k the possibilities to be in mode $Failed_1$ (δ_k^1, $\{\pi_k^i\}$ and $\{h_k^j\}$) and the one to be in mode $Failed_2$ (δ_k^2, $\{\pi_k^i\}$ and $\{h_k^l\}$) with the same set of particles $\{\pi_k^i\}$. This is particularly convenient because the particle filtering computation time increases with the number of particles.

Diagnoser Marking. In particle filtering, the number of particles defines the precision of the filter. A *possible mode* of the system is represented by a set of tokens composed of a configuration, n_k particles, and the n_k hybrid tokens that link the configuration to the particles, where n_k is representative of the

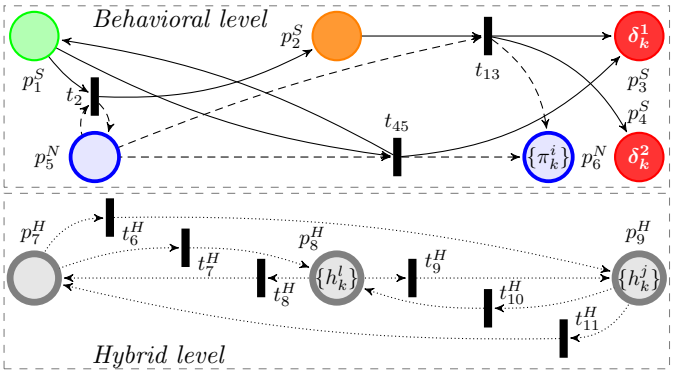

Fig. 3. HPPN-based diagnoser of the mobile robot.

precision associated to the monitoring of the mode at time k. The initial marking $M_0 = \{M_0^S, M_0^N, M_0^H\}$ represents the system's initial mode. It is composed of one configuration with value b_0, n_0^N particles with value x_0 and n_0^N hybrid tokens with value d_0, where n_0^N is the initial number of particles. The estimated marking at time k, $\hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N, \hat{M}_k^H\}$ where $\hat{M}_k = \hat{M}_{k|k}$, represents all the possible modes at time k. As long as only one mode is considered in the initial marking, two possible modes cannot share the same configuration, at any time k. However, two possible modes can share the same set of particles if they have the same continuous dynamics but different discrete states (see Example 3). As a consequence, the performance of the diagnoser regarding the uncertainty management is improved, in a way that the number of calculations is reduced where it can be. This is particularly true during the online process.

Diagnoser Process. The online process of the diagnoser is based on the evolution of the marking and on particle filters. A prediction step and a correction step are performed on the tokens to compute the marking of the diagnoser $\hat{M}_k$ at time k according to the observations $O_k = O_k^S \cup O_k^N$, where O^S and O^N respectively represent the observations corresponding to the symbolic part and the numerical part.

The prediction step aims at determining all possible next states of the diagnoser $\hat{M}_{k+1|k}$. It is based on the firing of the enabled transitions and on the update of the token values. All the enabled transitions are fired according to the rules described in [10]. This implies the assumption that a single event can occur at time k. The event set b_k of a configuration δ_k moved through an arc $a \in A$ during the transition firing, is updated according to the annotation $\mathcal{A}(a)$. The value x of a particle π is updated according to the continuous dynamics associated to the numerical place $p^N \in P^N$ in which π belongs after the transition firing. Noise is added during the particle value update to take into account uncertainty about model continuous dynamics. The value d of a hybrid token h is updated according to the hybrid dynamics associated to the hybrid place $p^H \in P^H$ in which h belongs after the transition firing.

The correction step updates the predicted marking $\hat{M}_{k+1|k}$ to the estimated marking $\hat{M}_{k+1|k+1}$ according to new observations O_{k+1}. It is based on the computation of the scores of all the possible modes represented by the marking and on the resampling of the tokens depending on the scores of the possible modes they represent. The scores of all possibles modes are computed with Pr^S and Pr^N, the probability distributions over the symbolic and the continuous states, respectively. Pr^S is the configuration weights. A configuration weight is computed as the inverse of exponential of the distance between the configuration event set and $O_{k+1}^- = \{O_\kappa | \kappa \leq k+1\}$, the set of symbolic observations until $k+1$. Pr^N is the normalized particle weights, calculated according to the distance between the particle values and numerical observations O_{k+1}^N. Then, the score of one possible mode is computed using a weighted function of the sum of

its particle weights and its configuration weight:

$$Score(\delta_k^i, \{\pi_k^j\}, \{h_k^l\}) = \alpha \times Pr^S(\delta_k^i) + (1-\alpha) \times \sum_{j=1}^{n_k^N} Pr^N(\pi_k^j). \quad (2)$$

where $\alpha \in [0,1]$ is the coefficient indicating the global confidence of the symbolic part relatively to the numerical part and $n_k^N = |\{\pi_k^j\}|$ is the number of particles considered for the given possible mode. The score of a possible mode is always between 0 and 1. A decision making process associates a new number of particles n_{k+1}^N to each set of particles, according to the best score of all the possible modes it belongs (see Sect. 3.2) and three scale parameters, denoted n_{min}^N, n_{suff}^N and n_{max}^N, of the HPPN. Each set of particles is then resampled with its associated n_{k+1}^N particles, like in classical particle filter. Parameters n_{min}^N and n_{suff}^N are respectively the minimum and the sufficient numbers of particles (but also hybrid tokens) to monitor a possible mode. It means that any n_{k+1}^N is chosen to satisfy the predicate $n_{min}^N \leq n_{k+1}^N \leq n_{suff}^N$. Parameter n_{max}^N is the maximum number of particles (hybrid tokens) available to monitor all possible modes. It means the total number of particles after the resampling is always less than or equal to n_{max}^N. During the resampling, hybrid tokens linked to duplicated particles are duplicated while those linked to deleted particles are deleted. Finally, configurations that are no longer linked with any hybrid tokens are deleted. The correction mechanism highlights that the hybrid tokens, in addition to estimate the hybrid state, prevent the particle distribution of one possible mode to be disturbed by the particle distributions of the other possible modes. In particle filtering, the number of particles defines the precision of the filter but is also a computational performed factor. The HPPN scale parameters thus compromise the number of possible modes to monitor and the precision granted to each one of them, relative to the available computational power (n_{max}^N can be set up to fulfill performance constraints).

The diagnosis Δ_k is deduced from the marking of the diagnoser at time k:

$$\Delta_k = \hat{M}_k = \{\hat{M}_k^S, \hat{M}_k^N, \hat{M}_k^H\}. \quad (3)$$

It represents the distribution of beliefs over the current health mode and how this mode has been reached. In other words, the marking $\hat{M}_k$ indicates the belief over the continuous state, the fault occurrences and the system degradation. The HPPN-based diagnoser results include the results of a classical diagnoser in terms of fault occurrences. In a classical diagnoser, however, every possible diagnosis has the same belief degree. A HPPN-based diagnoser handles more uncertainty and evaluates the ambiguity according to the tokens places and values.

4 Case Study

This section focuses on the application of the proposed methodology on the K11 planetary rover prototype. The K11 is a four-wheeled rover designed as a platform for testing power-efficient rover architectures in Antarctic conditions [16].

The K11 has then been redesigned by NASA Ames Research Center for diagnostics and Prognostics-enabled Decision Making research [1, 7, 23]. It has been transformed into a testbed to simulate some fault occurrences and failures. In this work, it is studied as a functional rover exposed to failures and executing missions.

4.1 Rover Description

The K11 rover is powered by twenty-four 2.2 Ah lithium-ion single cell batteries. A typical mission of the rover consists in visiting and performing desired science functions at a set of waypoints, before joining its charging station. A decision making module (DM) is responsible for determining the order in which to visit the waypoints according to the terrain map, the waypoint positions and rewards, and the rover conditions. The rover has four wheels, denominated by their location: the front-left (FL) wheel, the front-right (FR) wheel, the back-left (BL) wheel and the back-right (BR) wheel. Each wheel is driven by an independent 250 W graphite-brush motor, with control performed by a single-axis digital motion controller. An onboard laptop computer runs the control and data acquisition software. The rover is a skid-steered vehicle, meaning that the wheels cannot be steered and the rover is rotated by commanding the wheel speeds on the left and right sides to different values. The battery management system provides battery charging and load balancing capabilities. It also sends voltage and temperature measurements for each of the individual cells to the onboard computer. The data acquisition module collects current and motor temperature measurements and sends them to the onboard computer. The motor controllers send back motion data such as commanded speeds and actual speeds. More details on the rover can be found in [1].

All the continuous observations on the rover and the list of faults we consider in this study are presented in Table 1. Four signals command the wheels with a proportional-integral-derivative controller and the set of sensors returns 61 measurement signals. Several fault types have been implemented on the testbed and are related to the power system (battery), the electro-mechanical system (motors, controller), and the sensors (drift, bias, scaling or failure).

The K11 rover has no discrete actuator or discrete sensor and thus has mostly been studied as a continuous system, where faults were defined as constraints on the continuous state. We propose to abstract anticipated faults into unobservable events. The multi-mode system that describes the rover health evolution is presented in Fig. 4. To simplify the description, only a part of the multi-mode system is shown. The modes corresponding to consecutive fault occurrences are not included and only the front-left motor is considered.

The rover is in mode $Nominal_1$ with continuous dynamics C_1 as long as no fault has occurred. Fault f_1 occurrence represents the *end of discharge* (EOD) of the battery, i.e. the date when the battery is too discharged to power the system. This is assumed to occur when the battery voltage is lower than 3.25 V and it leads to the mission failure (mode $Failed_1$ with continuous dynamics C_5).

Table 1. Continuous commands, continuous measurements, and fault types on the K11

Command type	Comments	Units
Wheel speed	Commanded speeds for wheels on the same side are the same	rad/s
Measurement type	Comments	Units
Wheel speed	One for each wheel	rad/s
Total current	A current sensor on the power bus	A
Motor current	One for each motor	A
Motor temperature	One for each motor	°C
Battery temperature	One for each battery cell	°C
Battery voltage	One for each battery cell	V
Fault event labels	Fault descriptions	Effects
f_1	Battery charge depletion	Lead to failure
f_2	Parasitic electric load	Increase battery drain
f_3, f_4, f_5, f_6	Increased motor frictions	Increase battery drain and motor temperatures
f_7, f_8, f_9, f_{10}	Motor overheating	Lead to failure
$f_{11}, f_{12}, f_{13}, f_{14}$	Failed motor temperature sensors	Unable to estimate motor temperatures

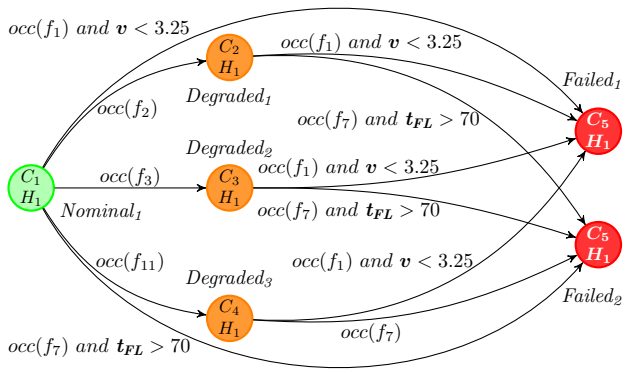

Fig. 4. Streamlined description of the rover health evolution.

Fault f_2 represents the emergence of a parasitic battery load arising from an electrical submodule continuously engaged, for example. The parasitic load increases the total current and thus the battery drain (mode *Degraded₁* with continuous dynamics C_2), which causes the system to reach the EOD prematurely. Fault f_3 (f_4, f_5 and f_6) represents an increased friction of the FL (FR, BL and BR) motor. The increased friction induces the need for a larger amount of current to satisfy the same speed (mode *Degraded₂* with continuous dynamics C_3). Furthermore, the load demands will be higher, raising the motor temperature. The most feared scenario for a motor is an overheating. In such case, the heat will eventually destroy the insulation of the windings, causing electrical shorts and leading to motor failure. The overheating of the FL (FR, BL and BR) motor is represented by fault f_7 (f_8, f_9 and f_{10}). The occurrence of any one of these faults leads to the rover failure (mode *Failed₂* with continuous dynamics C_5) and thus represents the rover *end of life* (EOL). A motor is assumed to overheat when its temperature exceeds 70 °C. The motor temperatures are measured by four sensors. These sensors, however, are known to fail unexpectedly, sending inconsistent values. These failures are represented by faults f_{11} f_{12}, f_{13} and f_{14}. We consider that the temperature model is not accurate enough without a correction step with observations. As a consequence, once f_{11} (f_{12}, f_{13} and f_{14}) has occurred, the occurrence of fault f_7 (f_8, f_9 and f_{10}) does not match with any condition on the FL (FR, BL and BR) motor temperature (see the arc between *Degraded₃* and *Failed₂*). In Fig. 4, mode *Degraded₃* with continuous dynamics C_4 represents the mode where the temperature sensor of the FL motor has failed. The rover has no hybrid state to monitor, so all modes have the same hybrid dynamics H_1, which corresponds to the identity dynamics.

4.2 Rover Modeling

Considering all the motors and the consecutive fault combinations, we identified 192 modes and 240 mode changes. The HPPN-based model of the rover has 241 places (192 symbolic, 48 numerical places, 1 hybrid place) and 240 transitions. The HPPN-based diagnoser has the same number of places and transitions. The merging step of the diagnoser generation does not reduce the number of transitions (specific to the case study) but still the hybrid place is removed from the transition inputs and outputs, reducing the complexity of the net. Because there is only one hybrid place, there is no hybrid transition. The underlying DES of the multi-mode system and HPPN-based model and diagnoser of the K11 rover are available at https://homepages.laas.fr/echanthe/PetriNets2016.

The nominal continuous model is represented as a set of differential equations that unifies the battery model with the rover motion model and the temperature models. It can be converted to a discrete-time representation and solved with a sample time of $1/20s$, while continuous observation sampling is about $1s$. We consider 30 state variables for the rover, including the rover 3-dimensional position, its relative angle position, the wheel control errors, the motor temperatures and motor winding temperatures. The 24 batteries are lumped into a single one to only consider 5 battery state variables (3 charges, the temperature and the

voltage) instead of 120. The battery model has been validated with experimental data in previous works [7, 23]. Unifying the battery model with motion and temperatures, however, increases uncertainty about the rover model.

Fault f_2 occurrence and effect are modeled as a time varying parameter. The parasitic battery load is captured as an additional current reaching a value between 1.5 A and 4.5 A from value 0 A in a few seconds after the fault occurrence. First, two parameters are added to the continuous state vector to monitor both the duration since the fault occurrence and the additional current value. Then, the uncertain rise of the additional current is modeled by adding a Gaussian noise, with a mean and standard deviation values starting respectively at 3 and 0.3, and decreasing to 0 while the duration since the fault occurrence increases.

Finally, the temperature model is quite uncertain so temperature measurements are assumed to be reliable when sensors are not failed. We model fault $f_{11}, f_{12}, f_{13}, f_{14}$ by increasing significantly the motor temperature sensor noise because failed sensors only send inconsistent large values with no pattern. Fault f_3, f_4, f_5 and f_6 and increased motor frictions can be modeled with time varying parameters (as additional motor resistances) like f_2 but are not monitored in this study.

4.3 Results

The HPPN framework is implemented in Python 3.4. The tests were performed on a 4 Intel(R) Core(TM) i5-4590 CPU at 3.30 GHz with 16 GB of RAM and running GNU/Linux (Linux 3.13.0 − 74, x86_64). In order to reduce computation time, the token value update step is multithreaded on the 4 physical cores. The rest of this implementation only uses one core.

Two scenarios studied in [23] are considered in this work. The rover mission is to visit a maximum of 12 waypoints and to go back to its starting position. All waypoints have different associated rewards. In nominal conditions, the rover DM system returns a 5-waypoints path, starting and finishing at the same position. For all scenarios, the K11 rover starts at 0s with batteries fully charged and with all components at the ambient temperature. The K11 rover currently has, however, 2 motor temperature sensors (FL and BL) failed. These faults do affect the monitoring but not the physical system, so the DM returns the same path as in nominal conditions.

The sensors faults are diagnosed in one sampling period by the diagnoser if we consider the initial mode to be unknown. We assume to know the rover initial degraded mode.

For the sake of clarity, in the rest of the paper, modes are designated with representative keywords of the rover state. For example, the initial mode is designated as *Sensor BL FL fault*. The initial number of particles and hybrid tokens is $n_0^N = 100$. Finally, due to the high uncertainty related to the unified model of the rover, we set the scale parameters to $(n_{min}^N, n_{suff}^N, n_{max}^N) = (40, 80, 6000)$.

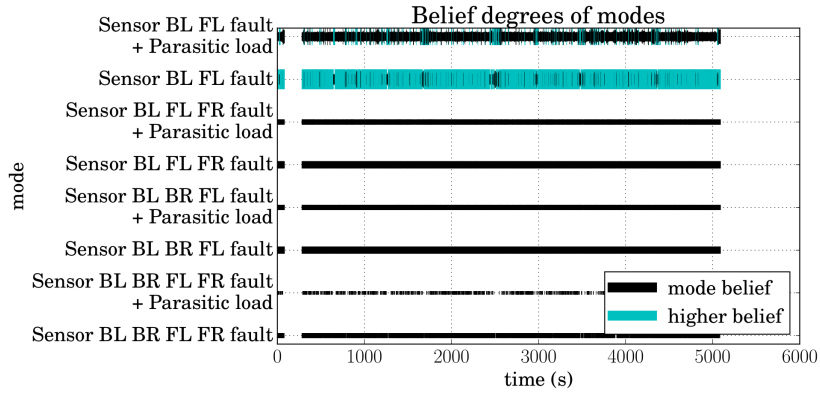

Fig. 5. Scenario 1: Mode belief at any time.

Scenario 1. In Scenario 1, no fault occurs. The rover successfully executes its mission. Figure 5 presents the distribution of beliefs over the current health mode at any time.

The belief degree of a possible mode is its score computed with Eq. 2 and α set to 0.5. Any belief degree is between 0 and 1, but the sum of the belief degrees of all possible modes is not 1. In Fig. 5, the maximum belief degree of a mode at any time is represented by the thickness of the line and the highest belief degree of all the modes is plotted in blue. The gap between 81s and 281s corresponds to a break during the experiment. The figure shows that the diagnoser keeps the real mode *Sensor BL FL fault* in its set of candidates and assigns it the highest belief degree almost all along the scenario. Other modes are also highly considered by the diagnoser at any time because of the model-based uncertainty.

Scenario 2. In Scenario 2, a battery parasitic load occurs between 660s and 695s, and the DM system cancels the visit of the farthest waypoint. Fault f_2 occurrence is immediately detected by the diagnoser (Fig. 6). After 678s, the possibility of being in mode *Sensor BL FL fault + Parasitic load* is the highest until the end of the mission. The fault load is estimated (most likely) at 1.39 A at 678s, 1.73 A at 679s, 2.16 A at 683s and 2.16 A at 3906s. A zoom between 570s and 760s on the trajectories of the modes that are still possible at 3906s (Fig. 7) shows that fault f_2 is believed to occur between 631s and 694s, and most likely between 677s and 689s. These results are consistent with our analysis of the measured total current.

Faults are always detected in one sampling period because the HPPN considers all possibilities during the online process prediction step and keeps the matching marking during the correction step. The results show that the diagnoser grants most of the time but not always, the highest belief to the real mode. The diagnosis, however, carries all the explanation of the observations as a distribution of beliefs, and then the real mode is always considered in the set

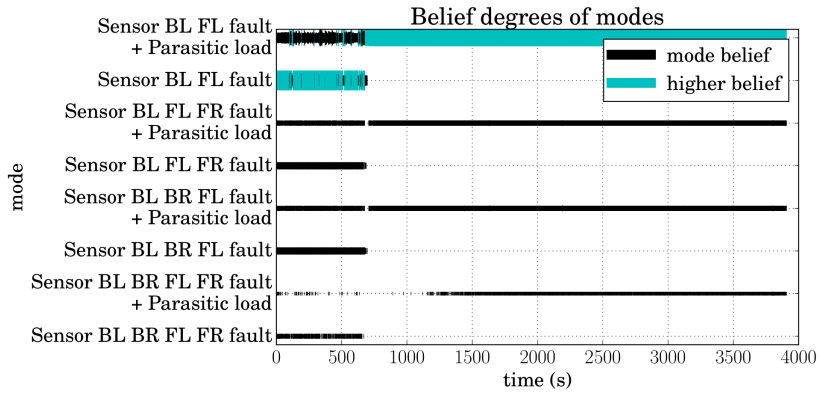

Fig. 6. Scenario 2: Mode belief at any time.

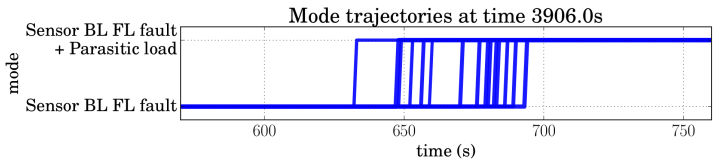

Fig. 7. Scenario 2: Trajectories of possible modes at time 3906s.

of candidates. This illustrates the robustness of the HPPN-based diagnoser to the rover model and data. The average diagnosis computation time and token number are 13.3s and 8801.4, respectively. These metrics point out the diagnosis computation time remains acceptable compared to the system model computational complexity. The maximum RAM used by Scenario 1 and 2 are 140.7 MB and 141.8 MB.

The case study results show that HPPN-based diagnosis is robust to real system data and constraints and adaptable to systems without discrete observations nor degradation knowledge.

5 Conclusion

This work applies the approach of health monitoring based on Hybrid Particle Petri Nets to a real case study, the K11 planetary rover prototype. The HPPN approach is particularly useful to take into account knowledge-based and observation-based uncertainty. The HPPN-based diagnoser deals with event occurrence possibility and knowledge imprecision. It monitors both discrete and continuous dynamics, as well as hybrid characteristics, such as degradation, in order to introduce concepts that will be useful to perform prognosis and health management of hybrid systems under uncertainty. In addition, diagnosis results can be used as probability distributions for decision making.

Then, the methodology was applied on the K11 rover. A hybrid model of the rover has been proposed by discretizing its health evolution and defining fault events. The model and diagnoser have been generated and two scenarios have been tested to illustrate the proposed method advantages. The diagnoser results are consistent with the expected ones and show that HPPN-based diagnosis is robust to real system data and constraints and adaptable to systems without discrete observations nor degradation knowledge.

In future work, futher scenarios will be tested. We also aim at formalizing and developing a prognosis process that will interleave diagnosis and prognosis methods to obtain more accurate results. The HPPN-based prognostics methodology will be defined and tested on a three-tank system as well as on the K11 rover.

References

1. Balaban, E., Narasimhan, S., Daigle, M.J., Roychoudhury, I., Sweet, A., Bond, C., Celaya, J.R., Gorospe, G.: Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms. Int. J. Prognostics Health Manag. **4**(006), 1–19 (2013)
2. Basile, F., Chiacchio, P., Tommasi, G.D.: Fault diagnosis and prognosis in Petri Nets by using a single generalized marking estimation. In: 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes. Spain (2009)
3. Bayoudh, M., Travé-Massuyes, L., Olive, X.: Hybrid systems diagnosis by coupling continuous and discrete event techniques. In: IFAC World Congress. pp. 7265–7270. Korea (2008)
4. Cabasino, M.P., Giua, A., Seatzu, C.: Diagnosability of discrete-event systems using labeled Petri Nets. IEEE Trans. Autom. Sci. Eng. **11**(1), 144–153 (2014)
5. Chanthery, E., Ribot, P.: An integrated framework for diagnosis and prognosis of hybrid systems. In: 3rd Workshop on Hybrid Autonomous System. Italy (2013)
6. Daigle, M., Roychoudhury, I., Bregon, A.: Qualitative event-based diagnosis applied to a spacecraft electrical power distribution system. Control Eng. Practice **38**, 75–91 (2015)
7. Daigle, M., Roychoudhury, I., Bregon, A.: Integrated diagnostics and prognostics for the electrical power system of a planetary rover. In: Annual Conference of the PHM Society. USA (2014)
8. Daigle, M., Sankararaman, S., Kulkarni, C.S.: Stochastic prediction of remaining driving time and distance for a planetary rover. In: IEEE Aerospace Conf. (2015)
9. Gaudel, Q., Chanthery, E., Ribot, P., Le Corronc, E.: Hybrid systems diagnosis using modified particle Petri nets. In: 25th International Workshop on Principles of Diagnosis. Austria (2014)
10. Gaudel, Q., Chanthery, E., Ribot, P.: Health monitoring of hybrid systems using hybrid particle Petri Nets. In: Annual Conference of the PHM Society. USA (2014)
11. Gaudel, Q., Chanthery, E., Ribot, P.: Hybrid particle Petri Nets for systems health monitoring under uncertainty. Int. J. Prognostics Health Manag. **6**(022), 1–20 (2015)
12. Genc, S., Lafortune, S.: Distributed diagnosis of place-bordered Petri Nets. IEEE Trans. Autom. Sci. Eng. **4**(2), 206–219 (2007)

13. Henzinger, T.: The theory of hybrid automata. In: 11th Annual IEEE Symposium on Logic in Computer Science. pp. 278–292 (1996)
14. Jianxiong, W., Xudong, X., Xiaoying, B., Chuang, L., Xiangzhen, K., Jianxiang, L.: Performability analysis of avionics system with multilayer hm/fm using stochastic Petri Nets. Chin. J. Aeronaut. **26**(2), 363–377 (2013)
15. Koutsoukos, X., Kurien, J., Zhao, F.: Monitoring and diagnosis of hybrid systems using particle filtering methods. In: 15th International Symposium on Mathematical Theory of Networks and Systems. USA (2002)
16. Lachat, D., Krebs, A., Thueer, T., Siegwart, R.: Antarctica rover design and optimization for limited power consumption. In: 4th IFAC Symposium on Mechatronic Systems (2006)
17. Lesire, C., Tessier, C.: Particle Petri Nets for aircraft procedure monitoring under uncertainty. In: Cardio, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 329–348. Springer, Heidelberg (2005)
18. Narasimhan, S., Balaban, E., Daigle, M., Roychoudhury, I., Sweet, A., Celaya, J., Goebel, K.: Autonomous decision making for planetary rovers using diagnostic and prognostic information. In: 8th IFAC Symposium on Fault Dectection, Supervision and Safety of Technical Processes. pp. 289–294. Mexico (2012)
19. Narasimhan, S., Browston, L.: HyDE - a general framework for stochastic and hybrid modelbased diagnosis. In: 18th International Workshop on Principles of Diagnosis. pp. 162–169 (2007)
20. Ru, Y., Hadjicostis, C.N.: Fault diagnosis in discrete event systems modeled by partially observed Petri Nets. Discrete Event Dyn. Syst. **19**(4), 551–575 (2009)
21. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.: Diagnosability of discrete-event systems. IEEE Trans. Autom. Control **40**(9), 1555–1575 (1995)
22. Soldani, S., Combacau, M., Subias, A., Thomas, J.: On-board diagnosis system for intermittent fault: Application in automotive industry. In: 7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems. vol. 7-1, pp. 151–158 (2007)
23. Sweet, A., Gorospe, G., Daigle, M., Celaya, J.R., Balaban, E., Roychoudhury, I., Narasimhan, S.: Demonstration of prognostics-enabled decision making algorithms on a hardware mobile robot test platform. In: Annual Conference of the PHM Society. USA (2014)
24. Zouaghi, L., Alexopoulos, A., Wagner, A., Badreddin, E.: Modified particle Petri Nets for hybrid dynamical systems monitoring under environmental uncertainties. In: IEEE/SICE International Symposium on System Integration. pp. 497–502 (2011)