INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

Dynamic Probabilistic Graphical Model for FDIR Process in Autonomous Network

AKILANDESHWARI.D#1, MRS.LAKSHMIDEVI.B*2
#1. P.G SCHOLAR,

*SURYA GROUP OF INSTITUTIONS #1,*2.*
VIKRAVANDI, VILLUPURAM

ABSTRACT

To exploit the modeling features and inference capabilities of dynamic Bayesian networks(DBN), in designing and implementing an innovative approach to fault detection, identification, and recovery (FDIR) for autonomous spacecrafts . In particular,issues like partial observability, uncertain system evolution and system environment interaction, as well as the prediction and mitigation of imminent failures can be naturally addressed by the proposed approach. The DBN framework can augment the modeling and analytical power of standard FDIR methodologies,while still being able to be integrated into the usual system modeling procedures (like, for instance, fault tree analysis).

An FDIR cycle composed of the tasks of diagnosis ,prognosis ,and recovery is introduced and characterized through a DBN model. In particular, by considering the execution of recovery actions in response to either a current or a future abnormal situation, both reactive as well as preventive recovery can be addressed respectively. The proposed approach has been implemented in an on-board software architecture

called Anomaly resolution and prognostic health management for autonomy (ARPHA).The FDIR analysis of the power supply system of the ExoMars rover, by considering different anomalous and failure simulated scenarios; we conclude that ARPHA is able to properly detect and deal with the simulated problems.

***Index Terms*—Autonomous spacecraft, dynamic Bayesian networks, fault detection identification and recovery.**

I. INTRODUCTION

AUTONOMOUS spacecraft operation relies on the adequate and timely reaction of the system to changes in its operational environment, as well as in the operational status of the system. Both the system environment and the system status can exhibit various degrees of uncertain behavior

particular, the operational status of the system is dependent on the internal sub-system and component reliability factors, as well as on the external environment factors affecting the system reliability and safety (e.g., thermal, radiation, illumination conditions) and on system-environment interactions (e.g., stress factors, resource utilization profiles, degradation profiles, etc.). To address possible system faults and failures, the system under examination must be provided with some m

The goal is a timely detection of faults and the initiation of the corresponding recovery action, often using static precompiled look-up tables, and basically concerned with the execution of the actions to put the spacecraft into a known safe configuration, thus trans-ferring control to the ground operations for troubleshooting and planning actual recovery.

Moreover, classical FDIR represents a reactive approach, that cannot provide and utilize prognosis for the imminent failures. Such approaches are capable of providing adequate results for the statically captured system configurations, but they can poorly deal with the dynamic aspects of the sys-tems, such as recovery actions and reconfiguration. They do not address evolution of the system characteristics and history of the system interaction with the environment.

In summary, standard on-board FDIR procedures do not reflect probabilistic causal dependencies between the faults and general system capabilities on one hand, and between the system-environment interaction evolution and system dependability characteristics on the other hand.To address the above issues, an approach to on-board (and autonomous) FDIR is needed which has the capability to reason about anomalous observations based on the global knowledge of the system and its capabilities, system envi-ronment, and system-environment interaction in the presence of uncertainty

II. PROBABILISTIC GRAPHICAL MODELS AND FDIR

PGM [9] are a class of probabilistic models that have recently gained a lot of attention outside artificial intelligence; in particular, their suitability to model and analyze a wide range of problems and situations involving system failures and recovery has contributed to their spread inside the reliability, availability, maintainability, and safety (RAMS) community [10]–[14]. Modeling of probabilistic causal dependencies is one of the main capabilities of PGM like Bayesian networks (BN), decision networks (DN), and their dynamic counterparts as DBN and dynamic decision networks (DDN) [9]. This class of models naturally captures dependencies and evolutions under partial observability. From an FDIR perspective, such dependencies can involve system components as well as system-environment interaction and evolution, and system dependability characteristics; moreover, in decision models the effect of external actions can be modeled as well, and utility functions can be exploited in order to select the most

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

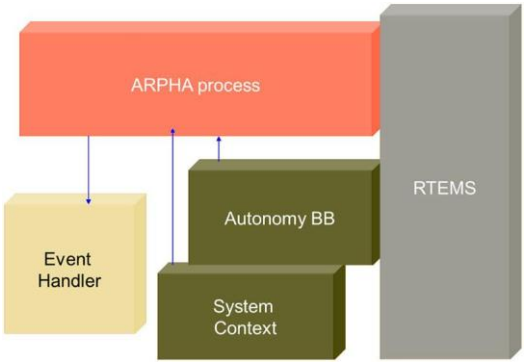


Fig. 1. VERIFIM block diagram.

(summarizing the history of the system uncertain evolution) and the effects of the recovery actions on future system states, the task of preventive recovery can be addressed as well.

III. VERIFIM FRAMEWORK

The ARPHA process is the core of the on-board autonomous FDIR and is intended to interact with an autonomy building block (ABB), setting, and exe-cuting a given spacecraft mission plan. The system context represents the interface between the external and the con-figuration environments (it represents a memory area that contains data received from sensors and from configuration parameters of the system); finally, the event handler compo-nent is in charge of handling the system events triggered by ARPHA, including the recovery events determined during the FDIR inference. In particular, in the VERIFIM framework, the following assumptions hold.

1) ARPHA inference takes place in a time interval called mission frame, which is a set of discrete time points separated by a constant temporal width _ called the discretization step.
2) The ABB plan actions are predefined, a given plan is loaded at the start of the mission frame, and only a single action is executed in a given instant; the ABB knows the current executing plan action and shares it with ARPHA through the system context.
3) In order to address recovery, ARPHA has a set of potential recovery policies composed of a set of temporally tagged atomic actions that the spacecraft can execute autonomously; this means that each policy consists of a sequence of autonomous actions with a temporal duration; the execution is triggered by the event handler and
4) The target system (i.e., the spacecraft) has a set of (possibly noisy) sensors concerning both its internal status (e.g., the battery level) and the external environment (e.g., the external temperature); the values of such sensors are made available to ARPHA (after a possible preprocessing phase like a discretization for continu-ous values) through the system context; ARPHA also has access to configuration parameters (possibly set by "ground") and to global information at the current time.

IV. DYNAMIC BAYESIAN NETWORKS FOR AUTONOMOUS SYSTEM HEALTH MANAGEMENT

Concerning the identification of dependencies and the quan-tification of the DBN model, in the VERIFIM project we adopted a mixed strategy involving FTA and knowledge engineering from domain experts. In particular, system and reliability engineers are very familiar with component-based formalisms like fault trees or dynamic fault tree (DFT) [20]; starting from a DFT model of our case study, we have been able to automatically compile a skeleton DBN model through the "DFT to DBN compiler" developed inside the RADYBAN tool. This has allowed us to greatly simplify the construction of the DBN model needed to implement the autonomous FDIR strategy.

Given the framework of Section III, we can characterize the DBN model used by ARPHA as follows.

1) The time interval between consecutive time slices is equal to the discretization step _ of ARPHA's mission frame.
2) Variables are discrete and can be subdivided as follows.
   a) One or more variables representing the possi-ble actions influencing the system behavior, either the plan actions the ABB can execute or the autonomous recovery actions being part of recovery policies.
   b) A set of variables representing the system components; the values of such variables correspond to the behavioral mode of the component (usually one "ok" mode and one or more failure or degraded modes).
3) Intraslice dependencies concern dependencies at the same time instant; they can be between internal system variables and system components, or between a system component and the sensor providing a measure of its status (the so called sensor or observation model).
4) Interslice dependencies concern the degradation model of the components (quantified for example by parameters like the failure rate), as well as the influence of actions (both plan actions and recovery actions) on system components or subsystems (the so called transition model); we assume that if a given (plan or recovery) action is executed at time t, the effect is measured at time $t + \_$, where _ is the discretization step.

   Finally, since we want to adopt a decision theoretic approach to select the best recovery policy, we identified a subset of variables whose outcomes are relevant for the recovery goals; a utility table is then built as a function of such selected variables

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

V. FDIR Inference Approach

In ARPHA, we decided to implement the FDIR analysis by resorting to junction tree (JT) inference algorithms [23]. In this class of algorithms, once the JT structure is obtained, one can get rid of the original network, so the actual on-board system model on which ARPHA is working is in fact a JT derived from the DBN produced by the modeling phase.

The decision of resorting to JT inference was made because approximate algorithms based on simulation, e.g., particle filtering [24], usually require several simulation runs in order to get a reasonable approximation, making them potentially unsuitable for real-time inference. Boyen–Koller (BK) algorithm . The algorithm depends on some input param-eters (set of nodes) called "clusters;" according to the clusters provided, it can produce approximate inference results with different degrees of accuracy.

A. ARPHA Implementation

From the implementation point of view, ARPHA is composed of one periodic process running (on a LEON3 CPU under RTEMS) in parallel to the other processes of on-board software. This "main" process, representing the on-board FDIR cycle, is composed of three sub-processes implementing the diagnosis, prognosis, and recovery tasks as indicated by the state diagram of Fig. 2.

These tasks are characterized as follows.

1) *Diagnosis at Time t*: A belief state on a set D of selected variables at time t (diagnostic variables like variables representing system components); i.e., the pos-terior probability at time t of each $d \in D$ given the evidence up to time t.
2) *Prognosis at Time* $\bar{t}$ *From Time* $t < \bar{t}$: The belief state of set D at time t, given the evidence up to time t (and possibly evidence about plan information up to $\bar{t}$ if available).
3) *Recovery at Time t*: Choice of the "best" available policy (i.e., the one with maximum EU), given the evidence up to time t.

In DBN terminology, diagnosis and prognosis are implemented with a filtering algorithm and correspond to a monitoring and prediction task respectively [7]. Recovery is implemented through filtering as well; it consists in setting the values of the variables involved in each policy to evaluate, followed by the computation of the EU of the current policy, and by finally selecting the one with the largest value of EU (maximum EU policy). The computation of the EU of a single policy is performed by computing the posterior probability of the variables involved in the utility function and by multiplying such probability with the corresponding utility value, summing up for all the variables of interest, i.e., those for which the utility table is defined (see Section VI for some examples).

(4)*Recovery at Time t*: Choice of the "best" available policy (i.e., the one with maximum EU), given the evidence up to time t.

In DBN terminology, diagnosis and prognosis are implemented with a filtering algorithm and correspond to a monitoring and prediction task respectively [7]. Recovery is implemented through filtering as well; it consists in setting the values of the variables involved in each policy to evaluate, followed by the computation of the EU of the current policy, and by finally selecting the one with the largest value of EU (maximum EU policy). The computation of the EU of a single policy is performed by computing the posterior probability of the variables involved in the utility function and by multiplying such probability with the corresponding utility value, summing up for all the variables of interest, i.e., those for which the utility table is defined (see Section VI for some examples).

In the current implementation, we assume that the management of policy events, triggered by recovery and managed by the Event Handler, runs concurrently to ARPHA. In this way, ARPHA does not have to wait for the conclusion of a recovery policy to perform a new diagnosis or prognosis on the system, but it can also consider the changes performed during the execution of a recovery policy to perform a new diagnosis or prognosis.

In summary, the following basic tasks are executed to implement the whole FDIR cycle of ARPHA (see again Fig. 2).

1) *Observation Collection:* Data necessary for on-board reasoning are periodically retrieved; more specifically, at the beginning of a mission frame, sensor, and plan data are retrieved from the system context and the ABB, respectively. Both kinds of data are converted to observations concerning the variables of the on-board model.
2) *Current State Detection:* Observations are inserted into the JT; then, inference is executed by JT propagation. Inspection of the probabilities of the diagnostic variables can provide the diagnosis at the current mission time. The possible system states are normal (no anomalies or failures are detected), anomalous (an anomaly is detected), or failed (a failure is detected).
3) *Reactive Recovery:* This kind of recovery is performed if the current state detection returns a failed state; after having incorporated the current evidence in the diagnostic phase, for each available recovery policy, the policy itself is loaded (propagated) into the on-board JT; this actually means to set a specific value to the DBN variables affected by the policy itself at the proper time slice.

ISSN : 2454-9924

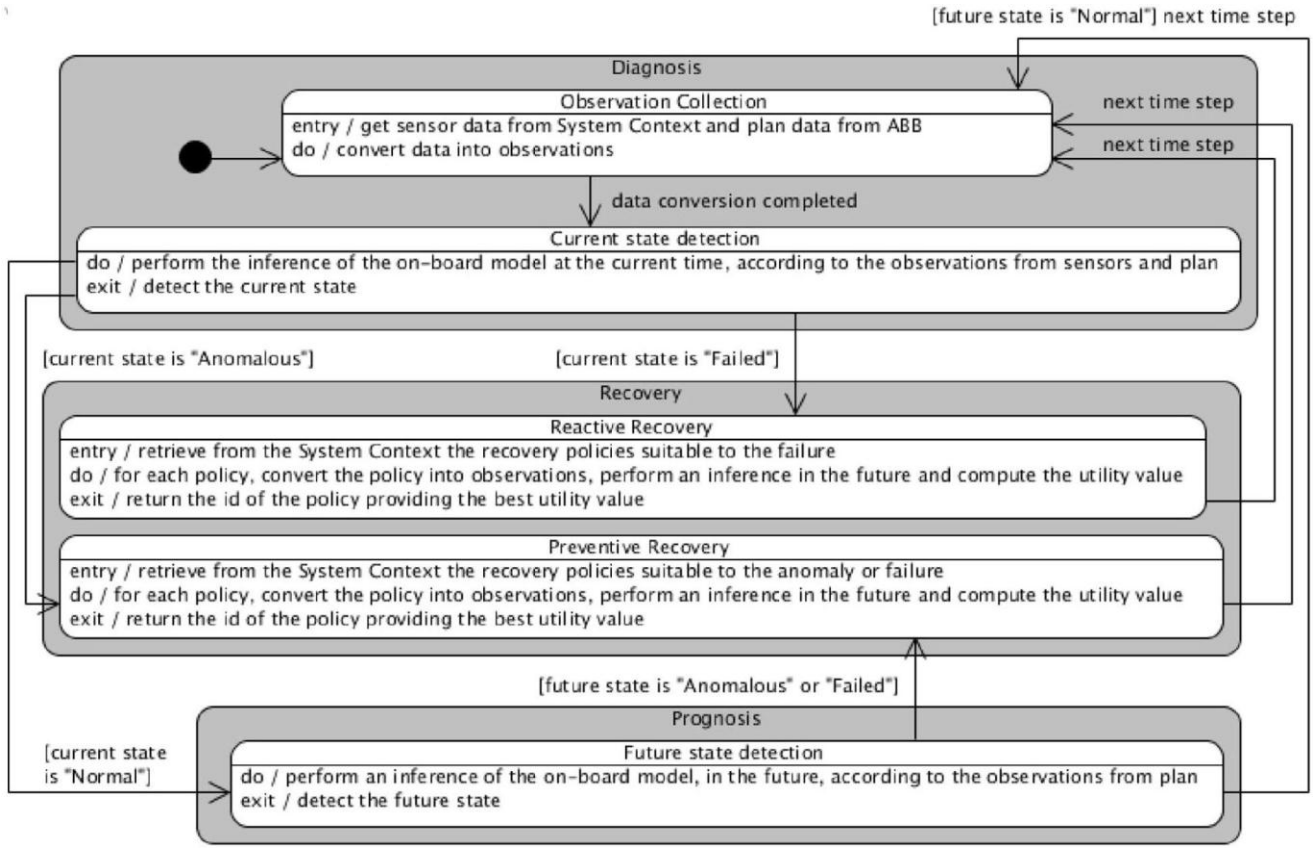


Fig. 2. FDIR cycle of ARPHA (state-chart diagram).

4) *Future State Detection:* If the current state is normal, then the time horizon t^- for prognosis is determined and JT propagation is performed with a time step of $\_$ (discretization step) until t^-, by also considering plan information at each time step as evidence.
5) *Preventive Recovery:* This kind of recovery is performed if the current state detection returns an anomalous state, or if the future state detection provides an anomalous or failed state; the choice of the best recovery policy fol-lows the same approach applied to the reactive recovery case, but the effects of policies are evaluated for a larger time horizon, since the model is analyzed for several time steps in the future.

6) VI. Application: FDIR for the Power Supply System of Mars Rover

In order to test ARPHA, we considered some simulated scenarios concerning the ExoMars rover. Moreover, since the actual rover was not available for the study, we employed a rover simulator (ROSEX), able to simulate the rover's sensoring and planning activities. In particular, the case study concerned the power supply system of the rover, with a particular attention to the following aspects and their combined behavior: rover's solar arrays, power load, and rover's battery.

A. Solar Arrays

We assume the presence of three solar arrays, namely SA1, SA2, and SA3.

In particular, SA1 is composed of two redundant strings, while SA2 and SA3 are composed of three strings. Each solar array can generate power if both the following conditions hold: 1) at least one string is not failed and 2) the combination of sun aspect angle (SAA), optical depth (OD), and local time (day or night) is suitable. In particular, the OD is given by the presence or absence of shadows or storms. The total amount of generated power is proportional to the number of solar arrays that are actually working.

B. Load

The amount of load depends on the current action performed by the rover.

C. Battery

We assume that the battery is composed of three redundant strings. The charge of the battery may be steady, decreasing, or increasing according to the current levels of load and generation by the solar arrays. The charge of the battery may be compromised by damage to the battery occurring in two situations: all the strings are failed, or the temperature of the battery is too low.

ISSN : 2454-9924

D. Scenario S1

In this scenario, we simulate the presence of a terrain slope that increases the SAA, causing lower power generation by solar arrays. The scenario S1 occurs when a low power genera-tion and a nonoptimal SAA are both present in the system. The SAA influences the degree of power generation; for instance, the SAA is optimal if the sun is perpendicular with respect to the solar arrays of the rover. The occurrence of the scenario may determine the anomalous state or the failed state of the system according to the degree of generated power (low or very low generation, respectively). Two recovery policies may be applied in case of detection of S1, with the aim of reducing the negative effects: (P1) transition to stand-by mode including the suspension of the plan, with the aim of reducing the load while the power generation is limited. (P2) change of inclina-tion of SA2 and SA3, with the goal of improving the SAA and consequently the level of power generation (the tilting system cannot act on SA1).

E. Scenario S2

In this scenario, we simulate the presence of dust or shadow that increases the OD and reduces the power generated by solar arrays. The scenario S2 occurs when both a low power generation and a compromised OD are present in the system. The presence of dust in the air or the rover positioned in a shadowed area reduces the irradiation of solar arrays, and the degree of generated power, as a consequence. In the best situation, there is no dust and the OD is null.

F. Scenario S3

In this scenario, we simulate an unexpected high request of energy by the drilling operation. The power generated by solar arrays may not be enough to cope with the request of energy, so the battery may be used to provide the additional power. The scenario S3 occurs when the level of battery charge is not optimal during the drilling operation. This leads the system to an anomalous or failed state according to degree of charge of the battery. The recovery policies facing S3 are the following: (P4) as above; the goal is improving the power generation thanks to a better SAA, and avoiding the use of the battery by suspending the drill. (P5) suspension of the plan, retraction of the drill if drilling is under execution, and transition to stand-by mode (with the aim of reducing the load).

H. DBN Model of the Case Study

In the DBN, if a variable has a temporal evolution it is rep-resented with two instances, one for each time slice (t, t + _). In particular, the instance at time slice t+_ is distinguished by the symbol # following the name of the variables (e.g., S1#). The two instances are connected by a temporal arc (appearing as a thick line in Fig. 3). Still in Fig. 3, the observable variables have a different color. The values coming from the sensors,the ABB, and the recovery policies, will become observations for such variables during the ARPHA cycles andthe inference analysis of the model.

G. Scenario S4

If the battery is damaged, the battery charge level may become low. The scenario S4 occurs when both the battery damage and the low battery charge characterize the system. The damage may be due to the low temperature of the battery or the failure of its strings. In this case, we simulate a damage due to a low external temperature; such a fault is transient, since an increase of the temperature can bring back the bat-tery to work correctly. The anomalous or failed state depends on the degree of charge. The recovery policies considered for S4 are P2 and P4 defined as above. In this case, the aim of the policies is to suspend the plan and then to try to get power from a better inclination of solar arrays. Notice that P2 and P4 are potentially different (even when a drilling operation is not involved), since P2 first stops the current plan and then moves the solar arrays, while P4 tries to move solar arrays before stopping the plan.

TABLE I
UTILITY FUNCTION FOR POLICIES

	gen=load	gen>load	gen<load
standby	0.2	0	0.9
drill	0.8	1	0.1
move	0.8	1	0.3
pancam	0.8	1	0.2
mast	0.8	1	0.2
wisdom	0.8	1	0.2
tilt	0.8	1	0.6
retract	0.2	0	0.9

The DBN model (Fig. 3) of the case study (obtained by first an automatic compilation from a DFT, then through introduc-tion of additional knowledge not available at the DFT level) has the following features.

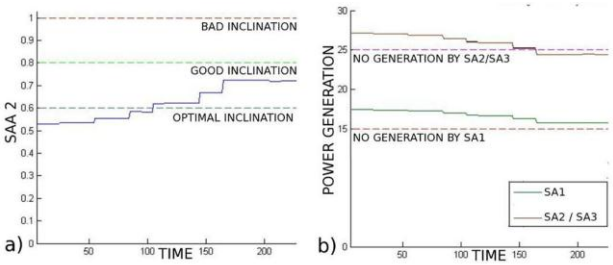
1) Solar Arrays: The variables representing the functioning or failure of basic components or subsystems, are binary; for example, StringSA11 and StringSA12 represent the state of the redundant strings composing the solar array SA1, while StringsSA1 represents the state of the set of strings. This variable, together with StringsSA1, influences the binary vari-able PowGenSA1 modeling the presence or absence of power generation by SA1. The size of PowGen is four, in order to represent four intermediate levels of power generation depending on the number of solar arrays generating power.

2) Load: The size of the variable ActionId is eight, in order to represent eight actions of interest in the model (in this exam-ple, actions may concern either the plan or the recovery, as previously mentioned). Such variable influences load whose size is five, in order to represent five intermediate levels of power consumption according to the action under execution. The variable balance is ternary and indicates if PowGen is equal to, higher or lower than load.

3) Battery: The state (working or failed) of each redundant string composing the battery is represented by BattString1, BattString2, and BattString3, while the state of the set of strings is modeled by BattStrings. The variable Temp is ternary and represents the temperature of the battery (low, medium, and high). Temp and BattStrings influence BattFail represent-ing the damage of the battery.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924



c) ARPHA output:

```
00   *** MISSION STEP: 3 (MISSION TIME: 187 sec.) ***
01   *************** ROSEX VALUES ***************
02   opticaldepht = 1.00000   pwrsa1 = 15.71248
03   pwrsa2 = 24.40224        pwrsa3 = 24.41719
04   saa1 = 0.72340           saa2 = 0.72340
05   saa3 = 0.72340           batterycharge = 90.24533
06   batttemp = 273.00000     time = 10.04274
07   SVF_action = 1           SVF_plan = 1
08   ********************************************
09     *** Diagnosis ***
10   Propagate PLAN STREAM
11   3:ActionId#:1 0 0 0 0 0 0 0
12   Propagate SENSORS STREAM
13   3:OpticalDepth#:1 0    3:PowGenSA1#:1 0
14   3:PowGenSA2#:0 1       3:PowGenSA3#:0 1
15   3:AngleSA1#:0 1 0      3:AngleSA2#:0 1 0
16   3:AngleSA3#:0 1 0      3:BattCharge#:0 0 0 1
17   3:Temp#:0 1 0          3:Time#:1 0
18   Current inference (STEP 3)
19   Pr{S1#=2} = 0.000<0.990    Pr{S2#=2} = 0.000<0.590
20   Pr{S3#=2} = 0.000<0.990    Pr{S4#=2} = 0.000<0.990
21   Pr{S1#=1} = 1.000>=0.990
22   Pr{S2#=1} under recovery or minor criticality
23   Pr{S3#=1} = 0.000<0.990    Pr{S4#=1} = 0.000<0.990
24   SYSTEM STATE: "ANOMALOUS" (S1#=1)
25     ## Preventive Recovery ##
26   Policy to convert: P1
27   Propagate POLICY STREAM
28    4:ActionId#:1 0 0 0 0 0 0 0
29    5:ActionId#:1 0 0 0 0 0 0 0
30    6:ActionId#:1 0 0 0 0 0 0 0
31    7:ActionId#:1 0 0 0 0 0 0 0
32    8:ActionId#:1 0 0 0 0 0 0 0
33    9:ActionId#:1 0 0 0 0 0 0 0
34   10:ActionId#:1 0 0 0 0 0 0 0
35   11:ActionId#:1 0 0 0 0 0 0 0
36   12:ActionId#:1 0 0 0 0 0 0 0
37   13:ActionId#:1 0 0 0 0 0 0 0
38   Future inference (STEP 13)
39   Utility Function= 0.4736
40   Policy to convert: P2
41   Propagate POLICY STREAM
42   4:ActionId#:0 0 0 0 0 0 1 0
43   5:ActionId#:0 0 0 0 0 0 1 0
44   Future inference (STEP 13)
45   Utility Function= 0.0622
46   Best policy for Preventive Recovery is P1
```

Fig. 4. Scenario S1. (a) Sun aspect angle (SAA) of SA2. (b) Power generation by SA1, SA2, and SA3. (c) ARPHA output at time step 3.

PowGenSA1. In other words, PowGenSA1 is observed to be equal to its first value (index 0) with probability 1 at the same time step, in order to represent that the solar array SA1 is generating enough power (*pwrsa*1 > 15) in that step [the second value (index 1) instead, represents that the generated power is too low]. We assume that the power is good if greater than 15 in the case of SA1, and greater than 25 in the case of SA2 and SA3 (SA1 is composed of two strings, while SA2 and SA3 are composed of three strings).

We provide an example of ARPHA execution during a simulated mission, in the scenarios defined at the beginning of Section VI.

1) S1: Slope of the Terrain: In Fig. 4(a), we show the SAA used to generate the power generation profile [Fig. 4(b)], exploiting the ROSEX. In time steps from 0 to 2, ARPHA estimates that both the current and the future state are nor-mal. At time step 3, diagnosis detects that the current state is anomalous. The output of ARPHA in this step is shown in Fig. 4(c): lines 01–07 contain the values of the sensors (generated by ROSEX) and the plan action under execution (SVF_action); lines 09–24 concern the diagnosis. In particular, at lines 10–11, the plan action (SVF_action=1="wait") performed in the current step is converted into the observation ActionId = 0; at lines 12–17, the sensor values are mapped into observations of the corresponding variable values: *pwrsa*1 = 15.71248 becomes PowGenSA1 = 0 (power generation by SA1 is high), *pwrsa*2 = 24.40224 becomes PowGenSA2 = 1 (power generation by SA2 is low), *pwrsa*1 = 24.41719 becomes PowGenSA3 = 1 (power generation by SA3 is low), *saa*1 = 0.722340 becomes AngleSA1 = 1 (SAA1 is not optimal), *saa*2 = 0.722340 becomes AngleSA2 = 1 (SAA2 is not optimal), *saa*3 = 0.722340 becomes AngleSA3 = 1 (SAA3 is not optimal), etc. Given such observations, ARPHA performs the inference of the model at the current time step (line 18), querying the variables S1#, S2#, S3#, S4# repre-senting the occurrence of the scenarios (lines 19–23). The probability that S1# = 1 is higher than a predefined threshold (line 21). In this way, ARPHA detects the scenario S1 and the anomalous state of the system (line 24); as a consequence, the preventive recovery is activated (lines 25–46), in order to evaluate the policies P1 and P2, suitable to deal with S1.

2) S2: Presence of Dust or Shadow: In Fig. 5(a), the ROSEX profile of OD is plotted. In Fig. 5(b), the ROSEX profile of power generated by solar arrays is plotted. At time steps 0, 1, and 2, ARPHA detects the normal state as the result of both diagnosis and prognosis. According to the out-put reported in Fig. 5(c), at time step 3, the normal state is still detected by diagnosis, but not by prognosis: lines 01–07 show the sensor values and the plan action.

3) S3: High Energy Request by Drill: In Fig. 6, we show the ROSEX battery profile that decreases in linear way. For the sake of brevity, we omit the output of ARPHA. No anoma-lies or failures are detected from time step 0 to time step 4. At time step 5, the drill action is performed while a reduced level of the battery is detected by sensors. The model infer-ence at the current step returns an anomalous state concerning the Scenario S3. Preventive recovery is activated and the poli-cies P4 and P5 are evaluated; P5 is suggested by ARPHA

ISSN : 2454-9924

4) S4: Damage to Battery: In Fig. 7(a) and (b), we pro-vide the ROSEX profile of battery temperature and charge, respectively. No anomalies or failures are detected from time step 0 to time step 26. At time step 27, a reduced level of the battery charge and the low temperature of the battery are indicated by the sensors. The model inference at the current step returns an anomalous state concerning the scenario S4. Preventive recovery is activated and the policies P2 and P4 are evaluated, and P4 is suggested by ARPHA. This is justified by the fact that the power level is enough to try a tilt before stopping the plan, which is definitely the best choice.

VII. Conclusion

In this paper, we have presented an approach to intelligent FDIR through the use of DBN. We have discussed how the tasks typical of FDIR analysis (monitoring, diagnosis, progno-sis, and recovery) can be naturally addressed in a DBN-based framework, by exploiting standard inference capabilities of such models. We have discussed how this approach can be applied in a real-world application, namely the FDIR anal-ysis of autonomous spacecraft. In choosing the architecture of ARPHA, we have taken into account different aspects: the uncertainty in the target system's evolution, the uncertainty resulting in autonomous actions, the utility provided by the available actions, the on-line inference capabilities required by the application at hand. We believe that all of the above issues, naturally lead to the adoption of PGMs (and of DBN in particular) as a suitable choice, together with the adoption of a JT-based algorithm, suitable to approximate inference, without resorting to simulation-based techniques (that could be very unsuitable for the kind of on-line inference required by our task).

Several approaches address the problem of monitoring and diagnosis of dynamic systems. The problem is tackled by resorting to hybrid dynamic Bayesian networks (HDBN) with a mixture of discrete and continuous variables in [25]; both partial observability and noisy sensors can be dealt with, and the inference task is implemented by combining a strategy similar to Kalman filter and approximation tech-niques based on the BK algorithm. However, the inference is limited to monitoring and diagnosis and the approach does not address the recovery task. Moreover, in this paper
the HDBN is obtained from a more abstract specifica-tion called temporal causal graphs (TCG) [26] which is not a standard system engineering language as in the case of DFT.

Another interesting probabilistic approach to the diagno-sis of electrical power systems (EPS) is presented in [29]; in this case the reference model is a static BN, obtained from a high-level specification language and then compiled into an arithmetic circuit (AC) for inference [30]. Differently from our approach, the emphasis of the work is particularly on the diagnosis of EPS faults, without considering the prob-lem of autonomous preventive and reactive recovery. Common aspects include the automatic generation of the PGM (a static BN in the case of [27] and a DBN in our case) from a specification language (which is a standard system engineer-ing language in our case), and the offline compilation of the graphical model into a run-time (on-line) inference engine (AC in [27] and JT in our case). While ACs are claimed to be very efficient, providing a fast (real-time) exact inference, the use of JT on DBN is justified in our case for two main reasons: 1) a JT is generally a more compact representation than an AC, and this is essential in the autonomous spacecraft domain, where the on-board resources are very limited also in terms of available memory and 2) the JT approach to DBN infer-ence allows to analyze (before the final on-board deployment of the inference engine) the suitability of approximate infer-ence, by changing the parameter of the BK algorithm and by trading-off inference speed and accuracy on simulated scenar-ios. Moreover, in our approach we can naturally deal with soft evidence and with different kinds of faults.

Finally, in [31] a system health management approach to autonomous spacecraft is proposed, again by compiling BN into ACs, with the goal of fault detection and fast on-board diagnosis. The work does not approach the whole FDIR cycle for autonomous spacecraft, since imminent faults and recovery problems are not addressed.

In conclusion, the framework we have investigated in the VERIFIM project has produced a proof of concept case study involving the definition of autonomous FDIR strategies for a Mars rover. We have discussed the resulting on-board software architecture, called ARPHA, and showed how the inference capabilities of DBNs can be suitably applied to deal with failure scenarios that need autonomous (preventive as well as reactive) recovery, under partial observability of both sys-tem behavior and environment

ISSN : 2454-9924

REFERENCES

[1] R. McDermott, R. J. Mikulak, and M. Beauregard, *The Basics of FMEA*. New York, NY, USA: Productivity Press, 1996.

[2] W. G. Schneeweiss, *The Fault Tree Method*. Bundesanzeiger, Germany: LiLoLe Verla

[3] M. Schwabacher, M. Feather, and L. Markosian, "Verification and validation of advanced fault detection, isolation and recovery for a NASA space system," in *Proc. ISSRE*, Washington, DC, USA, 2008.

[4] P. Robinson *et al*., "Applying model-based reasoning to the FDIR of the command and data handling subsystem of the ISS," in *Proc. i-SAIRAS*, Nara, Japan, 2003.

[5] W. Glover, J. Cross, A. Lucas, C. Stecki, and J. Stecki, "The use of PHM for autonomous unmanned systems," in *Proc. PHM Soc.*, Portland, OR, USA, 2010.

[6] U. Kjaerulff, "A computational scheme for reasoning in dynamic probabilistic networks," in *Proc. Int. Conf. UAI*, Stanford, CA, USA, 1992, pp. 121–129.

[7] K. Murphy, "Dynamic Bayesian networks: Representation, inference, and learning," Ph.D. Thesis, UC Berkley, Berkeley, CA, USA, 2002 [Online]. Available: http://www.cs.ubc.ca/~murphyk/Thesis/thesis.html

[8] D. Codetta-Raiteri, L. Portinale, S. D. Nolfo, and A. Guiotto, "ARPHA: A software prototype for fault detection, identification and recovery in autonomous spacecrafts," *Acta Futura*, vol. 5, no. 15, pp. 99–110, Jan. 2012.

[9] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[10] A. Bobbio, L. Portinale, M. Minichino, and E. Ciancamerla, "Improving the analysis of dependable systems by mapping fault trees into Bayesian networks," *Reliab. Eng. Syst. Safe.*, vol. 71, no. 3, pp. 249–260, 2001.

[11] H. Langseth and L. Portinale, "Bayesian networks in reliability," *Reliab. Eng. Syst. Safe.*, vol. 92, no. 1, pp. 92–108, 2007.

[12] S. Montani, L. Portinale, A. Bobbio, and D. Codetta-Raiteri, "RADYBAN: A tool for reliability analysis of dynamic fault trees through conversion into dynamic Bayesian networks," *Reliab. Eng. Syst. Safe.*, vol. 93, no. 7, pp. 922–932, 2008.

[13] P. Weber, G. Medina-Oliva, C. Simon, and B. Iung, "Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas," *Eng. Appl. Artif. Intell.*, vol. 25, no. 4, pp. 671–682, 2012.

[14] L. Portinale, "Bayesian belief networks in reliability: Advanced tutorial," in *Proc. RAMS*, 2012.

[15] A. Nicholson and J. Brady, "Dynamic belief networks for discrete monitoring," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 11, pp. 1593–1610, Nov. 1994.

[16] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes," in *Proc. Int. Conf. UAI*, San Francisco, CA, USA, 1998, pp. 33–42.

[17] L. Kallenberg, "Finite state and action Markov decision processes," in *Handbook of Markov Decision Processes: Methods and Applications*. Berlin, Germany: Springer, 2002, pp. 21–87.

[18] T. Dean and K. Kanazawa, "A model for projection and action," in *Proc. IJCAI*, 1989, pp. 985–990.

[19] S. Russell, "Learning agents for uncertain environments," in *Proc. COLT*, Madison, WI, USA, 1998, pp. 101–103.

[20] J. B. Dugan, S. Bavuso, and M. Boyd, "Dynamic fault-tree models for fault-tolerant computer systems," *IEEE Trans. Reliab.*, vol. 41, no. 3, pp. 363–377, Sep. 1992.

[21] D. Codetta-Raiteri, S. Montani, and L. Portinale, "Supporting reliability engineers in exploiting the power of dynamic Bayesian networks," *Int. J. Approx. Reason.*, vol. 51, no. 2, pp. 179–195, 2010.

[22] D. Codetta-Raiteri, L. Portinale, S. D. Nolfo, A. Guiotto, and Y. Yusthein, "A unified modeling and operational framework for fault detection, identification and recovery in autonomous spacecrafts," in *Proc. Res. Use Multiformalism Model. Meth.*, London, U.K., 2012, pp. 24–31.

[23] C. Huang and A. Darwiche, "Inference in belief networks: A procedural guide," *Int. J. Approx. Reason.*, vol. 15, no. 3, pp. 225–263, 1996.

[24] A. Doucet, N. De Freitas, K. P. Murphy, and S. J. Russell, "Rao-blackwellised particle filtering for dynamic Bayesian networks," in *Proc. Int. Conf. UAI*, Stanford, CA, USA, 2000, pp. 176–183.

[25] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian fault detection and diagnosis in dynamic systems," in *Proc. Nat. Conf. Artif. Intell.*, Austin, TX, USA, 2000, pp. 531–537.

[26] P. Mosterman and G. Biswas, "Monitoring, prediction and fault isolation in dynamic physical systems," in *Proc. Nat. Conf. Artif. Intell.*, Providence, RI, USA, 1997, pp. 100–105.

[27] M. Daigle *et al*., "Comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 5, pp. 917–931, Sep. 2010.

[28] P. Mosterman and G. Biswas, "A theory of discontinuities in physical system models," *J. Franklin Inst.*, vol. 335, no. 3, pp. 401–439, 1998.

[29] O. Mengshoel *et al*., "Probabilistic model-based diagnosis: An electrical power system case study," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 5, pp. 874–885, Sep. 2010.

[30] A. Darwiche, "A differential approach to inference in Bayesian networks," *J. ACM*, vol. 50, no. 3, pp. 280–305, 2003.

[31] J. Schumann, T. Mbaya, and O. Mengshoel, "Software and system health management for autonomous robotics missions," in *Proc. i-SAIRAS*, Torino, Italy, 2012.

Daniele Codetta-Raiteri received the Ph.D. degree in computer science from the University of Torino, Torino, Italy, in 2006.

He is an Assistant Professor of Computer Science with the University of Piemonte Orientale, Alessandria, Italy. He researched on probabilistic models for dependability and reliability analysis, with a particular experience in fault trees, Petri nets, Bayesian networks, and multiformalism mod-eling. He has authored over 50 papers published in journals, books, and proceedings.

Luigi Portinale (M'11) received the M.Sc. and Ph.D. degrees in computer science from the University of Torino, Torino, Italy, in 1988 and 1994, respectively.

He is a Full Professor of Computer Science and the Head of the Computer Science Institute with the University of Piemonte Orientale, Alessandria, Italy. His current research interests include the arti-ficial intelligence field, with particular attention to case-based reasoning and probabilistic uncertain reasoning for dependability and reliability applications. He has authored over 120 publications on the above topics.

Prof. Portinale is a member of the Italian Association for AI and the Association for the Advancement of AI.