Predictive and diagnostic analysis of an holdup tank by means of Dynamic Bayesian Networks

Daniele Codetta-Raiteri[1], **Luigi Portinale**[1]
[1]DiSIT, Computer Science Institute, University of Piemonte Orientale, Alessandria, Italy

Abstract

In dynamic reliability evaluation, the complete behaviour of the system has to be taken into account. In this paper, a benchmark taken from the literature is examined. To this aim, we exploit Dynamic Bayesian Networks (DBN) extending standard Bayesian networks by introducing a discrete temporal dimension. The goals are the prediction of the system unreliability and the computation of diagnostic indices. Because of the achievement of such goals, we propose DBN to be a valid approach for dynamic reliability evaluation.

1. Introduction

We talk about dynamic reliability [1] when the system configuration changes during the mission time. In these cases, we may have to consider the whole system behaviour in order to evaluate the reliability. This means modelling the normal functioning of the system, the occurrence of component failure events and their effect on the system functioning. Combinatorial models [2] such as *Fault Trees* and *Reliability Block Diagrams* can only represent combinations of component failure events assumed to be independent. The complete system behaviour can be represented by means of state space based models [2], such as *Markov Chains* or *Petri Nets*. They rely on the specification of the whole set of the possible system states, so that the stochastic behaviour of each component may depend on the state of all the other components. However, their use in dynamic reliability may determine the state space explosion making the model analysis unpractical because of the high computing cost (and time).

Bayesian Networks (BN) [3] are an interesting trade-off between combinatorial and state space based models; in particular, *Dynamic Bayesian Networks* (DBN) [4] provide an explicit discrete temporal dimension: a DBN represents the system at several discrete time slices, and conditional dependencies among variables at different slices are introduced to capture the temporal evolution. Both BN and DBN have been recently investigated as very promising formalisms for dependability and reliability analysis [3, 5, 6]. We argue that DBN are a possible and suitable approach to examine dynamic reliability cases; we show this point by investigating the analysis of a specific benchmark taken from the literature [1]. The benchmark is a system consisting of a tank containing some liquid whose level is influenced by a controller commanding two pumps and one valve, with the aim of avoiding the dry out or overflow of the liquid. In the past, the benchmark was evaluated by means of Monte Carlo simulation [1] and Petri Nets [7, 8]. In this paper, the system is modelled as a DBN, with the purpose of computing the system unreliability (the original goal of the benchmark [1]), and diagnostic indices which are an additional possibility offered by DBN.

The paper is organized as follows: Section 2 contains the related work about this benchmark; Section 3 describes the system behaviour; Section 4 provides the essential notions about the DBN formalism; Section 5 describes the DBN model of the system; finally, Section 6 reports the results of the model analysis.

2. Related work

The benchmark is specified in [1], where the system unreliability is evaluated by means of the Monte Carlo simulation. In [7] the benchmark is modelled as a *Generalized Stochastic Petri Net* (GSPN) [2]. The GSPN model can undergo analysis, but it suffers from two approximations: **1)** the liquid level is discretized into few intermediate levels, because only discrete variables can be represented as the number of tokens (marking) inside places; **2)** some deterministic timed events such as the action of the pumps on the liquid level, are considered as stochastic events. Still in [7], the benchmark is modelled and simulated as a *Fluid Stochastic Petri Net* (FSPN) [9] including also fluid places which directly represent continuous variables such as the liquid level in the tank. Finally, in [8], *Stochastic Activity Networks* (SAN) [10] are applied in order to model and simulate the benchmark. SAN extend Petri Nets introducing input or output gates able to express complex conditions and effects about the firing of transitions, compacting the model as a consequence. SAN can represent float variables by means of extended places.

In [1], other versions of the benchmark are presented and are characterized by particular features (state dependent failure rates, failure on demand, repair). They are evaluated using Petri Net based approaches in [7, 8], and using DBN in [11].

3. The case study

The system (Figure 1.a) is composed by a tank containing liquid, two pumps (P1, P2) to fill the tank, one valve (V) to remove liquid, and the controller (C) monitoring the liquid level (H) and switching P1, P2, V on or off. The state of P1, P2, V can be ON, OFF, Stuck ON (S_ON), or Stuck OFF (S_OFF). Initially H is equal to 0, with P1 and V in state ON, and P2 in state OFF; since both pumps and the valve have the same level variation rate (Q=0.6 m/h), H does not change while the initial configuration holds (Tab. 1). The cause of a variation of H may be the occurrence of a component failure during the ON or OFF state. The failure probability obeys the negative exponential distribution: the failure rate λ of P1, P2 and V is equal to 0.004566 h^{-1}, 0.005714 h^{-1} and 0.003125 h^{-1}, respectively. The effect of the failure is the stuck condition, while the state transitions toward S_ON and S_OFF, are uniformly distributed (Figure 1.b).

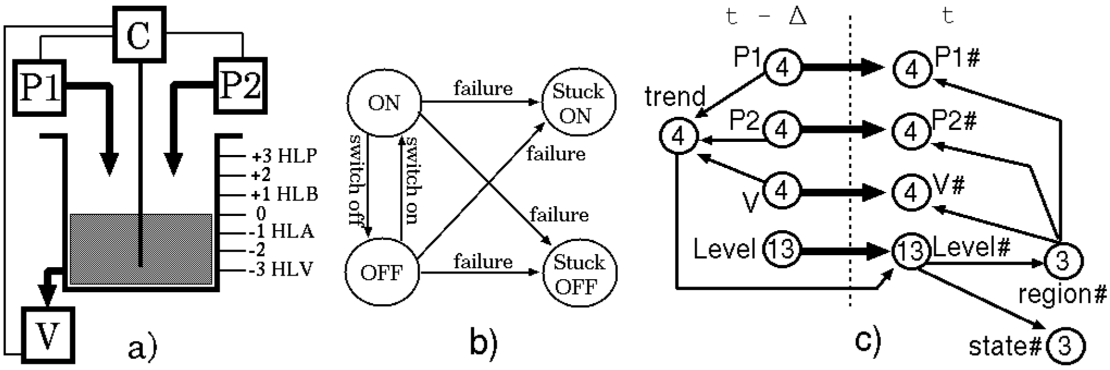

Figure 1: **a)** The system scheme. **b)** The possible states of P1, P2 and V. **c)** The graph of the DBN model.

Comp. states				Comp. states				Comp. states			
P1	P2	V	effect	P1	P2	V	effect	P1	P2	V	effect
OFF	OFF	OFF	=	OFF	ON	ON	=	ON	ON	OFF	↑↑
OFF	OFF	ON	↓	ON	OFF	OFF	↑	ON	ON	ON	↑
OFF	ON	OFF	↑	ON	OFF	ON	=				

Table 1: The effect on H in each state configuration (S_ON and S_OFF have the same effect of ON and OFF respectively).

Tab. 1 shows how H changes with respect to the current configuration of the component states; C believes that the system is correctly functioning while H is inside the region between the levels denoted by HLA (-1 m) and HLB (+1 m) shown in Figure 1.a. If H reaches HLA, then C orders to P1 and P2 to switch on, and V to switch off (**order n. 1**), with the aim of increasing H and avoiding the **dry out**; this event occurs when H reaches the level denoted as HLV (-3 m). If a component is stuck, it does not obey the controller order and maintains its current state. The other system failure condition is the **overflow**; this happens when H reaches HLP (+3 m). If H reaches HLB, C orders to P1 and P2 to switch off, and V to switch on (**order n. 2**), with the aim of decreasing H and avoiding the overflow.

4. Basic notions about DBN

BN are defined by a directed acyclic graph (DAG) in which nodes correspond to discrete random variables having a conditional dependence on the parent nodes. DBN extend BN by providing a discrete temporal dimension. The advantage with respect to a classical probabilistic temporal model like *Markov Chains*, is that a DBN is a stochastic transition model factored over a number of random variables. While a DBN can in general represent semi-Markovian stochastic processes of order $k-1$, providing the modelling for k time slices, the term DBN is usually adopted when $k=2$. If so, the Markovian assumption holds and only 2 time slices are considered in order to model the system temporal evolution: the slice at time t depends only on the previous slice at $t - \Delta$, and is conditionally independent of the past ones (Δ is the time discretization step). An example of DBN is shown in Figure 1.c.

In a DBN, we can distinguish between two kinds of arcs: intra-slice and inter-slice arcs establishing dependencies between variables in the same time slice, and dependencies between variables in different time slices, respectively. The variables characterized by a temporal evolution, have two instances, one for each time slice. connected by a temporal arc graphically appearing as a thick line. Inter-slice arcs connecting two instances of a variable are called *temporal arcs*. For instance, in Figure 1.c, $P1$ and $P1\#$ are present in the time slices $t - \Delta$ and t respectively, and are connected by the temporal arc $(P1, P1\#)$. This means that P1 may change its state during the interval Δ between two consecutive time steps.

The dependencies of a certain DBN node are quantified in terms of conditional probabilities and are stored in its *Conditional Probability Table* (CPT). The probability in every CPT entry has to be set according to the state of the parent nodes (possibly including the other instance of the node).

Analysis. Let X^t be a set of variables at time t and $y_{a:b}$ any stream of observations between the time points a and b (a set of instantiated variables Y_i^j with $a \leq j \leq b$). The following tasks can be performed over a DBN:
• *Prediction*: computing $P(X^{t+h}|y_{a:b})$ for some horizon $h > 0$, i.e. predicting a future state taking into consideration the observation up to now (if $h = 0$ the task is more properly called **Filtering** or **Monitoring**);
• *Smoothing*: computing $P(X^{t-l}|y_{a:b})$ for some $l < t$, i.e. estimating what happened l steps in the past, given all the evidence (observations) up to now.

In this work, the DBN model is designed and analyzed by means of the RADYBAN software tool [12]. In particular, for the analysis, we resort to the *Junction Tree* (JT) algorithm based on the construction of a classical BN inference data structure called junction or join tree [4]. The JT algorithm returns exact results for both the above tasks.

5 . Modelling the benchmark

State of components. The DBN model of the benchmark is shown in Figure 1.c. The state of P1 is represented by the variables $P1$ and $P1\#$. Their value can be 0, 1, 2, or 3, in order to represent the states OFF, ON, S_OFF, S_ON, respectively. The variable $P1$ in the time slice $t - \Delta$ does not depend on any other variable ($P1$ is a root node). Therefore its CPT (Section 4) simply provides the initial probability distribution of the four possible values. In particular, the value 1 has probability 1 in order to express that the initial state of P1 is ON. The variable $P1\#$ in the time slice t depends on $P1$ and $region\#$ in order to model that the current state depends on the state in the previous time step, and on the current region determining the command currently provided by C

(Section 3). So, the variable $region\#$ is ternary: the value 0 corresponds to the order n. 1, 1 represents the absence of orders (correct region), 2 corresponds to the order n. 2 (Section 3). The CPT of $P1\#$ (Tab. 2) contains the probability distribution of the possible values of $P1\#$ given all the possible combinations of the values of $P1$ and $region\#$. Let us consider the entries of this CPT:

- the **entry n. 1** is the case where $P1 = 0$ and $region\# = 0$; this means that P1 was OFF in the previous time step and C orders the pumps to switch on in the current time. So, the probability that $P1\# = 0$ (P1 is currently OFF) is null because P1 can not ignore the order if it is not stuck. The probability that $P1\# = 1$ (P1 is currently ON) is the probability that P1 does not fail (reliability of P1) during the transition between the time slices $t - \Delta$ and t, according to the negative exponential distribution, the rate λ (Section 3) and the time step Δ (the value of Δ will be specified in the following). The probability that $P1\# = 2$ (P1 is currently S_OFF) is half of the probability of failure, because the probability to turn S_ON or S_OFF after the failure, is uniformly distributed (Section 3). The probability that $P1\# = 3$ (P1 is currently S_ON) is computed in the same way. The sum of the probabilities in the entry n. 1 and in the following entries has to be 1. For the sake of brevity, the lines characterized by null probability are omitted in the CPTs.
- The **entry n. 2** is the case where P1 was OFF and C provides no order. Therefore $Pr\{P1\# = 1\}$ is null, while $Pr\{P1\# = 0\}$ is the reliability of P1 during the time step Δ. $Pr\{P1\# = 2\}$ and $Pr\{P1\# = 3\}$ are computed in the same way as in the entry n. 1 and the following ones, up to entry n. 6.
- The **entry n. 3** is the situation where P1 was OFF and C orders the pumps to switch off. So, $Pr\{P1\# = 0\}$ is the reliability of P1 during the time step Δ, while $Pr\{P1\# = 1\}$ is null because of the order from C.
- In the **entry n. 4**, P1 was ON and the command is to turn ON. Therefore $Pr\{P1\# = 0\}$ is null, while $Pr\{P1\# = 1\}$ is the probability that P1 does not fail (reliability).
- In the **entry n. 5**, P1 was ON and no commands are provided, so the same probability distribution as in the entry n. 4, holds.
- The **entry n. 6** is the case where P1 was ON and C orders the pumps to switch off. Therefore $Pr\{P1\# = 0\}$ is the component reliability, while $Pr\{P1\# = 1\}$ is equal to 0.
- In the **entries n. 7, 8, 9**, P1 was in the S_OFF state ($P1 = 2$) in the previous time step. Since P1 is not repairable, P1 maintains such state in the current time step, ignoring any command from C. Therefore $Pr\{P1\# = 2\}$ is equal to 1 in all the entries, while the probabilities of the other values are null.
- In the **entries n. 10, 11, 12**, P1 was S_ON ($P1 = 3$), so $Pr\{P1\# = 3\}$ is equal to 1 in all such entries.

The states of P2 are modelled in the same way by the variable $P2\#$ depending on $P2$ and $region\#$. The states of V are represented by the variable $V\#$ influenced by V and $region\#$. The CPT of $V\#$ takes into account the opposite reactions of V to the orders, and the failure rate λ of V (Section 3).

Variations to H. The variable $trend$ depends on the variables $P1$, $P2$ and V, and its value can vary between 0 and 3. The role of this variable is to represent

n.	$P1$	$region\#$	***P1#***	prob.	n.	$P1$	$region\#$	***P1#***	prob.
	0	0	1	$e^{\lambda\Delta}$		1	1	1	$e^{-\lambda\Delta}$
1	0	0	2	$(1-e^{-\lambda\Delta})/2$	5	1	1	2	$(1-e^{-\lambda\Delta})/2$
	0	0	3	$(1-e^{-\lambda\Delta})/2$		1	1	3	$(1-e^{-\lambda\Delta})/2$
	0	1	0	$e^{\lambda\Delta}$		1	2	0	$e^{-\lambda\Delta}$
2	0	1	2	$(1-e^{-\lambda\Delta})/2$	6	1	2	2	$(1-e^{-\lambda\Delta})/2$
	0	1	3	$(1-e^{-\lambda\Delta})/2$		1	2	3	$(1-e^{-\lambda\Delta})/2$
	0	2	0	$e^{\lambda\Delta}$	7	2	0	2	1
3	0	2	2	$(1-e^{\lambda\Delta})/2$	8	2	1	2	1
	0	2	3	$(1-e^{\lambda\Delta})/2$	9	2	2	2	1
	1	0	1	$e^{-\lambda\Delta}$	10	3	0	3	1
4	1	0	2	$(1-e^{-\lambda\Delta})/2$	11	3	1	3	1
	1	0	3	$(1-e^{-\lambda\Delta})/2$	12	3	2	3	1

Table 2: The CPT of $P1\#$ and $P2\#$.

the four possible effects on H according to the current state of P1, P2 and V, as specified in Tab. 1. In particular, the value 0 represents the decrease of H, 1 represents the steadiness of H, 2 models the slow growth of H, 3 models the quick growth of H. There is no difference between the states OFF and S_OFF, or ON and S_ON, in terms of effects on H. The CPT of $trend$ reflects the content of Tab. 1. For instance, the first entry specifies that if all the components P1, P2 and V are OFF, then H is steady ($trend = 1$) with probability 1.

A DBN can represent discrete quantities in terms of the values of variables. H is a continuous measure to be discretized in order to be modelled by a DBN variable. On one hand, a low number of discrete intermediate levels may lead to some approximation of the inference results. On the other hand, a high number may increase in a relevant way the size of several CPTs and as a consequence, the complexity of the model analysis. In order to achieve a good trade-off between accuracy and complexity, in the DBN we discretize H into 13 intermediate levels. To this aim, we exploit the variable $Level$ whose value can vary between 0 and 12. This means that the distance between an intermediate level and the following one is 0.5 m: Tab. 3 defines the correspondence between the 13 values of $Level$ and the effective liquid level in the tank. Given that two consecutive intermediate values differ by 0.5 m, in the DBN we can represent the variation of H for the same quantity by increasing or decreasing $Level$ by one unit. If the variation rate for P1, P2 and V is Q=0.6 m/h (Section 3), then a variation of H by 0.5 m (1 unit for $Level$) due to the action of a single component, takes 0.8333 h of time. We set the time discretization step Δ to this value in such a way that $Level$ may change by 1 during one time step. The parameter Δ is used in the CPTs of $P1\#$ (Tab. 2), $P2\#$, $V\#$ to compute the component (un)reliability.

In the DBN, the variable $Level\#$ (current H) depends on $Level$ (H in the previous time step) and on $trend$ (the effect due to current state of P1, P2 and V). In particular, with respect to the value of $Level$, the value of $Level\#$ is the same if $trend = 1$, is decreased by 1 if $trend = 0$, is increased by 1 if $trend = 2$, or by 2 if $trend = 3$. All of this is specified in the CPT of $Level\#$ (Tab. 4).

$Level$	actual level	region	$Level$	actual level	region	$Level$	actual level	region
12	+3.0 m	2	7	+0.5 m	1	2	-2.0 m	0
11	+2.5 m	2	6	+0.0 m	1	1	-2.5 m	0
10	+2.0 m	2	5	-0.5 m	1	0	-3.0 m	0
9	+1.5 m	2	4	-1.0 m	0			
8	+1.0 m	2	3	-1.5 m	0			

Table 3: The values of $Level$ and the corresponding intermediate liquid levels (Figure 1.a).

$Level$	$trend$	***Level#***	prob.	$Level$	$trend$	***Level#***	prob.
0	0	0	1	7	0	6	1
0	1	0	1	7	1	7	1
0	2	0	1	7	2	8	1
0	3	0	1	7	3	9	1
1	0	0	1	8	0	7	1
1	1	1	1	8	1	8	1
1	2	2	1	8	2	9	1
1	3	3	1	8	3	10	1
2	0	1	1	9	0	8	1
2	1	2	1	9	1	9	1
2	2	3	1	9	2	10	1
2	3	4	1	9	3	11	1
3	0	2	1	10	0	9	1
3	1	3	1	10	1	10	1
3	2	4	1	10	2	11	1
3	3	5	1	10	3	12	1
4	0	3	1	11	0	10	1
4	1	4	1	11	1	11	1
4	2	5	1	11	2	12	1
4	3	6	1	11	3	12	1
5	0	4	1	12	0	12	1
5	1	5	1	12	1	12	1
5	2	6	1	12	2	12	1
5	3	7	1	12	3	12	1
6	0	5	1				
6	1	6	1				
6	2	7	1				
6	3	8	1				

Table 4: The CPT of $Level\#$.

The entries with $Level = 0$ and $Level = 12$ correspond to the dry out and the overflow respectively; the probabilities in such entries express the assumption that H does not change any more if a system failure condition is reached.

H can be inside one of three regions: H≤HLA, HLA<H<HLB, H≥HLB (Section 3). The variable $Level\#$ influences $region\#$ whose value can be 0, 1, or 2 in order to represent the above three regions respectively, and the corresponding commands (Section 3). The CPT of $region\#$ maps the values of $Level\#$ into the corresponding value of $region\#$ according to Tab. 3.

System states. We need the ternary variable $state\#$ to model the three possible states of the system: working, dry out or overflow. In particular, the working state is any situation where the dry out or the overflow has not occurred yet. These states are determined by H, so $state\#$ is influenced by $Level\#$. The value 0 of $state\#$ represents the dry out, the value 1 models the working state, and 2 indicates the overflow. In the CPT of $state\#$, we set this variable to 0 only when $Level\# = 0$ and we set it to 2 only when $Level\# = 12$, according to Tab. 3. In any other case, $state\#$ is set to 1. Since $Level\#$ does not change any more its value in case of dry out or overflow (as described above), $state\#$ maintains its value if set to 0 or 2.

6. Model analysis

Predictive results. First, we compute the *cumulative distribution function* for the dry out (cdf_{dry}) and the overflow (cdf_{ov}). The value of cdf_{dry} and cdf_{ov} at time t is the probability that the system has failed because of the dry out and overflow, respectively, during the time period $(0, t)$. In other words, cdf_{dry} and cdf_{ov} are the system unreliability because of the dry out and the overflow, respectively. These measures can be computed on the DBN models by means of the filtering task with an empty stream of observations (Section 4).

As in [1], the system is evaluated for a mission time varying between 0 and 1000 h. In DBN, the time is discrete (Section 4), and two consecutive time steps differ by the interval Δ which is set to 0.8333 h in the DBN models of the benchmark (Section 5). So, in order to evaluate the system from 0 to 1000 h, we have to inference the model from 0 to 1200 time steps. For example, the system evaluation at 400 h is given by the DBN analysis at 480 time steps (480 = 400 h / 0.8333 h).

At each time step, the variable $state\#$ is queried to obtain the probability distribution of its values 0, 1, 2, corresponding to the dry out, working, and overflow condition, respectively (Section 5). So, the probability that $state\#$ is equal to 0 at time t, provides cdf_{dry} at that time, while cdf_{ov} is given by the probability that $state\#$ is equal to 2.

The results returned by DBN analysis are quite similar to those obtained by the techniques described in Section 2, as shown in Tab. 5 (cdf_{dry}) and in Tab. 6

time	**step**	**DBN an.**	SAN sim.	GSPN an.	FSPN sim.
200 h	240	2.2789E-2	2.2390E-2	2.2077E-2	2.400E-2
400 h	480	6.6455E-2	6.5990E-2	6.5827E-2	6.730E-2
600 h	720	9.5366E-2	9.5290E-2	9.5014E-2	9.360E-2
800 h	960	1.1040E-1	1.1003E-1	1.1022E-2	1.084E-1
1000 h	1200	1.1777E-1	1.1747E-1	1.1768E-2	1.165E-1

Table 5: The cumulative distribution function for the dry out (cdf_{dry}).

time	**step**	**DBN an.**	SAN sim.	GSPN an.	FSPN sim.
200 h	240	1.9890E-1	1.9914E-1	1.9518E-1	2.0050E-1
400 h	480	3.6172E-1	3.6207E-1	3.5987E-1	3.6220E-1
600 h	720	4.3652E-1	4.3665E-1	4.3568E-1	4.4160E-1
800 h	960	4.6997E-1	4.7063E-1	4.6959E-1	4.7630E-1
1000 h	1200	4.8538E-1	4.8572E-1	4.8520E-1	4.9100E-1

Table 6: The cumulative distribution function for the overflow (cdf_{ov}).

(cdf_{ov}). This verifies that DBN analysis generates results with a good degree of accuracy. The differences in the results are due to the model evaluation approach (analysis or simulation), the modelling power of each formalism (DBN, SAN, GSPN, FSPN), and the assumptions holding in the model (Section 2). For instance, the DBN, the GSPN, and the SAN model capture variations of H by 0.5 m, 1 m and 0.01 m respectively, while H is a continuous variable in the FSPN. The DBN and the GSPN model undergo analysis, while the SAN and FSPN model are simulated.

Diagnostic results. DBN can be exploited to compute measures conditioned by observations (Section 4). In this case, the difference between a filtering and a smoothing inference (Section 4) relies on the fact that in the former case, while computing the probability at time t, only the evidence (observations) gathered up to time t is considered; on the contrary, in the case of smoothing the whole evidence stream is always considered in the posterior probability computation. For diagnosis purposes, filtering can be exploited to perform the on line diagnosis of the system. This means evaluating the state of components during the monitoring of the system behaviour. For instance, in our case study, if we assume that the value of H is observable at each time step t, then we can compute the probability of each possible state of P1, P2 and V at t. In this way, we can estimate the causes of the current value of H. Smoothing instead, may be exploited in order to reconstruct the history of the system components for a kind of temporal diagnosis. For instance, we may be interested in evaluating the probability of each state of P1, P2 and V at each time step, based on the observations about H, collected during all the system mission time. These kinds of measures were not computed in the previous works about the benchmark (Section 2). They are an additional value given by DBN.

In order to clarify these concepts, we provide an example of filtering and smooth-

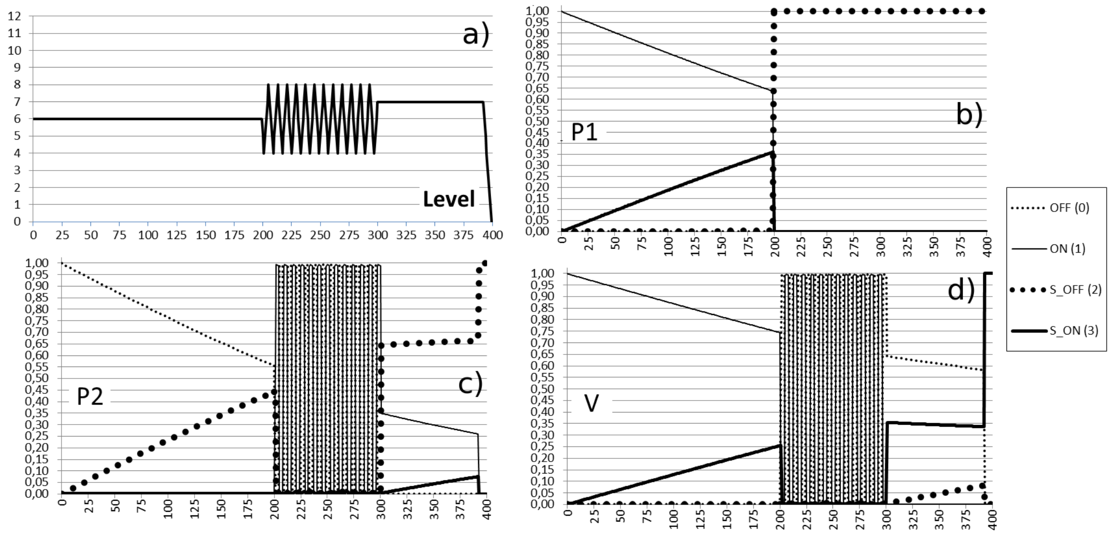

Figure 2: The observations about $Level$ and the diagnostic results given by the **filtering** task.

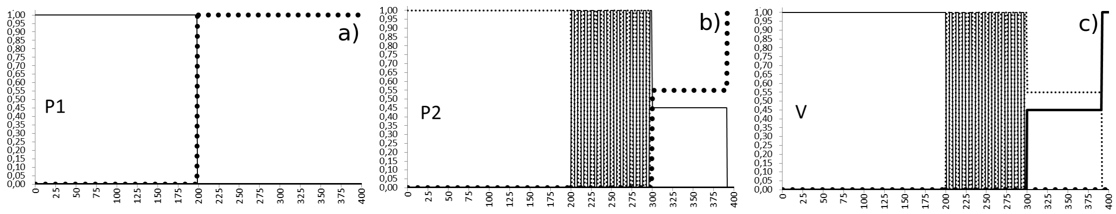

Figure 3: The diagnostic results given by the **smoothing** task.

ing applied to the DBN model, assuming to observe the value of the variable $Level$ at each time step; $Level$ represents H in the model (Tab. 3). The progress of $Level$ during the time is depicted in Figure 2.a. In order to detect if current H is due to a particular state of P1, P2 or V, we perform the filtering task on the DBN model, querying the variables $P1\#$, $P2\#$ and $V\#$, with the aim of computing the probabilities of their possible values (0, 1, 2, 3) corresponding to the possible states (OFF, ON, S_OFF, S_ON, respectively) of the components (Section 5).

The **filtering** results are depicted in Figure 2. $Level$ is observed steady to 6 from time step 0 to time step 199. So, initially, P1 is ON, P2 is OFF and V is ON with probability 1. This configuration is still possible from 1 to 199, but there is an increasing probability of S_ON for P1 and V, and S_OFF for P2, due to the observed steadiness of $Level$ and the failure rate of the components.

At time step 200, $Level$ becomes 5. The filtering results indicate that the cause of this decrease is certainly the S_OFF state of P1. At 201, $Level$ reaches 4 (H=HLA), so we expect an order from C with the aim of switching on the pumps and switching off V. Actually in this time step, P2 turns to ON, while V turns to OFF, with probability 1. $Level$ grows from 4 to 8 during the time steps from 201 to 205. In particular, at 205, due to $Level = 8$ (H=HLB), C should order the pumps to switch off, and V to switch off. This is confirmed by the probabilities of P2 to be OFF and of V to be ON, both equal to 1 in this time step. This configuration of the components leads to a decrease of $Level$ from 8 to 4 during the time steps from 205 to 209. At 209, $Level$ is equal to 4 (H=HLA), so C successfully sets P2 to ON and V to OFF according to the filtering results. This configuration will determine another growth of $Level$ up to 8 (H=HLB) with another inversion of the states of P2 and V to decrease $Level$ again. This fluctuation of $Level$ lasts until the time step 300 and is reflected by the alternation of the states ON and OFF for P2 and V, during this time.

At 301, we observe that $Level$ is equal to 7, the same value observed at 300. In other words, $Level$ has interrupted its growing stage maintaining its value. The filtering results provide two alternative causes for this event, with different probabilities: P2 is S_OFF or V is S_ON. From 302 to 392 $Level$ maintains the value 7; because of this, during these steps, P2 may be ON or S_ON (combined with V in S_ON state), or S_OFF (combined with V in OFF or S_OFF); V instead, may be OFF or S_OFF (assuming that P2 is S_OFF), or S_ON (assuming that P2 is ON or S_ON). At 393, $Level$ becomes 6 (H is decreasing). The filtering task deduces that the certain cause is the contemporary S_OFF state of P1 and the S_ON state of V. This is confirmed in the next steps leading to the dry out.

The **smoothing** results for Scenario 2 (Figure 3) show a more precise diagnosis: the anticipated knowledge about the values of $Level$ excludes the possibility that P1, P2 or V may be S_OFF or S_ON between the time steps 1 and 199. Their state is certain during that period. The diagnosis between the time steps 200 and 300 is confirmed; the probability that P2 is S_ON and the probability that V is S_OFF between 301 and 392, are both null in the smoothing task results, while such states were possible according to the filtering output (Figure 2).

7. Conclusions

A benchmark on dynamic reliability taken from the literature has been evaluated. The predictive results about the system unreliability that we obtained, are in general quite similar to those computed by means of other techniques. This proposes DBN as suitable models to deal with dynamic reliability cases, with two main advantages: **1)** with respect to state space based models, the use of a DBN takes advantage of factorization of the system state space into the model variables. **2)** DBN introduce the possibility of computing measures conditioned by observations, at specific times. This has been applied in order to compute diagnostic measures about the state of components.

References

[1] Marseguerra, M. and Zio, E., Monte Carlo Approach to PSA for dynamic process system, *Reliability Engineering and System Safety,* **52** 227–241 (1996).

[2] Sahner, R., Trivedi, K. and Puliafito, A., *Performance and Reliability Analysis of Computer Systems*, Kluwer Academic Publisher (1996).

[3] Langseth, H. and Portinale, L., Bayesian Networks in reliability, *Reliability Engineering and System Safety*, **92** [1] 92–108 (2007).

[4] Murphy, K., *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, UC Berkley (2002).

[5] Weber, P. and Jouffe, L., Reliability modelling with dynamic Bayesian networks, in *Proceedings of the Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Washington DC, USA (2003).

[6] Codetta-Raiteri, D., Bobbio, A., Montani, S. and Portinale, L., A dynamic bayesian network based framework to evaluate cascading effects in a power grid, *Engineering Applications of Artificial Intelligence*, **25** [4] 683–697 (2012).

[7] Codetta-Raiteri, D. and Bobbio, A., Solving Dynamic Reliability Problems by means of Ordinary and Fluid Stochastic Petri Nets, in *Proceedings of the European Safety and Reliability Conference*, pp 381–389, Gdansk, Poland (2005).

[8] Codetta-Raiteri, D., Modeling and simulating a benchmark on dynamic reliability, as a Stochastic Activity Network, in *Proceedings of the European Modeling & Simulation Symposium*, pp 545–554, Rome, Italy (2011).

[9] Gribaudo, M., Sereno, M., Horvath, A. and Bobbio, A., Fluid Stochastic Petri Nets augmented with flush-out arcs: Modelling and analysis, *Discrete Event Dynamic Systems*, **11** [1/2] 97–117 (2001).

[10] Sanders, W. and Meyer, J., Stochastic activity networks: Formal definitions and concepts, *Lecture Notes in Computer Science*, **2090** 315–343 (2001).

[11] Codetta-Raiteri, D. and Portinale, L., Bayesian networks applied to dynamic reliability with predictive and diagnostic purposes, *to be submitted* (2013).

[12] Portinale, L., Bobbio, A., Codetta-Raiteri, D. and Montani, S., Compiling dynamic fault trees into dynamic Bayesian nets for reliability analysis: The Radyban tool, *CEUR Workshop Proceedings*, **268** (2007).