Sequential Dependency and Reliability Analysis of Embedded Systems

Hehua Zhang*, Yu Jiang†, Xiaoyu Song§, William N. N. Hung¶, Ming Gu*, and Jiaguang Sun* *
School of Software, Tsinghua university, TNList, KLIss, Beijing.
†Department of Computer Science and Technology, Tsinghua university, TNList, KLIss, Beijing.
§Dept. ECE, Portland State University, Oregon, USA.
¶Synopsys Inc., Mountain View, USA.

***Abstract*—Embedded systems are becoming increasingly popular due to their widespread applications and the reliability of them is a crucial issue. The complexity of the reliability analysis arises in handling the sequential feedback that make the system output depends not only on the present input but also the internal state. In this paper, we propose a novel probabilistic model, named sequential dependency model (SDM), for the reliability analysis of embedded systems with sequential feedback. It is constructed based on the structure of the system components and the signals among them. We prove that the SDM model is s Dynamic Bayesian Network (DBN) that captures: the spatial dependencies between system components in a single time slice, the temporal dependencies between system components of different time slices, and the temporal dependencies due to the sequential feedback. We initiate the conditional probability distribution (CPD) table of the SDM node with the failure probability of the corresponding system component. Then, the SDM model handles the spatial-temporal correlations at internal components as well as the higher order temporal correlations due to the sequential feedback with the computational mechanism of DBN, experiment results demonstrate the accuracy of our model.**

I. Introduction

Reliability analysis of embedded systems has become an important part of system life circle. This is especially true for systems performing critical applications such as nuclear power plants. The typical task for the reliability analysis is to build a statistical mathematically model representing the system through a set of random variables. Then, the distributions of these variables should be fully specified to calculate the quantities we are interested in.

Traditionally, the reliability of a system is defined as the duration of its mission in given conditions and the reliability assessment is usually realized by combinatorial methods such as Fault Tree (FT) [1] and Reliability Block Diagram (RBD) [2]. FT involves specifying a top event such as the failure of the system to analyze, followed by identifying all associated events that could lead to the top event. RBD is a graphical depiction of the system components and connectors. It is used to determine the overall system reliability when the reliability of each component is given. FT is inevitably the most widely used model for the reliability analysis, because it is easy to use, presents the designer with an high level abstraction of the system, and can be solved quickly using techniques such as Binary Decision Diagrams [3]. However, traditional FT represents just a logical relation and can not handle complex functional dependencies between internal system components. In order to overcome the lack of modeling power, analysts extend the traditional FT by associating a particular markov process to the leaf node [4]. Embedding markov process into the FT leaf node makes it easy for dynamic system modeling. These methods are easy to use. But they are not very useful to model the temporal correlation between internal components as well as the higher order temporal correlation due to the sequential feedback.

The drawbacks of those traditional methods lead to a more flexible modeling framework named Bayesian Network (BN). It is originated in the field of artificial intelligence. BN is based on the theory of graphical and probabilistic reasoning for handling probabilistic events. The causal network is an inference engine for the calculation of beliefs and the probability of events. The popularity grows among system reliability analysts and it has been applied in reliability analysis successfully in [5], [6]. Some comparisons between BN and FT in terms of the modeling and analysis capabilities are presented in [7], [8]. An extension of BN, named DBN, is also used for the reliability analysis. The extension is introduced to deal with the temporal correlations between time slices of the system components [9], while preserving the internal dependencies at each time slice.

However, all introduced work above can not handle the higher order correlation caused by the sequential feedback. For these reasons, we propose a novel probabilistic model, named SDM, over all the input signals, internal signals, output signals and the state variables, to model the joint reliability of the sequential embedded system. The joint reliability can be mapped onto a SDM model that preserves the underlying dependency model of the spatial-temporal correlations at internal components as well as the higher order temporal correlations due to the sequential feedback. The SDM construction is divided into three steps: (1) build a BN based on the underlying logical structure of the system, (2) extend the BN over n time slices to get a complete dependency model, and (3) delete some redundant dependency relation at the internal component and add the higher order temporal relation due to the sequential feedback to get the SDM model. We prove that the constructed

This research is supported in part by NSFC Programs (No.61202010, No.91218302), National Key Technologies R&D Program (No.SQ2012BAJY4052) and 973 Program (No.2010CB328003) of China.

978-1-4673-3030-5/13/$31.00 ©2013 IEEE

SDM modle is a DBN which is a minimal representation that captures all independency correlations in the sequential embedded system, and initiate the CPD table of the SDM node with the failure probability of the corresponding system component. Then, the SDM model supports both predictive and diagnostic inferences about the reliability properties with the inference algorithms used in DBN. Generally speaking, the main contributions of our work are:

- It is the first time that a model for reliability analysis including : the spatial dependencies between system components in a single time slice, the temporal dependencies between system components of different time slices, and the temporal dependencies due to the sequential feedback.
- We prove that the constructed SDM model is a minimal representation of the underlying dependency model of the spatial-temporal correlations as well as the higher order temporal correlations of signals, and hence is a DBN. we also define some CPD tables for the SDM model nodes, and initialize the CPD tables with the failure probabilities of the corresponding system components.

The organization of this paper is organized as follows. Some background on BN and DBN are introduced in Section II. The proposed SDM model construction and initialization are presented in Section III. We apply our framework to some sequential embedded systems in Section IV, and conclude our work in Section V.

II. BACKGROUND

Bayesian Network [10] is a directed probability graphical model, each node in the graph represents a random variable, and arcs connecting the relative variables express the conditional probabilistic independence among the variables. The formal definition of BN is the tuple $\langle U, E, P\rangle$:

- U is the set of nodes: $U = \{x_1, x_2, \cdots x_n\}$, x_i is the label of the node.
- E is the set of arcs: $E = \{e_{ij} | there\ is\ an\ arc\ from\ node\ x_i\ to\ x_j\}$, e_{ij} is the label of the arc.
- P is the set of the CPD: $P = \{f(x_i|parent(x_i))\}$, $parent(x_i)$ denotes the parents of node x_i, $f(x_i|parent(x_i))$ is the conditional distribution of variable x_i given all its parents.

The first two items make up a directed acyclic graph (DAG), which is the qualitative part of BN. The qualitative part is used to encode the conditional independence statements of a multivariate statistical distribution, representing the assertion that a variable is conditionally independent of its non-descendants given its parents. The third item is the quantitative part of BN. Variables are described by a set of CPDs over their parents, which in turn defines the directed acyclic graph. The conditional independencies embedded among those random variables are embedded into the CPD. The CPD allows us to calculate the joint probability function in a simplified form (formula 2) compared to the original form (formula 1), where $f(x_1, x_2 \cdots x_n)$ is the joint distribution of those variables and $f(x_n|x_{n-1} \cdots x_1)$ is the distribution of x_n given all the other variables:

$$f(x_1, x_2 \cdots x_n) = f(x_n|x_{n-1} \cdots x_1) \cdot f(x_{n-1}|x_{n-2} \cdots x_1) \cdots f(x_1) \quad (1)$$

$$f(x_1, x_2 \cdots x_n) = \prod_{i=1}^{n} f(x_i|parent(x_i)) \quad (2)$$

Dynamic Bayesian Network [11], [12] is a generalization of BN to address the stochastic process and handle the temporal effect of random variables. DBN is also an extension of Markov Chains and Hidden Markov Model, which decreases the combinatorial explosion effect of complex systems by a more synthetic description. For those reasons, DBN can handle temporal dependencies between various time slices while preserving the internal dependencies at each time slice.

III. SDM MODEL CONSTRUCTION

Reliability of a system can be expressed as a joint probability function over some random variables and mapped onto a BN while preserving the original causal dependencies. We will introduce the method to model the casual dependencies and the higher order temporal dependencies of the embedded system with sequential feedback into an SDM model, and prove that the SDM is a DBN. Then, we will show how to construct the CPD tables for those random variables.

A. Qualitative part structure construction

While preserving all kinds of dependencies among the system signals, we build an SDM to represent a real system, proceed through the following steps:

1) Determine the boundary of the sequential embedded system, identify the main components in this boundary, and describe the data signal flow through those components. We can accomplish this by referring to the design documents, designers, implementers and deployers of the system. A simple example is shown in Fig. 1.

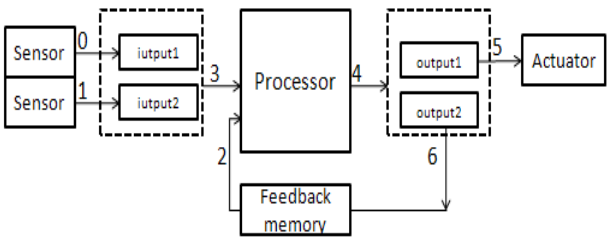

Fig. 1. Constructed meta-architecture of a given system

2) Construct a BN for the presented system, while ignoring the feedback. We map a node to each data signal. Edges presented in step 1 are regarded as causal dependencies. An arc is added between two nodes if the represented signals are connected by a component. Unroll the constructed BN for time slices to capture the temporal dependencies caused by signal feedback and correlation between time slices. This can be done by adding arcs between nodes from adjacent time slices and between

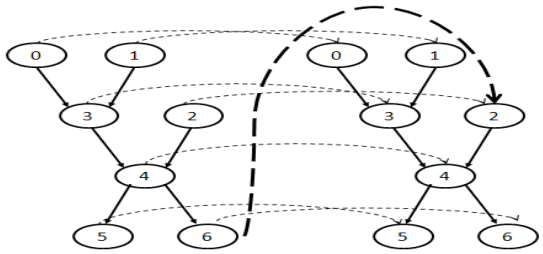

Fig. 2. Unrolled BN structure

nodes of the feedback signals. The unrolled BN structure is shown in Fig. 2.

3) Modify the unrolled BN structure to get the SDM model. This can be done by deleting some arcs to make the unrolled BN structure minimal. For example, given the node 2 and node 3 at time slice 2, the node 4 at time slice 2 is independent of the node 4 at time slice 1. We also assume that the root node signal is independent between time slices. Then, we can delete the arcs between the node x_i^t and x_i^{t+1} for any i while preserving the dependency correlations. The modified SDM for the unrolled BN structure is shown in Fig. 3. All the casual dependencies of the signals are maintained in each internal time slice and the feedback between signal x_6 and x_2 is captured by the thick full line.

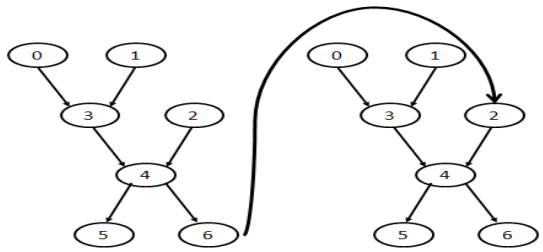

Fig. 3. Sequential dependence model(SDM)

With these three steps, we can build a SDM to capture all those correlations among the components, especially for the internal state of the feedback memory that makes the system not casual. Figure 3 shows the SDM model for the sequential embedded system presented in figure 1.

Then, we prove that the constructed SDM model, a directed acyclic graph, is a minimal representation of the underlying dependency model of the system components, and is a DBN. The proof process is based on the description of [13], [14]. We start by introducing some basic concepts. The following fundamental definitions have been defined in [13].

Definition 1 : Let M be a dependency model, which consists of a set of random variables $V = \{v_1, \cdots v_n\}$ and a probability function P over these variables. V_1, V_2 and V_3 are the subsets of V. Then, V_1 and V_2 are said to be conditional independent given V_3, if the joint probability function P over these three subsets satisfies :

$$P(V_1|V_2, V_3) = P(V_1|V_3) \quad \text{or} \quad P(V_2|V_1, V_3) = P(V_2|V_3)$$

The conditional independence between V_1 and V_2 given V_3 is denoted by the notation $I(V_1, V_3, V_2)$.

Definition 2 : A subset V_i of V is called a Markov blanket of element v_i if $I(v_i, V_i, V - V_i - v_i)$ holds. Further, the set V_i is called a Markov boundary of v_i if none of its proper subsets satisfy the triple independence relation.

Definition 3 : Let $d = \{v_{d1}, v_{d2} \cdots v_{dn}\}$ be an reordering of the random variables $\{v_1, v_2, \cdots v_n\}$ of the dependency model M. The boundary strata of M relative to d is an ordered set $\{V_{d1}, V_{d2} \cdots V_{dn}\}$, so that V_{di} is a Markov boundary of v_{di} with respect to the set $\{v_{d1}, v_{d2} \cdots v_{d(i-1)}\}$. The directed acyclic graph D, created by designating each V_{di} as the parents of the corresponding vertex v_{di}, is called a boundary directed acyclic graph of M relative to the ordering d.

Definition 4 : Let D be a directed acyclic graph, which consists of a set of nodes $U = \{x_1, x_2, \cdots x_n\}$ and arcs among them. X_1, X_2 and X_3 are the subsets of U. Then, X_1 and X_2 are said to be d-separated given X_3, if there is no path between any node in X_1 and X_2 satisfying the following two properties: every node on the path with converging arrows is in X_3 or has a descendent in X_3, and every node on the path without converging arrows is outside X_3. The d-separated relation between X_1 and X_2 given X_3 is denoted by the notation $\langle X_1|X_3|X_2\rangle$.

Definition 5 : A directed acyclic graph D is said to be an *I-map* of a dependency model M, if every d-separation condition displayed in D corresponds to a valid conditional independence relationship in M. Further, D is a minimal *I-map* of M if none of the arcs can be deleted without destroying its implied dependency model M.

Based on the dependency model M, the directed acyclic graph D, and those definitions of joint probability functions on the dependency model M and the directed acyclic graph D, the following definition proved in [13], shows the equivalence between directed acyclic graph and BN.

Definition 6 : If the directed acyclic graph D is a boundary directed acyclic graph of a dependency model M relative to an ordering d, then, D is a minimal I-map of the dependency model M. Further, if the directed acyclic graph D is a minimum *I-map* of the dependency model M, D is called a BN of M. Finally, we draw the conclusion of the SDM model with DBN as follows:

Theorem 1: ***The constructed sequential dependency model is a minimal I-map of the underlying reliability dependency model of the sequential embedded system, and is a dynamic bayesian network.***

Proof: Let us give an re-ordering d over the random signal variables $\{v_i^t\}$ of the embedded system($\{v_i^t\}$ denotes the signal i at time slice t and is a one to one mapping to $\{x_i^t\}$), and the ordering d satisfy the two properties: (1) for two random variables from the same time slice t, v_i^t must appear before v_j^t, if signal i is the input signal of component while signal j is the output of the same component; (2) for random variables from different time slices, the variables $\{v_i^t\}$ appear before the variables $\{v_i^{t+1}\}$. With this ordering, we derive the Markov boundary of a variable according to the dependency model as follow. If v_j^t represents the reliability of a output signal, then, its Markov boundary is the signal variables $\{v_i^t\}$ which is the input signals of the corresponding component. If v_j^t represents the reliability of a sequential input signal

variable, then its Markov boundary is the signal variables $\{v_i^{t-1}\}$ *which is the output signals of the corresponding components at previous time slice. Then we can see that the parents of each node variable are its Markov boundary in the sequential dependency model. By definition 6, the SDM is a boundary DAG, a minimal I-map of dependency model* M *and thus a Bayesian Network. Because the node variables and arcs also satisfy the definition presented in section II, the SDM is also a DBN.* ■

According to the theorem, the constructed SDM model in figure 3 is a minimal *I-map* of the underlying signal dependency model of the sequential embedded system presented in figure 1. The spatial-temporal dependencies as well as the higher order temporal dependencies of the sequential system components are mapped into the SDM nodes and arcs.

B. Quantitative part structure construction

As described in section II, the quantitative part of the DBN is a set of CPDs of the random variables given their parents. Each CPD is presented as a table driven by the core distribution function $P = \{f(x_i^t|parent(x_i^t))\}$. The joint probability distribution can be expressed by the product of each conditional probability as presented in the formula (2).

In the DBN, there are three kinds of nodes: the root node, feedback node and ordinary node. We incorporate the reliability of each component into these variables by well defined CPD tables. Those CPD tables for different kind of nodes and different processing structures are listed in the Table 1, 2, 3, 4.

TABLE I
CPD FOR THE ROOT NODE

$P(x_r^t\|E)$	$x_r^t = 1$	$x_r^t = 0$
$E = 0$	0	1
$E = 1$	$1-\varepsilon$	ε

Table I shows the conditional probability distribution of the root node, where E represents the environment of the system, x_r is the entry signal of the system and ε is the reliability of the system component that generate x_r. If the system is operated in a bad manner or the bad environment, x_r will take value 0 with probability 1. That means the signal will be in error state with probability 1. If the system is in a approximate environment, the entry signal x_r of the system will take value 1 with probability $1-\varepsilon$. For example, in the Figure 3, when the environment is approximate, the reliability of the signal x_0^1 is $1-\varepsilon$, where ε is the reliability of the system component sensor.

TABLE II
CPD FOR THE SERIAL-PROCESSING ORDINARY NODE

$f(x_i^t\|X_j^t)$	$x_i^t = 1$	$x_i^t = 0$
$X_j^t = \{1\cdots1, 1\cdots1\}$	$(1-\varepsilon_j)^{\|X_j^t\|}$	$1-(1-\varepsilon_j)^{\|X_j^t\|}$
$X_j^t = \{1\cdots1, 0\cdots0\}$	0	1

Table II shows the conditional probability distribution of the ordinary node, where X_j^t represents the parent set of x_i^t. These signals are combined in a serial manner to generate the x_i^t. If one of them fails, the x_i^t will take the value 0 with probability 1. When all the parent signals are correct, x_i^t will take the value 1 with probability $(1-\varepsilon_i)^{|X_j^t|}$, which means the reliability of the signal x_i^t under this condition. For example, in the Figure 3, when the signal x_0^1 and x_1^1 are correct, the reliability of the signal x_2^1 is $(1-\varepsilon_3)^2$, where ε_3 is the reliability of the system component I/O buffer.

TABLE III
CPD FOR THE PARALLEL-PROCESSING ORDINARY NODE

$f(x_i^t\|X_j^t)$	$x_i^t = 1$	$x_i^t = 0$
$X_j^t = \{1\cdots1, 1\cdots1\}$	$1-(\varepsilon_i)^{\|X_j^t\|}$	$(\varepsilon_i)^{\|X_j^t\|}$
$X_j^t = \{1\cdots1, 0\cdots0\}$	$1-\Pi_{x_j^t=1}(\varepsilon_i)$	$\Pi_{x_j^t=1}(\varepsilon_i)$

Table III shows the conditional probability distribution of the ordinary node, where those parent signals are combined in parallel redundancy to generate a signal x_i^t. The formula $1-\Pi_{x_j^t=1}(\varepsilon_i)$ is the probability that: when there is at least one parent signal taking value 1 and being processed by the system component correctly, x_i^t will take value 1. For example, in Fig. 1, if the sensor2 is used as a copy of sensor1, the I/O buffer receives a signal x_0 and a copy signal x_1 from the environment. The I/O buffer processes the two inputs independently and chooses a true-likely signal to initiate the signal x_3. When the value of (x_0, x_1) are $(1, 0)$, the reliability of the signal x_3 is $1-\Pi_{x_j^t=1}(\varepsilon_i) = 1-\varepsilon_3$.

TABLE IV
CPD FOR THE FEEDBACK NODE

$f(x_i^t\|X_j^{t-1})$	$x_i^t = 1$	$x_i^t = 0$
$X_j^{t-1} = \{1\cdots1, 1\cdots1\}$	$1-(\varepsilon_i)^{\|X_j^{t-1}\|}$	$(\varepsilon_i)^{\|X_j^{t-1}\|}$
$X_j^{t-1} = \{1\cdots1, 0\cdots0\}$	$1-\Pi_{x_j^t=1}(\varepsilon_i)$	$\Pi_{x_j^t=1}(\varepsilon_i)$

Table IV shows the conditional probability distribution of the feedback node, where x_i^t is the feedback variable, X_j^{t-1} represents parent set of x_i^t from the previous time slice. When all the parent signals are correct, x_i^t will take the value 1 with probability $(1-\varepsilon_i)^{|X_j^{t-1}|}$, which means the reliability of the signal x_i^t under this condition. For example, in the Fig. 3, when the signal x_6^1 is correct, the reliability of the feedback signal x_2^2 is $(1-\varepsilon_2)$, where ε_2 is the reliability of the component feedback memory.

With those CPD tables, it is easy to incorporate the individual component reliability into the SDM model of the sequential embedded systems. What we need is to change the value of ε according to the status of the system components and the stated operating environment. We can perform predictive and diagnostic inferences to study the behavior of the whole system or a single component.

IV. EXPERIMENT RESULT

In this section, we illustrate our method with a small example. Then, we valid our method through more complex industry applications. For those applications, the SDM model is constructed with the proposed steps and the conditional probabilities are assigned by the well defined CPD tables.

Many software tools such as BUGG, CODA and Nertica [15], [16], [17] have been implemented for the inferences of DBN, and we use the Nertica for the reliability analysis of the SDM model. We also calculate the reliability characterization of those systems with the original component-based BN method. Finally, we devise some random simulations to confirm the correctness of the reliability characterization. The simulations run on a computer with CPU 3.06 GHz and 2 GB of memory. The Monte Carlo framework for reliability analysis is based on fault injection. We embed the error probabilities of the system components into the Monte Carlo simulator.

A. Small Example

We will illustrate our method with an example, the reliability of a sequential embedded system: PLC for motor control. The system is used to control the motor to move forward and stop. It consists of two sensors to sample the move instruction and the stop instruction, an I/O buffer to store the two sampled inputs, a latch memory to store the feedback input variable, a processor to process those three inputs according to the embedded ladder program, and an I/O buffer to store the two outputs of the processer and a motor. The output signal will be stored and propagated to the next cycle as an input, and the order of the SDM model is two cycles. The meta structure of this embedded control system is shown in Fig. 1 and the SDM model of the system is the same as Fig. 3. We need to initiate the failure probability of each component to construct the quantitative part of the graph structure. A possible initiation of the ε for each component is shown in TABLE V.

TABLE V
STATIC FAILURE PROBABILITY OF EACH COMPONENT

component ‘	failure probability ε
Sensor1, Sensor2	0.05
Memory	0.01
Processor	0.02
Buffer1, Buffer2	0.01

TABLE VI
RELIABILITY FOR THE MOTOR CONTROL SYSTEM

processor ε	BN	SDM	simulation
0.01	0.42%	0.68%	0.61%
0.02	0.67%	0.99%	0.91%
0.03	0.73%	1.09%	1.02%
0.04	0.84%	1.34%	1.25%
0.05	1.57%	2.43%	2.30%
0.06	2.31%	3.91%	3.76%
0.07	3.92%	5.01%	4.83%
0.08	4.69%	5.76%	5.52%
0.09	5.21%	6.78%	6.59%
0.10	5.67%	7.69%	7.42%

Then, we can compute the reliability of the system with Nertica. The reliability for corresponding SDM in Figure 3 is 0.83. If we do not consider the higher order temporal correlations caused by the feedback memory, the reliability for corresponding BN in Fig. 2 is 0.67. While the failure probability of the system simulation is 0.91. We change the failure probability of the processor and the others are the same with the TABLE V. The results are presented in TABLE VI. Where the first column is the failure probability of the processor, the second column is the failure probability of the system based on BN, the third column is the failure probability of the system based on random the SDM method, and the fourth column is based on the system simulation. As can be observed in those results, the SDM model based reliability analysis is more closed to the simulation results, compared to the BN based method.

B. Complex Applications

The first complex application is an actual industrial PLC system, which is originally published in [18] and described in detail in [19]. It consists of four pistons (A, B, C, D) which are operated by four solenoid valves (V_1, V_2, V_3, V_4). Each piston has two corresponding normally open limit sensor contacts. Three push buttons are provided to start the system, to stop the system normally and to stop the system immediately in emergency. In a manufacturing facility, such piston system can be used to load/unload parts from a machine table, and extend/retract a cutting tool spindle. We set the failure probabilities of all the system components 0.05.

In the system, many output signals will be stored in the feedback memory, and propagated to the next cycle as an input. We construct the BN and SDM model based on the proposed steps and initiate the CPD tables. The order of the SDM model is decided by the cycles of the application. The failure probability of the system is presented in the Table VII.

TABLE VII
RELIABILITY FOR EMBEDDED SYSTEM

processor ε	BN	SDM	simulation
0.01	0.76%	1.28%	1.12%
0.02	1.32%	2.38%	2.27%
0.03	2.57%	4.10%	3.91%
0.04	4.52%	6.57%	6.41%
0.05	6.20%	8.55%	8.37%
0.06	7.91%	11.69%	11.46%
0.07	9.56%	12.98%	12.81%
0.08	10.43%	14.43%	15.21%
0.09	11.04%	16.56%	16.35%
0.10	11.92%	17.91%	17.72%

As can be observed from the simulation results, the values get by the BN based method are not closed to the simulation results. Because they mainly consider the casual dependencies of the system components and the signal among them, and the temporal dependencies caused by the feedback signals between time slices are ignored. The SDM model based method is more accurate. As shown in the third column of each table. The error between the SDM model results and the simulation results is less than that of BN. The results get by the SDM model is more closed to the run time station. Because the failure probability of the system components, the spatial dependencies between system components in a single time slice, the temporal dependencies between system components of different time slices, and the temporal dependencies due to the sequential feedback are captured by the SDM model and the CPD tables.

The other two more complex industry applications are introduced in detail in [20], [21], including the system com-

ponent distributions and the feedback signals. The first is a control system for a secondary clarifier scum removal that is installed in the Deer Island Water Pollution Treatment Facility near Boston, MA. The second is a double door control PLC system in the Lingshan stage control system in China. We set the failure probability of the system components similar to the piston system. All failure probabilities of the system components are set to be 0.05. Then, the failure probabilities of the three systems are presented in the Table VIII, IX, respectively. We can see from the two tables that, the SDM model is accurate even with these complex applications, the error between the simulation results and the SDM model based results is also less than that of BN.

TABLE VIII
RELIABILITY FOR THE DOUBLE DOOR CONTROL SYSTEM

processor ε	BN	SDM	simulation
0.01	1.53%	1.89%	1.83%
0.02	1.66%	2.17%	2.11%
0.03	2.13%	2.83%	2.76%
0.04	2.94%	3.59%	3.51%
0.05	3.87%	4.90%	4.83%
0.06	6.12%	8.72%	8.62%
0.07	7.67%	11.24%	11.02%
0.08	9.13%	13.72%	13.42%
0.09	11.32%	16.73%	16.21%
0.10	13.11%	19.74%	19.23%

TABLE IX
RELIABILITY FOR THE CLARIFIER SCUM REMOVAL SYSTEM

processor ε	BN	SDM	simulation
0.01	1.57%	1.90%	1.87%
0.02	1.69%	2.31%	2.25%
0.03	2.61%	3.49%	3.42%
0.04	0.94%	4.07%	3.98%
0.05	4.59%	6.58%	6.47%
0.06	5.41%	9.10%	8.96%
0.07	7.62%	12.41%	12.10%
0.08	9.89%	14.58%	14.25%
0.09	11.31%	17.87%	17.21%
0.10	14.19%	21.11%	20.58%

From those results, we can see that higher order correlations caused by the feedback has a high effect on the reliability of the system. We must take this kind of high order correlation into consideration when analyze systems. The proposed SDM is straightforward and closed to the simulation results than that of BN framework. The flexibility of the SDM is also useful, since for different samples of failures, only the failure probability of the component is changing. The constructed graph structure of sequential embedded system will not change, and the CPD table for each node needs to be adapted only once. Except for the reliability computation, we can also do some diagnosis and sensitivity analysis in the same manner.

V. CONCLUSION

In this paper, we have constructed an sequential dependency model to handle higher order spatial and temporal dependencies among the system components, especially the dependencies caused by the feedback of component signals. The constructed sequential dependency model is proved to be a Dynamic Bayesian Network formally. We define some conditional probability distribution tables for the nodes according to the processing mechanisms of the signal node, and initiate the tables with the failure probabilities of the corresponding system components. Then, the reliability of the sequential system is mapped onto a joint distribution function of the sequential dependency model over the signal nodes. The SDM model provides us a convenient way to incorporate the failure probabilities of each system component to carry on predictive inference and diagnose inference.

REFERENCES

[1] W. Lee, D. Grosh, and F. Tillman, "Fault tree analysis, methods, and applications- a review." *IEEE transactions on reliability*, 1985.

[2] M. Shooman, *Reliability of computer systems and networks*. Wiley Online Library, 2002.

[3] X. Zang, H. Sun, and K. Trivedi, "A BDD-based algorithm for reliability evaluation of phased mission systems," *IEEE Transactions on Reliability*, vol. 48, no. 1, pp. 50–60, 1999.

[4] M. Bouissou and J. Bon, "A new formalism that combines advantages of fault-trees and Markov models: Boolean logic Driven Markov Processes," *Reliability Engineering & System Safety*, vol. 82, no. 2, pp. 149–163, 2003.

[5] D. Wooff, M. Goldstein, and F. Coolen, "Bayesian graphical models for software testing," *IEEE Transactions on Software Engineering*, pp. 510–525, 2002.

[6] C. Bai, Q. Hu, M. Xie, and S. Ng, "Software failure prediction based on a Markov Bayesian network model," *Journal of Systems and Software*, vol. 74, no. 3, pp. 275–282, 2005.

[7] R. Donat, L. Bouillaut, A. Neji, and P. Aknin, "Comparison of two graphical models approaches for the modelling of multi-components system's reliability," in *Computers & Industrial Engineering, 2009. CIE 2009. International Conference on*. IEEE, 2009, pp. 1261–1266.

[8] H. Langseth and L. Portinale, "Bayesian networks in reliability," *Reliability Engineering & System Safety*, vol. 92, no. 1, pp. 92–108, 2007.

[9] P. Weber and L. Jouffe, "Reliability modelling with dynamic bayesian networks," 2003.

[10] J. Pearl, "Fusion, propagation, and structuring in belief networks* 1," *Artificial intelligence*, vol. 29, no. 3, pp. 241–288, 1986.

[11] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Computational intelligence*, vol. 5, no. 2, pp. 142–150, 1989.

[12] K. Murphy, "Dynamic bayesian networks: representation, inference and learning," Ph.D. dissertation, Citeseer, 2002.

[13] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[14] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter, "Probabilistic networks and expert systems. Statistics for Engineering and Information Science," *New York*, 1999.

[15] W. Gilks, A. Thomas, and D. Spiegelhalter, "A language and program for complex Bayesian modelling," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 43, no. 1, pp. 169–177, 1994.

[16] N. Best, M. Cowles, and S. Vines, "CODA Manual version 0.30," *MRC Biostatistics Unit, Cambridge, UK*, vol. 46, pp. 2020–2027, 1995.

[17] N. Manual, "Netica V1.05," *Norsys Software Corp*, 1997.

[18] K. Venkatesh, M. Zhou, and R. Caudill, "Comparing ladder logic diagrams and petri nets for sequence controller design through a discrete manufacturing system," *Industrial Electronics, IEEE Transactions on*, vol. 41, no. 6, pp. 611–619, 1994.

[19] H. Zhang, Y. Jiang, W. N. N. Hung, X. Song, and M. Gu, "Domain-driven probabilistic analysis of programmable logic controllers," in *Proceedings of the 13th international conference on Formal methods and software engineering*. Springer-Verlag, 2011, pp. 115–130.

[20] R. Wang, X. Song, J. Zhu, and M. Gu, "Formal modeling and synthesis of programmable logic controllers," *Computers in Industry*, vol. 62, no. 1, pp. 23–31, 2011.

[21] M. Zhou and E. Twiss, "Design of industrial automated systems via relay ladder logic programming and Petri nets," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 28, no. 1, pp. 137–150, 1998.