# Performance Evaluation of Smart Systems under Uncertainty

Saideep Nannapaneni, Abhishek Dubey, Sankaran Mahadevan
School of Engineering, Vanderbilt University, Nashville, TN 37235, USA

***Abstract*—This paper develops a model-based framework for the quantification and propagation of multiple uncertainty sources affecting the performance of a smart system. A smart system, in general, performs sensing, control and actuation for proper functioning of a physical subsystem (also referred to as a plant). With strong feedback coupling between several subsystems, the uncertainty in the quantities of interest (QoI) amplifies over time. The coupling in a generic smart system occurs at two levels: (1) coupling between individual subsystems (plant, cyber, actuation, sensors), and (2) coupling between nodes in a distributed computational subsystem. In this paper, a coupled smart system is decoupled and considered as a feed-forward system over time and modeled using a two-level Dynamic Bayesian Network (DBN), one at each level of coupling (between subsystems and between nodes). A DBN can aggregate uncertainty from multiple sources within a time step and across time steps. The DBN associated with a smart system can be learned using available system models, physics models and data. The proposed methodology is demonstrated for the design of a smart indoor heating system (identification of sensors and a wireless network) within cost constraints that enables room-by-room temperature control. We observe that sensor uncertainty has a higher impact on the performance of the heating system compared to the uncertainty in the wireless network.**

***Index Terms*—Smart System; Performance; Uncertainty; Bayesian; Dynamic;Reliability; Thermostat**

## I. Introduction

**Emerging Trends:** Recent developments in sensing, computational and communication technologies are enabling the development and use of smart systems to handle critical tasks in several domains of engineering, medicine, manufacturing and transportation. Dependability evaluation of smart system becomes essential as any failures in handling the critical tasks can result in serious financial and safety consequences. Dependability, which corresponds to the likelihood in performing the intended tasks is affected by different uncertainty sources arising from various aspects of a smart system. Thus, a thorough identification and quantification of uncertainty sources along with their impact on pre-defined performance measures becomes essential for the design of smart systems.

Model Integrated Computing (MIC) [1] presents a powerful framework for the design and analysis of complex systems that can be developed using the system architecture and component-behavior models. The first step before developing design options for a system is the creation of a modeling language (such as the Generic Modeling Environment (GME)[2]), which encodes all the semantic and syntactic information regarding several objects, their properties and relationships. Using the custom modeling language, several design alternatives can be developed as shown in [3]; these alternatives are evaluated against several design criteria such as cost and performance, to obtain the best design alternative. This works assumes the availability of such a modeling language along with several design alternatives. The primary objective of this paper is the development of a performance evaluation methodology under uncertainty of a smart system, which allows the system designers to explore several design options during design-time decision-making.

**Challenges:** The problem of performance evaluation under uncertainty can be considered as a generalization of reliability evaluation, which computes the probability that a pre-defined performance function crosses a design threshold. Smart systems can be viewed as Cyber-Physical Systems (CPS) as they comprise the actions of sensing, actuation and control. Current literature on CPS reliability have primarily considered data-driven approaches [4], [5], [6], [7] where failure rates for individual components are assumed to be available. Failure rates are generally obtained by repeated experimental or simulation-based evaluation of components until failure. As opposed to the existing approaches, this paper considers a probabilistic model-based approach, which is particularly useful when experimentation or simulation under uncertainty becomes prohibitively expensive or not feasible.

**Solution Approach:** This paper uses a partitioned approach [8] to analyze a coupled smart system, considering it as a feed-forward system over time. In addition to coupling between several subsystems (Fig. 1), there exists coupled interaction between several computational nodes when a distributed computational subsystem is used rather than a single node. We model the smart system using a two-level Dynamic Bayesian Network (DBN), a higher-level DBN to model the interactions between the subsystems and a lower-level DBN to model the interactions between the computational nodes.

A DBN model is used here for performance evaluation as opposed to performing Monte Carlo analysis on the system model (such as a Simulink model) for the following reasons: (1) In addition to prediction over time, a DBN naturally allows for performing Bayesian inference when new sensor data is available, and (2) Prediction from a smart system model (simulink) is generally deterministic for a given input whereas the prediction from a DBN is stochastic after aggregating several uncertainty sources. DBNs have been previously used for uncertainty modeling and performance evaluation of mechanical [9], industrial [10] and aerospace systems [11]. In this work, we use its capabilities to model a smart system.

978-1-5386-0435-9/17/$31.00 ©2017 IEEE

**Research Contributions:** The overall contributions made in this paper are: **(1)** Identification and quantification of multiple uncertainty sources through a probabilistic framework, **(2)** Construction of a two-level DBN to model a smart system, and **(3)** Performance analysis of a smart system using the DBN.

**Paper Organization:** Section II discusses the uncertainty sources in a conceptual smart system and illustrates them in the context of a smart indoor heating system. Section III provides a background to a DBN. Section IV presents the proposed methodology and Section V demonstrates the proposed methods using a smart indoor heating system, followed by a summary and future work in Section VI.

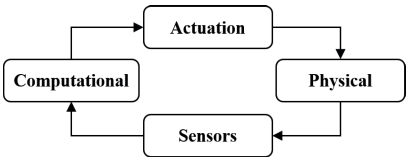


Fig. 1. Interactions between subsystems of a smart system

## II. Understanding the Problem

### A. Uncertainty sources in a smart system

*1) Computational subsystem:* We assume that a software application that runs on a computational subsystem is designed, tested and validated to perform within a set of design input ranges. We do not consider the random coding/latent errors, which are traditionally used to assess software performance [12]. Aside from software coding errors, we consider the following three uncertainty sources.

*Software inputs:* The inputs for the software application are obtained from the sensors collecting data about the physical subsystem and may be from the environment. When the sensor inputs are outside the design ranges (due to faulty sensors or large environmental variability), the software application may not provide correct outputs.

*Hardware resources:* A software application requires hardware resources (such as memory) to perform computation. In cases when hardware resources are unavailable, the computation cannot be completed resulting in faulty outputs.

*Communication uncertainty:* There are three types of communication in a smart system: (i) sensors to a computational subsystem, (ii) between computational nodes, and (iii) computational subsystem to an actuation subsystem. Faulty computational outputs and faulty control actions may arise due to unsuccessful communication. Robust communication protocols can be implemented to avoid any communication uncertainty but such robust protocols may not be always feasible due to higher design costs.

*2) Actuation and Physical subsystems:* Uncertainty sources during the modeling of these subsystems include the uncertainty in model parameters or in models, not being able to model the real-world phenomenon [13]. With time, an actuation or a physical subsystem may degrade; this degradation needs to be estimated and included in the models. If the degradation cannot be observed directly, it needs to be inferred through indirect methods, which may lead to uncertainty in its estimation. We explain below these uncertainty sources in context of a smart indoor heating system.

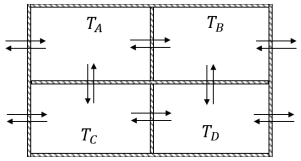


Fig. 2. Smart building with heat flow across rooms and outside environment

### B. Case Study: Smart Indoor Heating System

**Problem Description**: We consider here a smart indoor heating system in a commercial building that can control the heating vents in different rooms independently, enabling room-by-room temperature control similar to [14]. For illustration, we consider a building with four rooms as shown in Fig. 2.

Every occupant in a room is assumed to have a temperature comfort level, defined as the temperature range at which an occupant is comfortable in. These comfort levels are quantified using triangular distributions; their parameters are shown in Table I. In Table I, $C_A$, $C_B$, $C_C$ and $C_D$ represent the occupant comfort levels in each of the rooms. The table should be read as follows. The occupants in room $A$ are comfortable between temperatures 67 *F* and 73 *F* while their maximum comfort temperature is 70 *F*. Based on the current temperature, each occupant accumulates "comfort credits" defined in Eq. 1.

TABLE I
Comfort levels of occupants

| Parameter | Lower bound | Mode | Upper bound |
|---|---|---|---|
| $C_A$ | 67 | 70 | 73 |
| $C_B$ | 65 | 68 | 70 |
| $C_C$ | 68 | 72 | 74 |
| $C_D$ | 69 | 73 | 75 |

$$C_{cred} = \int_{t_1}^{t_2} \int_{T_1}^{T_2} \beta \; Pr(T,t) \; dTdt \tag{1}$$

In Eq. 1, $T_1$ and $T_2$ represent the room temperatures when time $t = t_1$ and $t = t_2$ respectively. $\beta$ is a comfort credit factor and $Pr(T,t)$ represents the probability density function of comfort level evaluated at the current room temperature at a given time $t$. A comfort credit can be regarded as a numerical measure that denotes the comfort level of an occupant.

Maintaining the rooms at the occupant comfort levels requires energy. To achieve energy conservation, a sustainability baseline temperature is set. If a room temperature is greater than a set baseline temperature, the occupant is penalized with negative "energy credits" and if the temperature is less than the baseline temperature, the occupant is awarded positive "energy credits". The energy credits are defined as

$$E_{cred} = \int_{t_1}^{t_2} \int_{0}^{T_B - T} \alpha \; dTdt \tag{2}$$

In Eq. 2, $T_B$, $T$ and $\alpha$ represent the baseline temperature, current room temperature and energy credit factor respectively.

The goal of this problem is to design a controller that maximizes the combination of comfort and energy credits such that a minimum comfort level is attained for all the occupants simultaneously. Due to the presence of uncertainty sources, the computed energy and comfort credits are not deterministic but stochastic. Therefore, we maximize the expected value of the sum of energy and comfort credits, formulated as

$$\underset{H_A,H_B,H_C,H_D}{\text{Maximize}} \sum_{i=A,B,C,D} E\Big[E^i_{cred} + C^i_{cred}\Big]$$

subject to

$$Pr(C^A_{cred} > 0 \cap C^B_{cred} > 0 \cap C^C_{cred} > 0 \cap C^D_{cred} > 0) > \gamma$$

In the above formulation, $H_A, H_B, H_C$ and $H_D$ are boolean variables representing if the heating system in a room is turned on or off. $\gamma$ represents a probability threshold that all room temperatures are in their occupant comfort ranges. $E[.]$ represents the expectation operator. $E^i_{cred}$ and $C^i_{cred}$, $i = A, B, C, D$ represent the energy and comfort credits accumulated by the occupants in each of the rooms.

TABLE II
COMPONENTS WITH THEIR COSTS AND UNCERTAINTIES

| Component | Performance | Cost |
|---|---|---|
| Type 1 sensor | 0.1 | 40 |
| Type 2 sensor | 0.15 | 25 |
| Type 3 sensor | 0.2 | 15 |
| Type 1 network | 0.95 | 200 |
| Type 2 network | 0.97 | 300 |
| Type 3 network | 0.99 | 400 |

Let three types of temperature sensors and three types of wireless network systems such as Bluetooth, 2.4 GHz Wi-Fi and 5 GHz Wi-Fi networks be available for design purposes. The uncertainty in a component performance and its costs are given in Table II. These values are chosen arbitrarily for illustration purposes and can be replaced with actual values from a manufacturer. The affordable budget to this design is assumed to be 450 units. In Table II, performance for a sensor refers to the standard deviation of the measurement error in *Fahrenheit* while that for a network refers to its reliability, i.e., probability that a data packet is successfully transmitted.

**Uncertainty sources in the case study:** The uncertainty sources with respect to the computational subsystem include: (1) Sensor uncertainty in the room temperature measurements, (2) Environmental variability in outside temperature, (3) Communication uncertainty between the temperature sensors, computational subsystem and the heating system, and (4) Resource availability as the computational subsystem is assumed to perform other operations such as lightening control and security systems. Uncertainty in the physical subsystem include the uncertainty in the estimation of thermal conductivity, which is used for calculating a suitable control action.

**Problem Statement:** The goal is to find the best design combination of sensors and a network that satisfies the budget constraints and maximizes a performance evaluation metric, defined as the probability that the controller cannot find a control action to maintain the occupant comfort requirements.

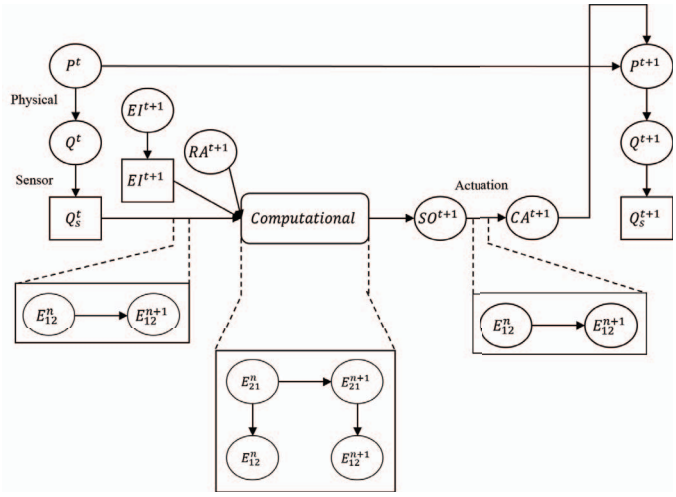


Fig. 3. A two-level DBN model of a conceptual smart system showing the dependence between several systems

## III. BACKGROUND: DYNAMIC BAYESIAN NETWORKS

Dynamic Bayesian Networks belong to a class of state-space models used to model the time-dependent behavior of dynamic systems. A DBN can be regarded as a composition of two BNs. (1) a BN connecting the variables in a single time instant, and (2) a BN connecting variables across time instants.We consider DBNs with the Markov assumption,i.e., the state variables in the current time step are only dependent on the state variables in the previous time step, resulting in a 2-slice DBN [15].

$$P^{t+1} = G(P^t, v^{t+1}) \tag{3}$$

$$Q^t = H(P^t) \tag{4}$$

In the above equations, $P^t$ and $P^{t+1}$ represent the state variables in two time steps. Similarly, $Q^t$ and $Q^{t+1}$ represent observation variables at two time steps. The evolution to $P^{t+1}$ from $P^t$ can be represented in Eq. 3. $v^t$ and $v^{t+1}$ refer to system inputs at time $t$ and $t+1$ respectively. Eq. 4 represents the relationship connecting observation variables $Q^t$ and state variables $P^t$. Probabilistic modeling of systems where Markov assumption does not hold good is not considered in this paper.

## IV. PROPOSED METHODOLOGY

### A. Constructing the DBN model for a Smart System

Fig. 3 shows a representative DBN model; the description of variables is presented in Table III. In Fig. 3, the rectangle with rounded corners named 'Computational' does not represent a DBN node, but represents a lower-level DBN depending on the interaction pattern as detailed below.

### B. Modeling the computation system

Distributed computing systems are often arranged into a sequence of components communicating over well-defined interaction protocols [16]. Such rigid interaction semantics help analyze the behavior of distributed system [13] and also enables developing online fault detection and diagnosis mechanisms [17]. In this paper, we assume that the smart

TABLE III
PARAMETERS IN THE DBN MODEL (FIG. 3)

| Parameter | Description |
|---|---|
| $P^t, P^{t+1}$ | State variables at time $t = t, t+1$ |
| $Q^t, Q^{t+1}$ | Observation variables at time $t = t, t+1$ |
| $Q_s^t, Q_s^{t+1}$ | Sensor measurements of observation variables at time $t = t, t+1$ |
| $EI_s^{t+1}$ | Sensor measurements of environmental inputs at time $t = t+1$ |
| $RA^{t+1}$ | Resource availability |
| $SO^{t+1}$ | computational output at time $t = t+1$ |
| $CA^{t+1}$ | Control action at time $t = t+1$ |

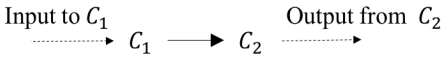


Fig. 4. 2-node asynchronous interaction pattern

system is using this distributed computing pattern. We focus on developing conditional relationships for a distributed computing subsystem as the conditional relationships for physical and actuation subsystems can be derived from physics models or data [18], [19]. The conditional relationships for complex computational subsystems are studied by breaking them down into two basic interaction patterns: (1) 2-node asynchronous, and (2) 2-node synchronous [16], discussed below.

**2-node asynchronous interaction pattern:** We first consider a 2-node asynchronous interaction pattern with two computational nodes $C_1$ and $C_2$ as shown in Fig. 4, where inputs to the computational system are input to $C_1$. $C_1$ performs some computation, communicates the results to $C_2$, which performs further computation and outputs the results. Let $E_1$ and $E_2$ denote the events that the $C_1$ and $C_2$ successfully perform their analysis. And let $E_{12}$ correspond to the event that the data from $C_1$ is successfully transmitted to $C_2$. In general, data is transmitted across a network in packets. Let $E_{12} = 0, 1$ correspond to the two states that a given packet reaches and does not reach the destination (here $C_2$). We define a smaller time-scale where a time step ($n$) corresponds to the time it takes to send a packet of data. Let $E_{12}^n$ corresponds to the event that a packet reaches the destination at time step $n$. We make a Markov assumption that the state of the event $E_{12}^n$ depends on the state of the event at the previous time step, $E_{12}^{n-1}$. In other words, if a packet cannot be transmitted at time $n-1$ due to network interruptions, then it is assumed to affect the probability packet transmission at time $n$. Therefore, the DBN corresponding to a 2-node asynchronous interaction pattern can be represented as shown in Fig. 5.

**2-node synchronous interaction pattern:** A synchronous interaction pattern (Fig. 6) is characterized by a sequence of *request* and *reply* messages as shown in Fig. 7, where $C_2$ requests for data and $C_1$ replies accordingly. Similar to the previous asynchronous case, we define a smaller time-scale


Fig. 5. DBN for 2-node asynchronous interaction pattern

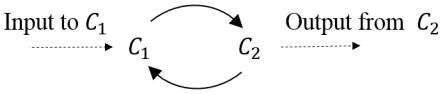


Fig. 6. 2-node synchronous (request-reply) interaction pattern

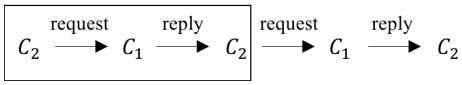


Fig. 7. Decoupling request and reply messaging

where a time step contains a set of request and reply messages (represented using a rectangle in Fig. 7). We decompose the coupled request-reply communication into a series of request and reply messages similar to the partitioned approach used for decoupling the interactions between the computational and physical subsystems. In addition to $E_1$, $E_2$ and $E_{12}$ defined above, we further define $E_{21}$ as the event representing the successful transmission of data packet from $C_2$ to $C_1$. Thus, at every time step $n$ one event each of $E_{21}$ and $E_{12}$ occur. Since both these events happen at the same time step, their states are assumed to be dependent one each other. If $E_{21} = 1$ (failure), then the probability of $E_{12} = 1$ will be higher when compared to when $E_{21} = 0$. In addition to the dependence at the same time step, there exists a dependence between the states of $E_{21}$ events at successive time steps, as mentioned in the case of asynchronous case. The DBN for this 2-node synchronous case is provided in Fig. 8. When $E_{21}^n$ is successful, then $E_{21}^{n+1}$ is dependent on $E_{12}^n$ and if $E_{21}^n$ is not successful (failed request message and this implies no reply message), then $E_{21}^{n+1}$ is assumed to be dependent on $E_{21}^n$.

**Extending to complex interaction patterns:** Consider a 4-node interaction pattern shown in Fig 9. The interactions between $C_1$ and $C_2$, and $C_2$ and $C_3$ are synchronous whereas interaction between $C_3$ and $C_4$ is asynchronous in nature. Therefore, the DBN in Fig. 8 can be used to model the synchronous interactions between $C_1$ and $C_2$, and $C_2$ and $C_3$, and the DBN in Fig. 5 to model the interaction between $C_3$ and $C_4$. It should be noted that the communication between the nodes are sequential in nature, i.e., communication between $C_2$ and $C_3$ occurs after the communication between $C_1$ and $C_2$. Therefore, their DBNs can also be represented sequentially. In

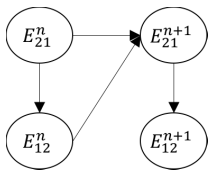


Fig. 8. DBN for a 2-node synchronous interaction pattern

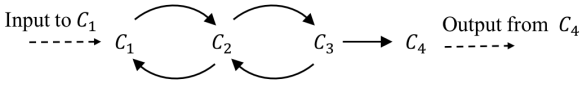


Fig. 9. 4-node complex interaction pattern

some cases, it might be possible that the outcome of one interaction (if data is transmitted) may influence the outcome of the following interaction; this influence may be quantified through a conditional dependence relationship across the DBNs.

### C. Performance evaluation

For design-time decision-making, we simulate the possible design options using the DBN models constructed in Section IV-A to evaluate their performance under uncertainty.

**Modeling sensor uncertainty:** The sensor uncertainty $\epsilon_s$ is typically modeled using a Gaussian distribution with zero mean; since positive and negative errors occur with equal probability [20]. The relationship of the measurement variable ($Q_s^t$) conditioned on the unknown value of the observation variable($Q^t$) can be represented as

$$Q_s^t|Q^t = Q^t + \epsilon_s \tag{5}$$

The sensor then reads a value from the probability distribution of $Q_S^t$ and sends the data to the computational subsystem.

**Simulating asynchronous interaction pattern:** Consider the 2-node interaction pattern as shown in Fig. 4, where $C_1$ receives the input, processes the information and outputs to $C_2$. Let $p$ data packets be transferred from $C_1$ to $C_2$ and if $r$ packets get successfully transmitted, then it is assumed that all the information can be reconstructed at $C_2$. In the lower-level DBN, one data packet is assumed to be transmitted in each time step. Since there are $p$ packets, $n$, which represents time in the lower-level DBN goes from $n = 1$ to $n = p$. The events corresponding to each data packet transmission are represented as $E_{12}^n$. The joint probability of all the events corresponding to transmission of $p$ data packets is equal to $P(E_{12}^1, E_{12}^2 ... E_{12}^p)$. This joint probability can then be decomposed into a product of marginal and conditional probabilities defined as

$$\begin{aligned} P(E_{12}^1, E_{12}^2 ... E_{12}^p) &= P(E_{12}^1) \times P(E_{12}^2|E_{12}^1) \\ &\times P(E_{12}^3|E_{12}^2, E_{12}^2) \times ... P(E_{12}^p|E_{12}^1, ... E_{12}^{p-1}) \end{aligned} \tag{6}$$

Using the Markov assumption as mentioned in Section IV-A, Eq. 6 can be simplified as

$$\begin{aligned} P(E_{12}^1, E_{12}^2 ... E_{12}^p) &= P(E_{12}^1) \times P(E_{12}^2|E_{12}^1) \\ &\times P(E_{12}^3|E_{12}^2) \times ... P(E_{12}^p|E_{12}^{p-1}) \end{aligned} \tag{7}$$

Let $R^1$ represent the probability of a successful data packet transfer at a lower time-scale time step $n = 1$. The conditional dependence relationship for successful data transmission are represented as shown in Table IV. In Table IV, $R_{ij}$ represents the probability of data transmission event in the current time step $j$ conditioned on the data transmission event in the previous time step $i$ ($i, j = \{0, 1\}$). 0 and 1 represent success and failure of an event respectively. The parameters in Table IV can be estimated through an aggregation of historical data, simulations and expert knowledge regarding the system.

For illustration, consider a case when $r = 2$ and $p = 3$. Across three time steps, there exist 8 combinations with two outcomes in each time step (data packet delivered or not delivered). If two successful transmissions are required, then there exist four combinations that result in successful data transmission. The set of successful combinations and their probabilities are given in Table V. Since $p = 3$, there are three elements in each combination. The overall success probability can be calculated as the sum of all the individual probabilities. Using the overall probability, a binary random sample can be drawn to simulate the data transmission in a 2-node asynchronous interaction pattern.

TABLE IV
CONDITIONAL PROBABILITIES OF SUCCESSFUL DATA TRANSMISSION BETWEEN TWO CONSECUTIVE TIME STEPS

| | Successful at time step $n+1$ | Unsuccessful at time step $n+1$ |
|---|---|---|
| Successful at time step $n$ | $R_{00}$ | $R_{01}$ |
| Unsuccessful at time step $n$ | $R_{10}$ | $R_{11}$ |

TABLE V
2-NODE ASYNCHRONOUS INTERACTION PATTERN: SUCCESSFUL COMBINATIONS AND THEIR PROBABILITIES

| **Combination** | **Probability** |
|---|---|
| [0,0,0] | $R^1 \times R_{00} \times R_{00}$ |
| [0,0,1] | $R^1 \times R_{00} \times R_{01}$ |
| [0,1,0] | $R^1 \times R_{01} \times R_{10}$ |
| [1,0,0] | $(1 - R^1) \times R_{10} \times R_{00}$ |

**Simulating synchronous interaction pattern:** As opposed to the asynchronous system, we assume we require $r$ successful request-reply pairs in a synchronous system since a *reply* does not occur unless there is a *request* and *reply* does not always occur for every *request*. The joint probability of $p$ $request - reply$ pairs, assuming one occurs at each lower-scale time step can be computed using Eqs. 6 and 7.

Let $R_2$ represent the probability of successful *request* message at lower time-scale time step $n = 1$. Let $R_{12}$ represent the probability of successful *reply* message when the request message is successful. Therefore, $R_2 \times R_{12}$ refers to the reliability of a request-reply pair at any time step $n = 1$. For illustration, assume the same conditional relationships between two requests across two successive time steps as provided in Table IV. Given the dependence relationships across time steps, the probability of $r$ successful pairs out of $p$ can be computed using Eq. 7. For $p = 3$ and $r = 2$, the set of successful combinations and their probabilities are provided in Table VI. Therefore, the overall success probability can be calculated as the the sum of all the individual probabilities.

TABLE VI
2-NODE SYNCHRONOUS INTERACTION PATTERN: SUCCESSFUL COMBINATIONS AND THEIR PROBABILITIES

| **Combination** | **Probability** |
|---|---|
| [0,0,0] | $(R_2 \times R_{12}) \times (R_{00} \times R_{12}) \times (R_{00} \times R_{12})$ |
| [0,0,1] | $(R_2 \times R_{12}) \times (R_{00} \times R_{12}) \times (1 - R_{00} \times R_{12})$ |
| [0,1,0] | $(R_2 \times R_{12}) \times (R_{01} \times (R_{10} \times R_{12}) + (R_{00} \times (1 - R_{12})) \times (R_{10} \times R_{12})$ |
| [1,0,0] | $(R_2 \times (1 - R_{12}) \times (R_{10} \times R_{12}) + (1 - R_2) \times (R_{10} \times R_{12})) \times (R_{00} \times R_{12})$ |

As mentioned in Section IV-A, a failure in a message transmission can be due to a failure in either *request* or *reply* message transmission. Therefore, the last two combinations in Table VI, have two terms representing the cases of failures in request and reply message transmissions. After the completion of computational analysis, the computational output is communicated to the actuation system; this communication can be simulated using an asynchronous or a synchronous system as described above. If the actuation system cannot receive the data, an assumption that the control action in the previous time step is continued in the current time step is made.

**Simulating resource availability:** In the DBN model for a generic smart system shown in Fig. 3, if the data is successfully transmitted from sensors, the computational subsystem estimates the state variables $P^t$ through Bayesian Inference (typically using particle filtering [21]) and calculates the necessary control action for the next time step. To perform the analyses, the computational nodes should have the necessary resources. Let there be $N$ computational nodes and $E_{i,k}$, $i = 1, 2, 3...N$ represent the events corresponding to their resource availability. The joint probability can be defined as

$$P(E_{1,k}, E_{2,k}..E_{N,k}) = P(E_{1,k}) \times P(E_{2,k}|E_{1,k}) \times ..P(E_{N,k}|E_{1,k}, E_{2,k}...E_{N-1,k}) \quad (8)$$

In this discussion, we consider two hardware resources: power and memory. We assume that the power is supplied through a battery and each node is assumed to have an associated battery. Under this assumption, the resource availability of one node is independent of the resource availability of another node. Thus, Eq. 8 can be simplified as

$$P(E_{1,k}, E_{2,k}..E_{N,k}) = P(E_{1,k}) \times P(E_{2,k}) \times ..P(E_{N,k}) \quad (9)$$

Let $S_r$ refers to the probability of having necessary resources at each computational node. Therefore, the probability that all the events , $E_{i,k}$, $i = 1, 2...N$, are successful (assuming the same resource probability) is equal to $S_r^N$. A binomial random sample with a success probability of $S_r^N$ is drawn to simulate the resource availability. It may be possible that different nodes might have different resource probabilities and dependent on each other. An example of such a case is when a group of nodes have a common power supply unit. In cases of dependent resource availability, Eq. 8 cannot be simplified to obtain Eq. 9, and the dependence needs to be modeled using either expert knowledge or simulations. Simulation of resource availability and data transmission in the computational subsystem provides the posterior distributions of $P^t$ and the control action at time $t+1$, which are then used to estimate the prior distributions of $P^{t+1}$. Sensor measurements at time $t+1$ are used to estimate the posterior distributions of $P^{t+1}$ and control action at time $t+2$. This process is repeated until a pre-defined analysis time for performance evaluation.

## V. Case Study: Implementation and Results

In this section, we demonstrate the proposed methodologies for the case study in Section II-B. For each design, five temperature sensors are used (one for each room and one for outside temperature) and the overall cost for each design is given in Table VII. Due to budget constraints, only 5 out of 9 configurations from Table VII are feasible; these configurations are indicated in bold. For each sensor type, the most reliable of all possible network types is identified and considered for analysis. This corresponds to the first two design options along the diagonal in Table VII.

TABLE VII
Design configurations and their costs

| | Type 1 network | Type 2 network | Type 3 network |
|---|---|---|---|
| Type 1 sensor | **400** | 500 | 600 |
| Type 2 sensor | **325** | **425** | 525 |
| Type 3 sensor | **275** | **375** | 475 |

The conditional dependence relationships for communications across two lower time-scale time steps (as discussed in Section IV-A) are provided in Table VIII.In Table VIII, $X$ equals 0.95, 0.97 and 0.99 when networks of Types 1,2 and 3 are used respectively (Table II).

TABLE VIII
Communication probability

| | Successful at time step $n+1$ | Unsuccessful at time step $n+1$ |
|---|---|---|
| Successful at time step $n$ | $X$ | 1-$X$ |
| Unsuccessful at time step $n$ | 0.9 | 0.1 |

**Problem Parameters**: Each room has dimensions $9m \times 9m \times 5m$ and one window with dimensions $1m \times 1m$. The thermal conductivity of the window is assumed to be 0.3 $W/F.m$ while that of the wall is not known precisely and need to be calibrated from data. The thickness of wall and window insulations are assumed to be 0.8 $m$ and 0.04 $m$ respectively. The density and heat capacity of air are assumed to be equal to 1.225 $kg/m^3$ and 558.55 $J/kg.F$. The temperature and the amount of hot air blown from the heater are 110 $F$ and 106 $kg/hr$ respectively. The values of $T_B$, $\alpha$ and $\beta$ are assumed as 70 $F$, 50 and 100 respectively. The outside temperature data are obtained from SML2010 Data Set [22], which can be downloaded from the UCI Machine Learning Repository [23]. We used the weather temperature data from this dataset as the outside temperature data for our analysis.

**Computational system**: The computational subsystem is assumed to consist of two nodes, $C_1$ and $C_2$. $C_1$ receives the sensor data and estimates the indoor temperatures through Bayesian Inference; these estimates are then transmitted to $C_2$ which calculates the control action for the next time step. The probability of resource availability for both nodes is assumed as 0.9. Two-node asynchronous interactions are assumed between the temperature sensors and computational subsystem, and between computational subsystem and actuation subsystem, and a two-node synchronous interaction between the computational nodes.

**DBN model**: Fig. 10 presents the DBN for the smart heating system. It should be noted that there exists no state variable $P_t$ but only an observation variable $Q_t$, which are the actual temperatures in each room and the outside temperature

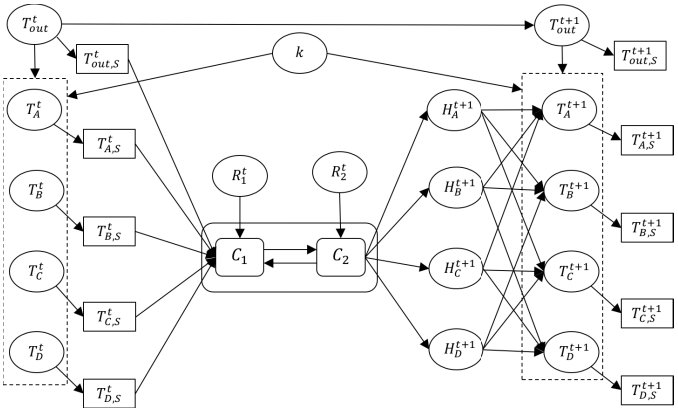


Fig. 10. DBN model for the smart indoor heating systems

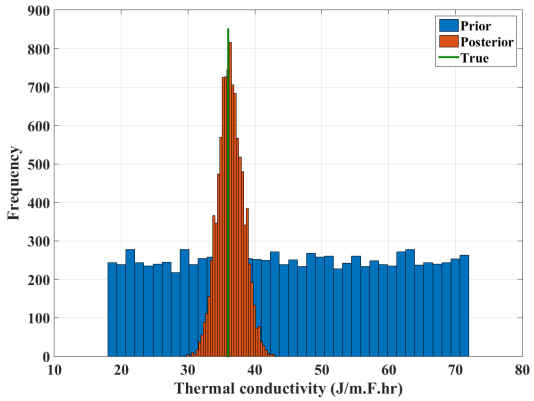


Fig. 12. Calibration results with sensor uncertainty = 0.15$F$

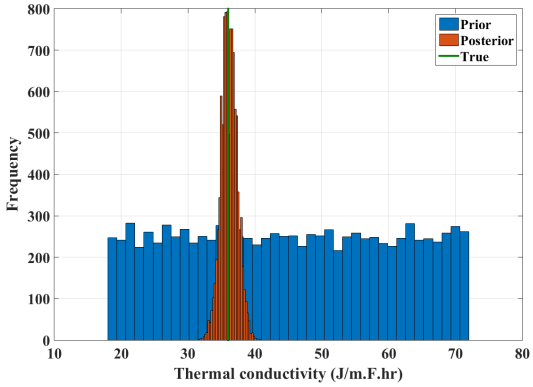


Fig. 11. Calibration results with sensor uncertainty = 0.1$F$

($T_A, T_B, T_C, T_D$ and $T_{out}$). The corresponding measurements of these variables are represented as $T_{A,s}, T_{B,s}, T_{C,s}, T_{D,s}$ and $T_{out,s}$ respectively. $k$ refers to the thermal conductivity coefficient of the wall.$R_1$ and $R_2$ refer to the resource availability variables corresponding to computational nodes $C_1$ and $C_2$ respectively. $H_A, H_B, H_C$ and $H_D$ refer to the control actions in each of the rooms. The thermal conductivity and outside temperature affect the temperatures in all rooms, which needs to be represented by arrows from $k$ and $T_{out}$ to every room temperature. For better visualization, all room temperatures are grouped in a 'dotted' rectangle. An arrow to the 'dotted' rectangle should be interpreted as a arrow going to every room temperature. Fig. 10 shows a higher-level DBN; the lower-level DBNs can be constructed as detailed in Section IV-A.

**Model Calibration:** Each design results in a different calibration of thermal conductivity due to the difference in sensor uncertainty. The prior and posterior distributions for both the design options are presented in Figs. 11 and 12 respectively. From the plots, it can be seen that the posterior variance in Fig. 12 is higher due to higher sensor uncertainty.

**Performance analysis:** We ran the smart indoor heating system for one day (with time steps of 15 min). We used the outside temperature corresponding to March 25 in the SML2010 Dataset. The reason for choosing this date is explained later in this section. We start with available sensor measurements at 12 am and the analysis continues until 12 am to next day. For each type of communication, two messages are transmitted and even if one reaches successfully, the data is assumed to be successfully transmitted. For simplicity, the conditional relationships for communication uncertainty within and across two lower time-scale time steps in the synchronous interaction pattern between the computational nodes are assumed to be the same as given in Table VIII. Eqs. 1 and 2 are evaluated between every two successive sensor measurements (15 min); therefore, $t_2 = t_1 + 15$ and $t_1 = 0$.

Using the sensor measurements, Bayesian inference is performed using particle filtering to estimate the outside temperature and room temperatures. For predicting the outside temperature in the next time step, we used the weather data from the previous 10 days to construct a Gaussian distribution to represent the change of temperature between the current time step and next time step; this Gaussian distribution is used for temperature forecasting in the next time step. We used data from March 15 to March 24 for temperature forecasting on March 25, 2012. Using the forecast outside temperature, the control action that optimizes the sum of energy and comfort credits is identified, and using this, the prior distributions of the room temperatures in the next time step are obtained.

The same process is repeated 250 times to obtain the failure probability. The practical reasoning for this is as follows. If a particular design alternative is installed at 250 homes, what is the probability that the occupants be uncomfortable at any time during the analysis? The analysis is carried out using both the design alternatives and the success probabilities are 0.99 and 0.93 respectively. Therefore, the better design option is the one with the sensor uncertainty of 0.1 *Fahrenheit* (Type 1 sensor) and 0.95 network reliability (Type 1 network).

This preliminary case study is used to discuss the model-based performance evaluation methodology for smart systems. Comparison with other solutions will be considered in the near future work. The case study with the current problem formulation does not adapt over time to new conditions. Thermal conductivity is a model parameter that varies with time due to the wear of insulation. Therefore, the system needs to re-calibrate the thermal conductivity on the go with the temperature sensor data for further decision-making. This learning process is not considered in this paper as the time of

analysis (1 day) is small compared to the time scale of the wear of insulation (typically in the order of several months).

**Application to complex smart systems:** The proposed methods are generic and can be applied to more complex interaction patterns between several nodes. An example with such properties of complex computational interaction pattern is a smart grid, which has a collection of sensors (smart meters), computational, actuation (electricity transfer mechanism such as lines) and physical subsystems (houses). The inputs (power generation sources such as fossil fuels, wind and solar) are associated with variability due to environmental conditions. In the case of a smart grid, the user requirements change with time everyday; these user requirements can be considered similar to occupant comfort levels in the smart indoor heating system. A smart grid consists of several utility units or substations, and each utility unit can be modeled as a computational node. These units collects sensor data from a region and communicate with each other to find an optimum way to meet the electricity demands.

## VI. Summary and Future Work

This paper developed a model-based framework for uncertainty quantification and aggregation in a smart system for its performance assessment to enable design-time decision-making. Uncertainty in the physical system may be due to uncertain model parameters and model inadequacy. Uncertainty in the computational subsystem is due to the uncertainty in the availability of hardware resources and in the network communication. A coupled smart system containing physical, sensing, computational and actuation subsystems is analyzed as a feed-forward system in time and modeled using a Dynamic Bayesian network. The computational subsystem has communication between several nodes; this interaction is also modeled using a DBN, resulting in a two-level DBN for modeling a smart system. Physics models and/or data are used to establish the conditional dependence relationships for the physical and actuation subsystems. DBNs corresponding to basic interaction patterns such as 2-node asynchronous and synchronous are detailed. Complex interaction patterns can be broken down into these basic patterns for performance evaluation. The proposed methods for performance evaluation are demonstrated for the design of a smart indoor heating system that enables room-by-room temperature control.

Future work should investigate the proposed methodology for large-scale systems such as smart grids. In addition, methods for performance evaluation of Human-in-the-loop CPS (H-CPS) need to be investigated. Also, procedures for automated extraction of a DBN from system models, physics models and available data regarding a smart system for its performance assessment need to be investigated.

## VII. Acknowledgments

The research reported in this paper was partly funded by Siemens Corporate Technology and National Institute of Standards and Technology (NIST) (Grant No. 70NANB16H297). The support is gratefully acknowledged. Any opinions, findings, and conclusions expressed here are those of the authors and do not reflect the views of the funding sources.

## References

[1] J. Sztipanovits and G. Karsai, "Model-integrated computing," Computer, vol. 30, no. 4, pp. 110–111, 1997.

[2] A. Ledeczi, M. Maroti, A. Bakay, G. Karsai, J. Garrett, C. Thomason, G. Nordstrom, J. Sprinkle, and P. Volgyesi, "The generic modeling environment," in Workshop on Intelligent Signal Processing, Budapest, Hungary, vol. 17, 2001, p. 1.

[3] Z. Lattmann, "An analysis-driven rapid design process for cyber-physical systems," PhD, Vanderbilt University, August 2016.

[4] X. Sun, N. Huang, B. Wang, and J. Zhou, "Reliability of cyber physical systems assessment of the aircraft fuel management system," in Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on. IEEE, 2014, pp. 424–428.

[5] L. Wu and G. Kaiser, "Fare: A framework for benchmarking reliability of cyber-physical systems," in Systems, Applications and Technology Conference (LISAT), IEEE Long Island. IEEE, 2013, pp. 1–6.

[6] Z. Li and R. Kang, "Strategy for reliability testing and evaluation of cyber physical systems," in Industrial Engineering and Engineering Management (IEEM), 2015 IEEE International Conference on. IEEE, 2015, pp. 1001–1006.

[7] S. Nannapaneni, S. Mahadevan, S. Pradhan, and A. Dubey, "Towards reliability-based decision making in cyber-physical systems," in 2016 IEEE International Conference on Smart Computing (SMARTCOMP), May 2016, pp. 1–6.

[8] C. A. Felippa, K. Park, and C. Farhat, "Partitioned analysis of coupled mechanical systems," Computer methods in applied mechanics and engineering, vol. 190, no. 24, pp. 3247–3270, 2001.

[9] D. Straub, "Stochastic modeling of deterioration processes through dynamic bayesian networks," Journal of Engineering Mechanics, vol. 135, no. 10, pp. 1089–1099, 2009.

[10] K. Medjaher, J.-Y. Moya, and N. Zerhouni, "Failure prognostic by using dynamic bayesian networks," IFAC Proceedings Volumes, vol. 42, no. 5, pp. 257–262, 2009.

[11] C. Li, S. Mahadevan, Y. Ling, L. Wang, and S. Choze, "A dynamic bayesian network approach for digital twin," in 19th AIAA Non-Deterministic Approaches Conference, 2017, p. 1566.

[12] A. K. Verma, S. Ajit, and D. R. Karanki, "Software reliability," in Reliability and Safety Engineering. Springer London, 2016, pp. 183–217.

[13] S. Nannapaneni and S. Mahadevan, "Reliability analysis under epistemic uncertainty," Reliability Engineering & System Safety, vol. 155, pp. 9–20, 2016.

[14] "Keen home." [Online]. Available: https://keenhome.io

[15] K. P. Murphy, "Dynamic bayesian networks: representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[16] A. Dubey, G. Karsai, and N. Mahadevan, "A component model for hard real-time systems: Ccm with arinc-653," Software: Practice and Experience, vol. 41, no. 12, pp. 1517–1550, 2011.

[17] N. Mahadevan, A. Dubey, and G. Karsai, "Application of software health management techniques," in Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. ACM, 2011, pp. 1–10.

[18] D. Koller and N. Friedman, Probabilistic graphical models: principles and techniques. MIT press, 2009.

[19] S. Nannapaneni and S. Mahadevan, "Manufacturing process evaluation under uncertainty: A hierarchical bayesian network approach," in ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2016, pp. V01BT02A026–V01BT02A026.

[20] N. C. Barford, "Experimental measurements: precision, error and truth," Chichester: Wiley, 1985, 2nd ed., 1985.

[21] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," IEEE Transactions on signal processing, vol. 50, no. 2, pp. 174–188, 2002.

[22] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," Energy and Buildings, vol. 83, pp. 162–172, 2014.

[23] "Sml2010 data set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SML2010