Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 8325054, 8 pages
http://dx.doi.org/10.1155/2016/8325054

Research Article

Mathematical Modeling of Smart Space for Context-Aware System: Linear Algebraic Representation of State-Space Method Based Approach

Sung-Hyun Yang, M. Humayun Kabir, and M. Robiul Hoque

Department of Electronic Engineering, Kwangwoon University, Seoul 139-701, Republic of Korea

Correspondence should be addressed to M. Humayun Kabir; humayun@kw.ac.kr

Received 28 August 2015; Revised 16 February 2016; Accepted 14 March 2016

Academic Editor: Mustafa Tutar

A smart space is embedded with several components such as sensors, actuators, and computing devices that enable the sensing and control of the environment, and the inhabitants interact with the devices in the smart space whenever they need to. To model a smart space, a dynamic relationship needs to be established among the elements of the space whereby the interactions with devices are considered a dynamic-process state. In this paper, a linear model of a smart space is presented using a state equation, where the two coefficient matrices L and H need to be defined to model the smart space, and the coefficient matrix L is used to determine the states of the devices; similarly, the situation of the smart space is determined using coefficient H. An algorithm is presented to make a linear model from the logical functions that are used to describe the system. This model is flexible in terms of the control of the smart-space environment because the environmental factors are represented by a matrix element. This linear smart-space model is helpful for the control of a context-aware system, and we use an example to illustrate the effectiveness of the proposed model.

1. Introduction

The mathematical modeling of a system is the most important step in the design of a control system, as it represents the behavior of a system in response to the changes of states and inputs. The following are the two modeling approaches for linear systems: the transfer-function approach and the state-space approach [1]. Out of the two approaches, the state-space approach is used to represent a dynamic system. The important feature of the state-space approach is the usage of first-order differential/difference equations to represent systems. The behavior of the system can be predicted by solving the differential/difference equations that are used for modeling.

A smart space is composed of a number of components such as sensors, actuators, and computing devices that enable the sensing and control of the environment. The goal of the smart space is to collect the environmental information and provide service automatically for user's comfort and safety [2]. In a smart space, users generally interact with various devices according to their needs, and these interactions can be considered a dynamic-process state. The next states of the devices can be described using the information regarding the current states of the devices and control inputs; in terms of the smart space, its situation can be described according to the current device states. The aim of this paper is to generalize the smart-space domain through linear modeling by using a state equation. This linear model describes the relations between the next states of devices and the current states of devices along with the control inputs and between the current states of devices and the situations. Several algorithms were proposed to implement context-aware systems using knowledge and resources such as production rules (if-then relationships), neural networks, support vector machines, fuzzy logic, Bayesian networks, and Hidden Markov Model [3–6]. In [7], the author proposed a unified and mathematically compatible method for logic-based intelligent system. However, this method requires special knowledge on mathematical logic and most of the deductions are tricky. Logic-based system used the logical function sets to show the relations between the states of devices and the control inputs and between the states of devices and the smart-space situation. The matrix expression is very convenient in logic inference because it converts the problem to solving linear algebraic equation. In this paper, we have presented an algorithm to convert these logical functions into a linear algebraic equation using the Sum of Products (SOP) canonical form and logic vector. The proposed smart-space linear model can be helpful for the control of a context-aware system because all of the variables are expressed by a matrix element and the relation between every variable is expressed by a coefficient matrix.

The rest of the paper is organized as follows: Section 2 presents an algorithm to convert a logic function into a linear algebraic equation; Section 3 illustrates the mathematical modeling of a smart space; and Section 4 presents an example to illustrate the effectiveness of the smart-space model, followed by the conclusion and a discussion of future work in Section 5.

2. Linear Algebraic Representation of Logic Function

Generally, a smart space can be described by logic expressions and logic related special knowledge needs to control the environment. If logic expressions are converted into algebraic equation, then control can be done effectively by manipulating matrix elements. To develop a linear model, we need conversion of logic expression into algebraic equations. Edwards uses a canonical form to express Boolean functions in matrix algebra [8]. This approach is different from "conventional" matrix algebra, as it requires the "unit" matrix $[A]_{\mathrm{UNIT}}$ for multiplication operations. Authors in [9] proposed the usage of a semitensor product to represent Boolean functions in an algebraic form. In this paper, we have proposed an algorithm using Sum of Products (SOP) canonical form and logic vector (ith column of an identity matrix) as logic value to represent the logic functions in the algebraic equations. Using this approach, the logic function has been represented similarly to its representation in conventional algebra, as follows:

$$y = D \cdot x, \tag{1}$$

where D is called the coefficient matrix that defines the logic function(s) and x and y are the input and output vectors, respectively.

2.1. Single Logic Function. In this section, a single logic function has been converted into a linear algebraic equation. Consider the following Boolean-logic function:

$$y = f(x_1, x_2, \ldots, x_n). \tag{2}$$

Any Boolean function can be expressed as a Sum of Products (SOP) in a canonical form, as follows:

$$\begin{aligned} &f(x_1, x_2, \ldots, x_n) \\ &\quad = (d_1 \cdot m_0 + d_2 \cdot m_1 + \cdots + d_{2^n} \cdot m_{2^{n-1}}). \end{aligned} \tag{3}$$

Equation (3) can be represented in a matrix form, as follows:

$$y = \begin{bmatrix} d_1 & d_2 & \cdots & d_{2^n} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}, \tag{4}$$

where $\begin{bmatrix} d_1 & d_2 & \cdots & d_{2^n} \end{bmatrix}$ is the coefficient matrix that holds the values of all of the minterms of the logic function, while the $\begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}$ vector holds all of the minterms. The order of the matrix and vector terms must be adhered to.

The ith column of an identity matrix (I_n) is used to represent the logic value in the logic equations [9]. Considering the identity matrix, $V = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ in terms of Boolean values. True (1) and False (0) are represented as $v_2^1 = \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]$ and $v_2^2 = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$, respectively. If there are n variables, then the Boolean function is a logical mapping from a set of $\{v_2^1, v_2^2\}^n$ to $\{v_2^1, v_2^1\}$.

Assuming that x_1 and x_2 are two logic variables that are represented as $x_1 = \left[\begin{smallmatrix} \bar{a} \\ a \end{smallmatrix}\right]$ and $x_2 = \left[\begin{smallmatrix} \bar{b} \\ b \end{smallmatrix}\right]$, with a and $b \in \{0, 1\}$, then the Kronecker product (⊗) of the two logic variables is calculated in the following way:

$$x_1 \otimes x_2 = \begin{bmatrix} \bar{a} \\ a \end{bmatrix} \otimes \begin{bmatrix} \bar{b} \\ a \end{bmatrix} = \begin{bmatrix} \bar{a}\bar{b} \\ \bar{a}b \\ a\bar{b} \\ ab \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \end{bmatrix}. \tag{5}$$

Equation (5) shows that the Kronecker product between two logic variables represents all of the minterms; similarly, the following can also be shown:

$$x = x_1 \otimes x_2 \cdots \otimes x_n = \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}. \tag{6}$$

The coefficient matrix elements $(d_1, d_2, \ldots, d_{2^n})$ consist of two values $\left[\begin{smallmatrix} \bar{d}_i \\ d_i \end{smallmatrix}\right]$, where $d_i \in \{0, 1\}$ and $i = 1, 2, \ldots, 2^n$ are the substituting values, and the dimension of the coefficient matrix D is equal to 2-by-2^n, where n is the number of logic variables:

$$D = \begin{bmatrix} \bar{d}_1 & \bar{d}_2 & \cdots & \bar{d}_{2^n} \\ d_1 & d_2 & \cdots & d_{2^n} \end{bmatrix}. \tag{7}$$

From (4), and using (6), it can be written as the following:

$$y = D \cdot x, \tag{8}$$

where $y = \left[\begin{smallmatrix} \bar{y} \\ y \end{smallmatrix}\right]$ with $y \in \{0, 1\}$. In the previously mentioned section, a single logic function is converted into an algebraic equation, whereas the next section will present multiple logic functions.

2.2. Multiple Logic Functions. Consider the following Boolean-logic functions:

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \ldots, x_n), \\ y_2 &= f_2(x_1, x_2, \ldots, x_n), \\ &\vdots \\ y_m &= f_m(x_1, x_2, \ldots, x_n). \end{aligned} \tag{9}$$

Equation (8) can be expressed as the SOP in a canonical form:

$$\begin{aligned} & f_1(x_1, x_2, \ldots, x_n) \\ &\quad = (e_{1,1} \cdot m_0 + e_{1,2} \cdot m_1 + \cdots + e_{1,2^n} \cdot m_{2^{n-1}}), \\ & f_2(x_1, x_2, \ldots, x_n) \\ &\quad = (e_{2,1} \cdot m_0 + e_{2,2} \cdot m_1 + \cdots + e_{2,2^n} \cdot m_{2^{n-1}}), \\ &\qquad \vdots \\ & f_m(x_1, x_2, \ldots, x_n) \\ &\quad = (e_{m,1} \cdot m_0 + e_{m,2} \cdot m_1 + \cdots + e_{m,2^n} \cdot m_{2^{n-1}}). \end{aligned} \tag{10}$$

Equation (10) can be represented by m linear equations akin to (4):

$$\begin{aligned} y_1 &= \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,2^n} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}, \\ y_2 &= \begin{bmatrix} e_{2,1} & e_{2,2} & \cdots & e_{2,2^n} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}, \\ &\vdots \\ y_m &= \begin{bmatrix} e_{m,1} & e_{m,2} & \cdots & e_{m,2^n} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}. \end{aligned} \tag{11}$$

All of the elements of the coefficient matrix and y_i in (10) are represented by $e_{i,j} = \left[\begin{smallmatrix} \bar{e} \\ e \end{smallmatrix}\right]$, $y_i = \left[\begin{smallmatrix} \bar{y} \\ y \end{smallmatrix}\right]$, with e, and $y \in \{0, 1\}$, and are expressed as follows:

$$\begin{aligned} \begin{bmatrix} \bar{y}_1 \\ y_1 \end{bmatrix} &= \begin{bmatrix} \bar{e}_{1,1} & \bar{e}_{1,2} & \cdots & \bar{e}_{1,2^n} \\ e_{1,1} & e_{1,2} & \cdots & e_{1,2^n} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}, \\ \begin{bmatrix} \bar{y}_2 \\ y_2 \end{bmatrix} &= \begin{bmatrix} \bar{e}_{2,1} & \bar{e}_{2,2} & \cdots & \bar{e}_{2,2^n} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,2^n} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}, \\ &\vdots \\ \begin{bmatrix} \bar{y}_m \\ y_m \end{bmatrix} &= \begin{bmatrix} \bar{e}_{m,1} & \bar{e}_{m,2} & \cdots & \bar{e}_{m,2^n} \\ e_{m,1} & e_{m,2} & \cdots & e_{m,2^n} \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{2^n-1} \end{bmatrix}. \end{aligned} \tag{12}$$

By applying the Khatri-Rao product ($*$) between all of the coefficient matrices, we can obtain the coefficient matrix E for (12) [10]. The dimensions of coefficient matrix E are equal to 2^m-by-2^n, where m is the number of logic functions and n is the number of logic variables, as follows:

$$\begin{aligned} & E = e_1 * e_2 * \cdots * e_m, \\ & E \\ &= [\mathrm{Col}_1(e_1) \otimes \mathrm{Col}_1(e_2) \otimes \cdots \mathrm{Col}_1(e_m) \;\; \mathrm{Col}_2(e_1) \otimes \mathrm{Col}_2(e_2) \otimes \cdots \mathrm{Col}_2(e_m) \;\; \cdots \;\; \mathrm{Col}_{2^n}(e_1) \otimes \mathrm{Col}_{2^n}(e_2) \otimes \cdots \mathrm{Col}_{2^n}(e_m)]. \end{aligned} \tag{13}$$

Applying the Kronecker product ($\otimes$) on the left side of (12),

$$y = \begin{bmatrix} \bar{y}_1 \\ y_1 \end{bmatrix} \otimes \begin{bmatrix} \bar{y}_2 \\ y_2 \end{bmatrix} \cdots \otimes \begin{bmatrix} \bar{y}_m \\ y_m \end{bmatrix}. \tag{14}$$

From (12), the usage of (6), (13), and (14) can be written as the following:

$$y = E \cdot x. \tag{15}$$

3. Linear Modeling of a Smart Space

Smart space is set up with the sensors and several devices. In this section, we have presented the linear model of a smart space. In a smart space, an inhabitant interacts with several devices through the combinational state of devices, and each device can be operated in two states. The next state of a device can be described by the logical function of

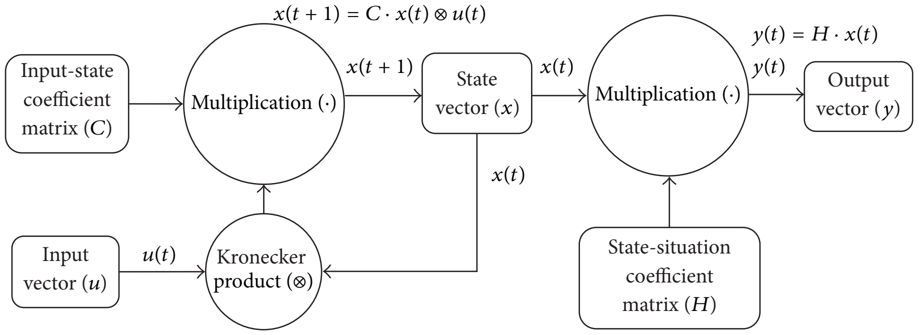

Figure 1: Block diagram of the proposed smart-space model.

the current states of the device and its input information about the environment. The situation of a smart space at any time is described by the combination of the states of the devices at that time. The sensor network in a smart space is used to collect the input data about the environment which is needed to represent a space state. The next states of the devices can be expressed as the following:

$$\begin{aligned} x_1(t+1) &= f_1(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ x_2(t+1) &= f_2(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \\ &\vdots \\ x_n(t+1) &= f_n(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)), \end{aligned} \tag{16}$$

where $f_i : V^{n+m} \to V$, $i = 1, 2, \ldots, n$, and $x_j \in V$, $j = 1, 2, \ldots, n$, are the states of the devices and $u_l \in V$, $l = 1, 2, \ldots, m$, are the inputs that represent the environment.

The situation of the smart space can be expressed as the following:

$$\begin{aligned} y_1(t) &= h_1(x_1(t), x_2(t), \ldots, x_n(t)), \\ y_2(t) &= h_2(x_1(t), x_2(t), \ldots, x_n(t)), \\ &\vdots \\ y_p(t) &= h_p(x_1(t), x_2(t), \ldots, x_n(t)), \end{aligned} \tag{17}$$

where $h_j : V^n \to V$, $j = 1, 2, \ldots, p$, are the logical functions; $x_j \in V$, $j = 1, 2, \ldots, n$, are the states of the devices; and $y_l \in V$, $l = 1, 2, \ldots, p$, are the situations.

Using the procedure described in Section 2, the multiple logical functions expressed by (16) and (17) can be converted into standard discrete-time dynamic systems, as follows:

$$\begin{aligned} x(t+1) &= C \cdot (x(t) \otimes u(t)), \\ y(t) &= H \cdot x(t), \end{aligned} \tag{18}$$

where the matrices C and H are called the coefficients of the logical functions of (16) and (17), respectively. One has $x(t+1) = x_1(t+1) \otimes x_2(t+1) \otimes \cdots x_n(t+1)$, $x(t) = x_1(t) \otimes x_2(t) \otimes \cdots x_n(t)$, $u(t) = u_1(t) \otimes u_2(t) \otimes \cdots u_m(t)$, and $y(t) = y_1(t) \otimes y_2(t) \otimes \cdots y_p(t)$.

Table 1: Logic rules for state change.

	Rules for state change
(i)	If $x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4 \wedge u_1 \wedge u_2 \wedge \neg u_3 \wedge \neg u_4 \wedge \neg u_5 = \text{True}$ then $x_1 = \text{True}$
(ii)	If $x_1 \vee x_2 \vee x_3 \vee \neg x_4 \vee x_5 \wedge u_1 \wedge \neg u_2 \wedge \neg u_3 \wedge \neg u_4 \wedge \neg u_5 = \text{True}$ then $x_2 = \text{True}$
(iii)	If $\neg x_1 \vee x_2 \vee x_3 \vee \neg x_4 \vee x_5 \wedge u_1 \wedge \neg u_2 \wedge \neg u_3 \wedge u_4 \wedge \neg u_5 = \text{True}$ then $x_3 = \text{True}$
(iv)	If $x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4 \vee x_5 \wedge u_1 \wedge \neg u_2 \wedge u_3 \wedge \neg u_4 \wedge \neg u_5 = \text{True}$ then $x_4 = \text{True}$

The input-state coefficient matrix (C) and the state-situation coefficient matrix (H) are matrix representations of the SOP of a logic variable that is used in logic expressions. Input (u), state (x), and output (y) vectors represent the possible control inputs, the states of the devices, and the situations in the smart space, respectively. A block diagram of the proposed linear model is shown in Figure 1.

4. An Example of a Smart Space

In this example, we assume that a smart space which is composed of a sensor network and four devices will be operated by sensor data. The input data about the environment u_1, u_2, u_3, u_4, and u_5 are formed on the basis of the sensor data and the four devices' states are considered as x_1, x_2, x_3, and x_4. The next states of the devices are determined based on the current states of the devices and the input information about the environment. The situation of a smart space at any time is dependent on the combinational states of the devices at that time. Two different logical functions are used to describe the relations between the information about the environment and device states and the device states and situations. All of the variables are considered Boolean logical variables ("ON," "OFF"). If the current states of the devices are known, then the next device states are updated according to the current states and input information about the environment [11]. The next states of the devices can be represented by (16) and we can express the function by logic rules (if-then relationship) as tabulated in Table 1.

x_4	x_3	x_2	x_1	u_5	u_4	u_3	u_2	u_1

Figure 2: The format variables that can be used in the minterm calculations.

These logic rules can be assumed as the following:

$$\begin{aligned}
x_1(t+1) &= \left(x_1(t) + \overline{x}_2(t) + \overline{x}_3(t) + x_4(t)\right) \cdot u_1(t) \\
&\quad \cdot u_2(t) \cdot \overline{u}_3(t) \cdot \overline{u}_4(t) \cdot \overline{u}_5(t), \\
x_2(t+1) &= \left(x_1(t) + x_2(t) + x_3(t) + \overline{x}_4(t)\right) \cdot u_1(t) \\
&\quad \cdot \overline{u}_2(t) \cdot \overline{u}_3(t) \cdot \overline{u}_4(t) \cdot \overline{u}_5(t), \\
x_3(t+1) &= \left(\overline{x}_1(t) + x_2(t) + x_3(t) + \overline{x}_4(t)\right) \cdot u_1(t) \\
&\quad \cdot \overline{u}_2 \cdot \overline{u}_3(t) \cdot u_4(t) \cdot \overline{u}_5(t), \\
x_4(t+1) &= \left(x_1(t) + \overline{x}_2(t) + \overline{x}_3(t) + x_4(t)\right) \cdot u_1(t) \\
&\quad \cdot \overline{u}_2(t) \cdot u_3(t) \cdot \overline{u}_4(t) \cdot \overline{u}_5(t).
\end{aligned} \tag{19}$$

There are four current states and five input variables on the right-hand side of (19). We consider that $2^9 = 512$, SOP for the expression of the logical function (19) into a canonical form. When we calculated the minterms, we used variable x_4 as MSB and u_1 as LSB. Figure 2 shows the variable minterm format.

Equation (19) can be expressed as the SOP in canonical form, as follows:

$$\begin{aligned}
x_1(t+1) &= m_3 + m_{35} + m_{67} + m_{99} + m_{131} + m_{163} \\
&\quad + m_{227} + m_{259} + m_{291} + m_{323} + m_{355} \\
&\quad + m_{387} + m_{419} + m_{451} + m_{483}, \\
x_2(t+1) &= m_1 + m_{33} + m_{65} + m_{97} + m_{129} + m_{161} \\
&\quad + m_{193} + m_{225} + m_{289} + m_{321} + m_{353} \\
&\quad + m_{385} + m_{417} + m_{449} + m_{481}, \\
x_3(t+1) &= m_9 + m_{41} + m_{73} + m_{105} + m_{137} + m_{169} \\
&\quad + m_{201} + m_{233} + m_{265} + m_{329} + m_{361} \\
&\quad + m_{393} + m_{425} + m_{457} + m_{489}, \\
x_4(t+1) &= m_5 + m_{37} + m_{69} + m_{101} + m_{133} + m_{169} \\
&\quad + m_{229} + m_{261} + m_{293} + m_{325} + m_{357} \\
&\quad + m_{389} + m_{421} + m_{453} + m_{485}.
\end{aligned} \tag{20}$$

Equation (20) can be represented in a matrix form as the following:

$$\begin{gathered}
\left[x_1(t+1)\right] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ \vdots \\ m_{511} \end{bmatrix}, \\
\left[x_2(t+1)\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ \vdots \\ m_{511} \end{bmatrix}, \\
\left[x_3(t+1)\right] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ \vdots \\ m_{511} \end{bmatrix}, \\
\left[x_4(t+1)\right] = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & \cdots & 0 \end{bmatrix} \cdot \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ \vdots \\ m_{511} \end{bmatrix}.
\end{gathered} \tag{21}$$

From (21), the first matrix on the right-hand side is the coefficient matrix that holds all of the minterms values of each function of (20), and the $\begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{511} \end{bmatrix}$ vector holds all of the 512 minterms of the logic functions of (20). All of the elements of the coefficient matrix are substituted by True (1) $= \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and False (0) $= \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. By applying the Khatri-Rao product ($*$) between all four of the coefficient matrices in (21), the following coefficient of matrix C can be obtained:

$$\begin{aligned} C &= c_1 * c_2 * \cdots * c_4, \\ C &\\ &= \left[\mathrm{Col}_1(c_1) \otimes \mathrm{Col}_1(c_2) \otimes \cdots \mathrm{Col}_1(c_4) \quad \mathrm{Col}_2(c_1) \otimes \mathrm{Col}_2(c_2) \otimes \cdots \mathrm{Col}_2(c_4) \quad \cdots \quad \mathrm{Col}_{512}(c_1) \otimes \mathrm{Col}_{512}(c_2) \otimes \cdots \mathrm{Col}_{512}(c_4)\right]. \end{aligned} \tag{22}$$

The dimension of the coefficient of matrix C is equal to 16-by-512. The minterm vector can be calculated by $x(t) \otimes u(t)$, where $x = x_1 \otimes x_2 \otimes \cdots \otimes x_4$ and $u = u_1 \otimes u_2 \otimes \cdots \otimes u_5$, as follows:

$$x(t) \otimes u(t) = \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{511} \end{bmatrix}. \tag{23}$$

By substituting $x_i(t+1)$ by $x_i = \left[\begin{smallmatrix} \overline{x}_i \\ x_i \end{smallmatrix}\right]$ with $x_i \in \{0, 1\}$, followed by the application of the Kronecker product ($\otimes$) on the left-hand side of (21), the following can be derived:

$$x(t+1) = \begin{bmatrix} \overline{x}_1 \\ x_1 \end{bmatrix} \otimes \begin{bmatrix} \overline{x}_2 \\ x_2 \end{bmatrix} \cdots \otimes \begin{bmatrix} \overline{x}_4 \\ x_4 \end{bmatrix}. \tag{24}$$

From the use of (22) by (21), (23) and (24) can be written in the following way:

$$x(t+1) = C \cdot (x(t) \otimes u(t)). \tag{25}$$

The situation of this smart space can be expressed by (17), and we can express the function by logic rules (if-then relationship) as shown in Table 2.

These logic rules can be assumed as the following:

$$\begin{aligned} y_1(t) &= x_1(t) \cdot \overline{x}_2(t) \cdot \overline{x}_3(t) \cdot \overline{x}_4(t), \\ y_2(t) &= (x_1(t) + x_2(t) + x_3(t)) \cdot \overline{x}_4(t), \\ y_3(t) &= (x_1(t) + x_2(t)) \cdot \overline{x}_3(t) \cdot \overline{x}_4(t), \\ y_4(t) &= \overline{x}_1(t) \cdot (x_2(t) + x_3(t) + x_4(t)). \end{aligned} \tag{26}$$

Similarly, a logical function that describes the situation can be represented by a linear algebraic equation in the following way:

$$y(t) = H \cdot x(t), \tag{27}$$

where H is the coefficient matrix and y and x are the output and input vectors, respectively. The dimensions of y, H, and x are 16-by-1, 16-by-16, and 16-by-1, respectively.

In this example, a total of nine variables (sum of device states and environmental information) are used. Using these nine variables, there are $2^9 = 512$ different input sequences available. These 512 input sequences are used to determine next states of devices. Then, smart-space situations can be calculated from combination of different devices states. We have calculated smart-space situation in two ways: firstly using logic rules and secondly using linear model. In logic rules method, we used rules for state change (Table 1) and rules for situation (Table 2) to infer the situation of smart space and considered the outcomes as manipulated data. In terms of the linear model, (25) and (27) represent the state equation and situation equation, respectively, and these matrix expressions generalize the logical operations that are convenient in logical inference. Control can be complete effectively by matrix calculations using this linear smart-space model. The situation of smart space can be inferred using Algorithm 1. Outcomes of Algorithm 1 are considered as linear-model data.

Using manipulated data, a plot was drawn for the "ON" states of the devices with a corresponding situation, while the linear-model data is presented for the purpose of comparison, as shown in Figure 3; every situation is composed of a set of device states. In Figure 3, different markers were used to show the "ON" state of each device. Our results show an overlap between the linear-model data and the manipulated data, proving that a linear model represents the logic function equally well and can be used to model a smart space. Using linear model control can be done easily by manipulating matrix elements.

5. Conclusion

We have proposed a smart-space linear model through the conversion of a logic function to a linear model. A smart space is typically represented by a logic function whereby smart devices operate in the smart space; in a state-space approach, the smart space is expressed as a linear model by the presented algorithm. This linear model describes the relations between the device states and inputs that represent the environment and between the device states and the situation of smart space. The example in this paper presents a linear model of smart space using four devices, each of which has two states and five control inputs based on the proposed model. In addition, the states of the devices and the corresponding situations are shown with different control inputs. Experimental result shows that outcomes using this linear model are the same as logic rules. Although the linear-model results are the same as logic functions, control can be done easily using linear model by manipulating the matrix elements. Moreover, if the infrastructure of smart space is changed, then it can be easily adopted because environmental factors are represented by matrix elements and the relation between every variable is expressed by the coefficient matrix. This model can be applied in an Internet of Things platform whereby correspondence between devices is necessary for a number of purposes. In

```
**Data**: input-state coefficient matrix $C$, state-output coefficient matrix $H$, at a given
time, List of input $I$, List of state $S$, input vector $u$, state vector $x$
**Result**: Situation $y$
$x, u$ initialization;
/* Next state calculation */
**for** $i = 1$ to $q$ **do**
    /* $q$ is the total no. of inputs in list of input $I$ */
    $u = u \otimes I_j$;
end
$x = C \times x \otimes u$;
**for** $j = 1$ to $l$ do
    /* $l$ is the total no. of inputs in list of input $I$ */
    $x = x \otimes S_j$;
end
$y = H \times x$;
```

Algorithm 1: Linear model (C, H, u, x).

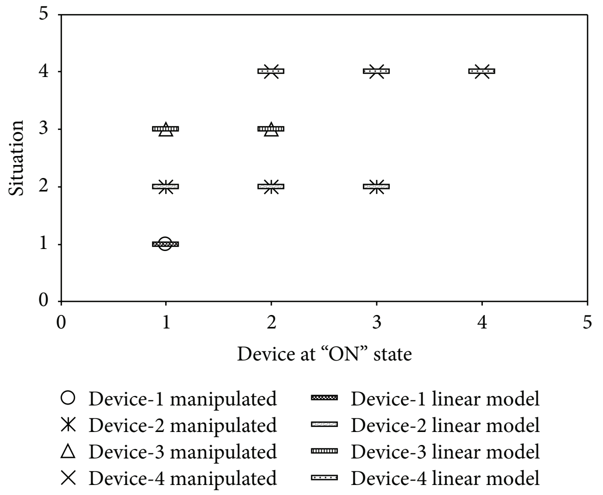

Figure 3: Representation of a situation with corresponding devices at "ON" states.

Table 2: Logic rules for situation.

	Rules for situation
(i)	If $x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4$ = True then y_1 = True
(ii)	If $x_1 \vee x_2 \vee x_3 \wedge \neg x_4$ = True then y_2 = True
(iii)	If $x_1 \vee x_2 \wedge \neg x_3 \wedge \neg x_4$ = True then y_3 = True
(iv)	If $\neg x_1 \wedge x_2 \vee x_3 \vee x_4$ = True then y_4 = True

this modeling, all of the logical variables are represented by the values of 1 or 0. In the future, k-value logic will be used to express the logic variable more effectively.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the Industrial Strategic Technology Development Program (10041788, Development of Smart Home Service Based on Advanced Context-Awareness) funded by the Ministry of Trade, Industry & Energy (MI, Korea). The work reported in this paper was conducted during the sabbatical year of Kwangwoon University in 2013.

References

[1] B. Friedland, *Control System Design: An Introduction to State-Space Methods*, Dover, New York, NY, USA, 2005.

[2] D. J. Cook and S. K. Das, *Smart Environments Technology, Protocols, and Applications*, Wiley-Interscience, Hoboken, NJ, USA, 2005.

[3] M. C. Mozer, "The neural network house: an environment that adapts to its inhabitants," in *Proceedings of the AAAI Spring Symposium on Intelligent Environments*, pp. 110–114, 1998.

[4] J.-Y. Hong, E.-H. Suh, and S.-J. Kim, "Context-aware systems: a literature review and classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009.

[5] H.-S. Park and S.-B. Cho, "A modular design of Bayesian networks using expert knowledge: context-aware home service robot," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2629–2642, 2012.

[6] M. H. Kabir, M. R. Hoque, K. Thapa, and S.-H. Yang, "Two-layer hidden Markov model for human activity recognition in home environments," *International Journal of Distributed Sensor Networks*, vol. 2016, Article ID 4560365, 12 pages, 2016.

[7] K. Truemper, *Design of Logic-Based Intelligent Systems*, John Wiley & Sons, Hoboken, NJ, USA, 2004.

[8] C. R. Edwards, "The logic of Boolean matrices," *The Computer Journal*, vol. 15, no. 3, pp. 247–253, 1972.

[9] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*, Springer, London, UK, 2011.

[10] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, The MIT Press, Cambridge, Mass, USA, 1982.

[11] J. L. Schiff, *Cellular Automata: A Discrete View of the World*, Wiley-Interscience, New York, NY, USA, 2008.