Illumination Modeling Method for Office Lighting Control by Using RBFNN

SUMMARY This paper represents an illumination modeling method for lighting control which can model the illumination distribution inside office buildings. The algorithm uses data from the illumination sensors to train Radial Basis Function Neural Networks (RBFNN) which can be used to calculate 1) the illuminance contribution from each luminaire to different positions in the office, 2) the natural illuminance distribution inside the office. This method can be used to provide detailed illumination contribution from both artificial and natural light sources for lighting control algorithms by using small amount of sensors. Simulations with DIALux are made to prove the feasibility and accuracy of the modeling method.

key words: office lighting, energy saving system, radial basis function neural networks

1. Introduction

Nowadays, in order to improve the control accuracy and power efficiency in lighting control of office buildings, many intelligent lighting control systems have been developed. In lighting systems, illumination control accuracy and energy saving efficiency are greatly dependent on the accuracy of illumination coefficients in the control algorithms. However, besides of the characteristic of luminaires and layout of offices, the luminosity is very sensitive to numerous factors such as building materials, furniture position, room decorations, etc. which make illumination modeling a bottleneck problem. To get more accurate data, many researchers choose feedback control by sensor measurement. Aghemo et. al conduct researches on a building automation and control system (BACs) which dim the luminaires in response to input from the illumination sensors [1]. Miki et al. have developed an intelligent lighting system using the Distributed Optimization by Resource Addition and Reduction (DORAR) method [2] which was capable of autonomous distributed control and was effective for comfortable lighting and energy saving [3]. Caicedo et al. use a distributed linear optimization algorithm to realize intelligent lighting control [4]. However the feedback control cannot avoid real-time lamp adjustment and always face the problem of slow convergence.

In our previous research [5]–[7], we introduced particle swarm optimization (PSO) into a wireless energy saving system for small sized offices to optimize the dimming ratio of luminaires. The system uses illumination sensors to measure the illumination condition inside the office. By real experiments, the algorithm was proved to be successful (save considerable energy as well as meet the target illuminance requirement) in controlling the luminaires in small office. We also improved the PSO algorithm for lighting control in deep-plan office [8], however, in practical applications, large amount of illumination sensors will be very expensive and difficult to maintain while less sensor number will reduce the accuracy of the illumination model. The research in this paper is aiming at developing proper strategy to provide precise illumination coefficients for lighting control algorithms by using small amount of illumination sensors.

Many programs for lighting simulation are developed which enable rough simulation for lighting projects and control strategy without real establishment of lighting systems [9]. Singhvi et al. have developed a utility-based building control strategy that optimizes the trade off between user comfort and reduction in operation cost using X10 system [10]. Ali et al. introduced a solution for energy saving in office building with DIALux [11]. Wen et al. proposed and demonstrated a wireless networked lighting system that satisfied both energy savings and user satisfaction using the RADIANCE Synthetic Imaging system [12] and SPOT™ [13]. However, simulators in these systems use theoretical models which often have some differences with the real environment.

As a popular method for solving real world industrial problems concerning functional prediction and system modeling, artificial neural networks (ANN) are being applied to many lighting systems. Wang et al. have developed a holistic and scalable ANN model which can represent the complex relationship between dimming ratios of luminaires and illuminance in target positions [14]. By training the ANN model using data from sensors in working tables, given the input of all the dimming ratios of luminaires, the output will be the measurements of the target sensors. However, this model does not consider natural illumination and is only applicable for artificial lighting control at night. Besides, the...
model can only calculate illuminance in limited positions (positions of training sensors), if the layout of any luminaire or table is changed, ANN training must be conducted again. Dong et al. have employed hierarchical radial basis function (HRBF) network to implement coarse-to-fine modeling of illumination environment so that the illuminance on continuous working plane can be calculated by measurements of limited sensors [15]. The lighting system then calculate the difference between real and target illuminance, and use a linear light transport model [16] to calculate how to adjust the dimming ratio of the luminaires. Since the HRBF model only calculate the integrated illuminance environment by using realtime sensor data, and cannot provide any information about contributions from each lamp to the target plane, it can only be used in feedback control.

RBFNN is one of the most common ANN structures used for function approximation [17]. By a sufficient set of training data, RBFNN is able to model the system with any desired amount of accuracy. In this research, RBFNN is introduced to model the illumination distribution so that the illuminance in any target area can be calculated without restriction. We use RBFNNs to represent the illuminance contribution from all the lamps to the working plane so that the illumination environment can be analyzed clearly. By using RBFNN instead of limited theoretical illumination models, the illumination distribution in any kind of office can be modeled regardless of their shape, decoration and lamp-type. Illumination coefficients for offices lighting control algorithms can be calculated accurately thus conquering the difficulty of environmental uncertainty in lighting industry.

2. Illumination Coefficient Modeling by RBFNN

2.1 Basic Principle

Figure 1 shows the basic principle of the control algorithm. For each target position, a virtual sensor is set to measure the illuminance. \( N_i \) represents the natural illuminance contribution to sensor \( S_i \) while the illumination coefficient \( a_{ij} \) represents the illuminance contribution to sensor \( i \) from lamp \( j \) when its dimming ratio is 100%. \( F_j \) is the power proportion of each lamp. So suppose there are \( n \) lamps \( \{L_1, L_2, \ldots, L_n\} \) and \( m \) sensors \( \{S_1, S_2, \ldots, S_m\} \) in the system, \( \{E_1, E_2, \ldots, E_m\} \) is the illumination of sensors, we can get the formula to calculate the total illuminance for each sensor:

\[
\begin{pmatrix}
E_1 \\
E_2 \\
\vdots \\
E_m
\end{pmatrix} = \begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{pmatrix} \begin{pmatrix}
F_1 \\
F_2 \\
\vdots \\
F_n
\end{pmatrix} + \begin{pmatrix}
N_1 \\
N_2 \\
\vdots \\
N_m
\end{pmatrix}
\]

(1)

2.2 RBFNN

To collect the training data for the proposed modeling method, the illumination contribution from each lamp to \( m \) different places (sensors) in the offices should be measured. To simplify the case, a unified height is set for the sensors. Thus the dimension of target positions can be reduced from 3 to 2 which make the size of the network input 2.

The structure of the neural networks in the proposed control algorithm is shown in Fig. 2. In general, RBFNN can be represented as:

\[
g(x) = \sum_{i=1}^{m} w_i \times \exp\left(-\frac{\|x - \mu_i\|^2}{b_1}\right) + b_2
\]

(2)

where \( x \) is input vector \((x=(x,y))\) in this paper as shown in Fig. 2, \( m \) is the number of radial basis functions, \( \mu_i \) is the center of each basis function (in Fig. 2 is \((x'_i,y'_i))\), \( b_1 \) is a given constant, \( w_i \) is the weight for each neuron and \( b_2 \) is a constant weight. The network contains 2 layers: a radial basis layer and a linear layer. The radial basis layer has \( m \) neurons whose centers have the same value with the training data. In each neuron the distance between the input vector and the center of the neuron is calculated and then multiplied by the bias. The result is then transferred by the radial basis function \( rb(x) = e^{-x^2} \).

In the linear layer, the output of the radial basis layer (a vector having \( m \) elements) is multiplied by the weight vector and added by the bias.

When implementing the training, we initialize the RBF network with \( m \) (which is same as pair number of the training data) neurons, whose centers \((\mu_i)\) in Eq. 2 are the same as the input training data. And then set \( b_1 \) to be 0.8326 which equals to the square root of \( ln(2) \), i.e. when the distance between input and the center of neuron is 1, the output of the radial layer will be 0.5. The smaller \( b_1 \) is, the
smoother the function will be. After $b_1$ and $\mu_b$ are determined, we can get the output of the radical layer, then we can get the weights and bias ($w_i$ and $b_2$ in Eq. 2) in the linear layer by solving a linear problem.

Two types of RBFNN are needed in the control algorithm, one is for calculating the illumination from luminaires, and the other is for calculating the natural illumination.

### 2.2.1 RBFNN for Illumination Coefficients

Totally, $n$ RBFNNs $\{Net_1, net_2, \ldots, net_n\}$ are needed to model the illumination distributions of $n$ lamps. The training data is a $n \times m \times 3$ matrix which contains the information of positions of $m$ training sensors (inputs) and the illumination contribution from each lamp to these sensors (outputs).

Once the RBFNNs are trained, they can be used to generate the illumination coefficients. For each $Net_j$, given the position of sensor $i$ ($x_i, y_i$), the output will be the illumination coefficient $a_{ij}$.

### 2.2.2 RBFNN for Natural Illumination

One RBFNN $Net_0$ is needed for calculating the natural illumination.

To adjust the natural illumination, there will be one illumination sensor $S_0$ in a fixed position inside the office. Illumination contributions from all the luminaires ($a_{0j}$ represents the illumination contribution from lamp $j$ to $S_0$) should be measured so that the natural illumination can be always calculated according to the lamp setting information $F_1, F_2, \ldots, F_n$ by:

$$N_0 = E_0 - \sum_{j=1}^{n} a_{0j} \times F_j$$  \hspace{1cm} (3)

The training data is a $m \times 3$ matrix which contains the information of positions of $m$ training sensors (inputs), and the normalized natural illumination of the $m$ sensors $\{N_1/N_0, N_2/N_0, \ldots, N_m/N_0\}$ (outputs).

Once the RBFNN is trained, it can be used to calculate the natural illuminations. Given the position of sensor $i$ ($x_i, y_i$) and the measurement of $S_0$, the output will be the natural illumination of Sensor $i$ $N_i$.

If more accurate data are needed, networks which represent different weather type and time can be trained. Network which fits certain condition can be chosen according to weather and time.

### 3. Illumination Distribution Modeling for Offices

Once all the networks are trained, training sensors are no longer needed. Only $S_0$ is need to adjust the intensity of natural illumination.

Suppose that a certain RBFNN $Net_0$ is chosen according to the time and weather condition. This $Net_0$ is actually equivalent to a function $f_0(x, y)$ which describe the shape of natural illumination distribution.

Suppose that $f(x, y)$ represents the illumination distribution inside the office, $f_i(x, y)$ represents the illumination distribution of each luminaire, it can be calculated by:

$$f(x, y) = f_0(x, y) \times N_0 + \sum_{j=1}^{n} f_j(x, y) \times F_j$$  \hspace{1cm} (4)

Thus as shown in Fig. 3 the RBFNNs can be integrated as a new neural network with higher dimension. With the input of $N_0, (x_i, y_i)$ (position of sensor $i$) and $(F_1, F_2, \ldots, F_n)$ (dimming ratio of lamps), the output will be the luminosity of sensor $i$ $E_i$. And each $Net_i$ in Fig. 3 has the structure described in Fig. 2.

### 4. Verification by Simulation

#### 4.1 Simulation Environment

DIALux is a illumination simulating software developed by the German Applied Light Technique Institute in order to gather armature data from companies and standardize the calculations made by using different methods. DIALux includes the armature and lamp information from many companies, such as Philips, Osram, SURYA, etc. in its database and supports many room shapes and allows the user to design their own. The program, which supports the addition of more than one piece of furniture to the place, also takes into consideration factors such as lighting level, reflection and glitter which can be seen on the furniture [9], [18].

To evaluate the performance of the control method and discuss the training sensor positions, we simulated a small office by DIALux to get training data for the RBFNN and test the real luminance pattern.

Figures 4 and 5 show the 3D standard view and layout plan of a small office model (5.4mx3.6m) with 9 lamps. The luminaire data and luminous emittance are shown in Fig. 6. The maximum power for each lamp was 118 w. The office model contained furniture such as desks and chairs which simulate the real office environments and increase the complexity of illumination distribution. 9 measurement points (virtual training sensors) were set in the models to generate training data for the RBFNNs. The heights of the sensors...
Fig. 4 3D standard view of the office model.

Fig. 5 Input protocol and layout plan of the office model.

Fig. 6 Luminaire data.

were set to be 0.71 m.

There was one illumination sensor on the wall (height 1.00 m) for natural illumination modeling. 108 × 108 positions which covered the marked estimation area in Fig. 5 (height 0.71 m) were measured by both net0 and DIALux in the following simulations.

In order to discuss the influence of training sensor positions to the RBFNN performance, we designed two training sensor distribution which were Training 1 and Training 2 shown in Fig. 7.

In Training 1, we divided the room into 4 × 4 = 16 areas and set one training sensor in each of the areas randomly.

In Training 2, we put 13 training sensors (9 sensors right under 9 lamps and 4 sensors in the 4 room corners.

The heights of the sensors were set to be 0.71 m in both Training 1 and Training 2.

4.2 Performance of RBFNN

4.2.1 RBFNNs for Illumination Coefficients

In the training, we set $b_1$ to be 0.8326 in Fig. 2. After training, the RBFNNs were able to calculate the illumination contribution from each lamp to any positions inside the office. We measured 108×108 positions inside the office using RBFNNs (measurement of position $i$: $E_t$) and compared these data to the DIALux data (measurement of position $i$: $E_r$).

To evaluate the performance we calculated the mean difference $\bar{D} = |E_t - E_r|$ and standard deviation $\sigma$ of the differences ($E_t - E_r$).

Table 1 shows the comparison of the two training methods. $\bar{D}$ and $\sigma$ are the mean value and standard deviation of all the positions. $\bar{D}'$ and $\sigma'$ are the values of the positions where the illumination is above 30 lx (in DIALux). It can be concluded that the second training method get much better performance than the first one.

Researches reveal that people perceive a change in desk-surface illuminance when variation width stays within +6% to -8% of current illuminance [19]. Since the error of the method is within 6%, it can be concluded successful in illumination modeling.

Comparing the modeling method with our previous work which use realtime sensor measurement to provide data for control [8], we can see that the number of required sensors for using RBFNN approximates to the number of luminaires, while in [8], the number of required sensors are the same as the number of target positions. So when the target positions are much larger than the luminaires, (such as classrooms), using RBFNN will reduce the amount of required sensors.

4.2.2 RBFNN for Natural Illumination

To check the performance of the RBFNN for natural illumination, we gathered natural illumination data from 9:00 to 17:00 in DIALux.

Table 2 shows the comparison of the two training methods for natural illumination. Since higher natural illumination will certainly cause bigger differences, besides of the mean differences, we also normalize the mean differences by the maximum luminosity in the room.
Table 1 Comparison of the two training methods (RBFNNs for luminaires). (Unit: lx)

<table>
<thead>
<tr>
<th>Lamp</th>
<th>Training 1</th>
<th></th>
<th>Training 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(D)</td>
<td>(\sigma)</td>
<td>(D)</td>
<td>(\sigma)</td>
</tr>
<tr>
<td>1</td>
<td>8.3</td>
<td>12.2</td>
<td>9.2</td>
<td>13.0</td>
</tr>
<tr>
<td>2</td>
<td>9.1</td>
<td>11.8</td>
<td>10.5</td>
<td>13.2</td>
</tr>
<tr>
<td>3</td>
<td>7.5</td>
<td>10.7</td>
<td>8.8</td>
<td>11.9</td>
</tr>
<tr>
<td>4</td>
<td>14.5</td>
<td>20.1</td>
<td>14.5</td>
<td>20.0</td>
</tr>
<tr>
<td>5</td>
<td>21.9</td>
<td>29.5</td>
<td>22.2</td>
<td>29.8</td>
</tr>
<tr>
<td>6</td>
<td>15.0</td>
<td>23.6</td>
<td>15.2</td>
<td>23.8</td>
</tr>
<tr>
<td>7</td>
<td>6.2</td>
<td>8.9</td>
<td>7.9</td>
<td>11.5</td>
</tr>
<tr>
<td>8</td>
<td>10.5</td>
<td>14.8</td>
<td>11.8</td>
<td>16.7</td>
</tr>
<tr>
<td>9</td>
<td>15.0</td>
<td>18.5</td>
<td>14.3</td>
<td>19.1</td>
</tr>
<tr>
<td>Avg.</td>
<td>12.0</td>
<td>16.7</td>
<td>12.7</td>
<td>17.7</td>
</tr>
</tbody>
</table>

Table 2 Comparison of the two training methods (RBFNNs for natural illumination).

<table>
<thead>
<tr>
<th>Time</th>
<th>Training 1</th>
<th>(D) (lx)</th>
<th>(D) (Normalized)</th>
<th>Training 2</th>
<th>(D) (lx)</th>
<th>(D) (Normalized)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:00</td>
<td>3.1</td>
<td>3.0</td>
<td>5.3%</td>
<td>14.9</td>
<td>14.9</td>
<td>4.8%</td>
</tr>
<tr>
<td>8:00</td>
<td>15.6</td>
<td>15.7</td>
<td>5.0%</td>
<td>14.9</td>
<td>14.9</td>
<td>4.8%</td>
</tr>
<tr>
<td>9:00</td>
<td>40.3</td>
<td>34.4</td>
<td>5.4%</td>
<td>67.5</td>
<td>67.5</td>
<td>5.0%</td>
</tr>
<tr>
<td>10:00</td>
<td>62.4</td>
<td>54.9</td>
<td>5.3%</td>
<td>67.5</td>
<td>67.5</td>
<td>5.0%</td>
</tr>
<tr>
<td>11:00</td>
<td>84.4</td>
<td>67.5</td>
<td>6.2%</td>
<td>67.5</td>
<td>67.5</td>
<td>5.0%</td>
</tr>
<tr>
<td>12:00</td>
<td>88.2</td>
<td>68.3</td>
<td>6.7%</td>
<td>68.3</td>
<td>68.3</td>
<td>5.2%</td>
</tr>
<tr>
<td>13:00</td>
<td>73.0</td>
<td>58.3</td>
<td>7.3%</td>
<td>58.3</td>
<td>58.3</td>
<td>5.8%</td>
</tr>
<tr>
<td>14:00</td>
<td>53.7</td>
<td>45.3</td>
<td>6.8%</td>
<td>45.3</td>
<td>45.3</td>
<td>5.8%</td>
</tr>
<tr>
<td>15:00</td>
<td>36.5</td>
<td>32.7</td>
<td>6.6%</td>
<td>32.7</td>
<td>32.7</td>
<td>5.9%</td>
</tr>
<tr>
<td>16:00</td>
<td>22.9</td>
<td>20.7</td>
<td>6.6%</td>
<td>20.7</td>
<td>20.7</td>
<td>6.0%</td>
</tr>
<tr>
<td>17:00</td>
<td>14.6</td>
<td>10.6</td>
<td>8.2%</td>
<td>10.6</td>
<td>10.6</td>
<td>6.0%</td>
</tr>
<tr>
<td>Avg.</td>
<td>47.7</td>
<td>37.3</td>
<td>6.8%</td>
<td>37.3</td>
<td>37.3</td>
<td>5.3%</td>
</tr>
</tbody>
</table>

Fig. 8 Scenarios for simulations.

4.2.3 Performance of the Integrated Neural Network

As shown in Fig. 8, 4 scenarios were designed for simulation. For each simulation, we used the integrated neural network to calculate the theoretical estimation in different positions, and compared them with the DIALux calculation to check the performance of the integrated network.

Figure 9 and 10 show the comparison of modeling results with the DIALux of the 4 scenarios after Training 1 and Training 2. Table 3 compares the two training positions. Besides of the mean differences, we also normalized the differences with the real illumination (DIALux value), the differences were well controlled under 4%.

From the simulation results, it can be concluded that the performance of Training 2 is much better. This is because that for each lamp, the related RBFNN is relevant to its illumination distribution curve, it can be concluded that the position with highest luminosity is right under the lamp. It can be noticed that the training sensor group should at least includes \(n\) sensors which have the same position with the \(n\) lamps to make sure that the maximum values of the luminosity curves are included in the training data (or the calculation under the lamps will be definitely lower than the real luminosity because the adjacent training values are low). Besides, training sensors should also be placed near the edge of the room to make sure the area near walls can be modelled well.

4.3 Performance in Control Algorithm

4.3.1 Without Natural Illumination

As introduced in Sect. 1, we have developed a PSO control algorithm for office lighting [5]–[7] (discussed in our previous research) which can control the illumination inside of-
To control the illumination, this method set several target sensors inside the office, and then use PSO to minimize the energy use while keeping the illumination around the target luminosity.

Simulations were made to test the performance of the PSO algorithm.

According to the discussion in the last section, the RBFNNs trained by the second way (13 training sensors) has a better performance, so we will use the RBFNNs trained by the second way.

There were 4 target sensors as shown Fig. 11 (Positions: Sensor 1 (1.7m, 0.6m), Sensor 2 (2.4m, 3.0m), Sensor 3 (3.8m, 3.0 m), Sensor 4 (3.2m, 0.6m)) All the target luminosity were set to be 600 lx. Table 4 shows the results of 10 simulations.

Figure 12 shows the luminance distribution of one simulation (Simulation 3 in Tab. 4). The luminance of target working area was well controlled above 600 lx while the rest part of the office was dark to save the energy.

Figure 13 shows the luminance of 4 target sensors and the configuration of each lamp.

Since the results were the theoretical values calculated by using the RBFNNs data, the real luminance pattern should also be considered. To get the real luminance pattern,
we used the final dimming result in DIALux simulator to get the luminance pattern.

Figure 14 shows the comparison of the theoretical luminance pattern and the real luminance pattern in the same case with Fig. 12. The mean difference was 27.79 lx while the standard deviation was 22.41 lx. The measurements of the 4 target sensors in DIALux were 572 lx, 631 lx, 602 lx and 611 lx, while the differences with the measurements calculated by RBFNN data were 31.3 lx, 30.7 lx, 1.8 lx and 7.2 lx.

4.3.2 Considering Natural Illumination

Since our control algorithm also consider natural illumination and get use of natural illumination to save the energy, changing of natural illumination during daytime was also simulated to test the performance of the control algorithm.

Figure 15 shows the illumination distribution of 10:00. By harvesting the natural illumination, the control algorithm was able to save 71.7% of the energy which was 26.1% higher than the average performance without natural illumination harvesting.

Figure 16 shows the comparison of the theoretical luminance pattern and the real luminance pattern in the same case with Fig. 15. The mean difference was 47.94 lx while the standard deviation was 37.20 lx.

Figure 17 shows the luminance of target sensors and the configurations of each lamp.

Figure 18 shows the relationship between natural illumination, energy saving proportion which proves that the modeling method helps the control algorithm to make full use of natural illumination for energy saving.
use of natural illumination inside the office and save extra energy. The changing of mean difference and the standard deviation of the difference of the theoretical luminance pattern and the real luminance pattern are also shown in Fig. 18. The mean differences ranged from 0.9 lx to 4.2 lx, while the standard deviation of the differences ranged from 0.7 lx to 7.9 lx. The differences were well controlled within a proper range.

5. Conclusions and Future Works

This paper presents an illumination modeling method by using RBFNN. By training the RBFNNs using the office illumination data, the RBFNNs are able to model the office environment well and generate proper illumination coefficients. An integrated neural network is built for calculation of the luminosity inside the offices according to the dimming ratios of luminaires, natural illumination intensity and time. A small office (9 lamps) was built in DIALux for simulations of the proposed method. The simulation result by using the RBFNNs are compared with DIALux calculations to check the modeling performance in small office during which two different strategies for placing the training sensors were compared. The simulation results have found a proper suggestion for training sensors and prove that the proposed modeling method can successfully model the illumination distribution inside small office (mean differences under 5%). Simulations also prove that the modeling method can provide illumination data with high quality for lighting control algorithms.

However, the algorithm different does not consider the continuity of the changing of sunlight direction which influence on the natural illumination distribution inside the office. Methods considering the continuous change of sunlight should be developed to have better performance on daylight modeling.

At present, in this modeling method, illumination distribution is simplified in a unified height. Methods of 3d distribution modeling which provide much more precise calculation may be considered in future studies.

Besides, online training for the neural network by real-time sensor data is also a potential method to improve the flexibility and accuracy in the future.

Acknowledgments

This research receives support from Kitakyushu city as an Environmental Future Technology Development project.

References

wa Si was born in Hefei, China, on May 6th, 1987. She received a Bachelor degree of Science from Fudan University (China) in 2007 and a Master degree of Engineering from Waseda University in 2010. She is presently a Ph.D. student in the Graduate School of Information, Production and Systems, Waseda University. She has worked on optimization algorithms and wireless sensor network.

Xun Pan was born in Zhenjiang, China, on February 23th, 1989. He received a Bachelor degree of Science from Nanjing University (China). He is presently a Master student in the Graduate School of Information, Production and Systems, Waseda University. He has worked on optimization algorithms and wireless sensor network.

Harutoshi Ogai was born in Kitakyushu, Japan, on March 1st, 1952. He received a Ph.D. degree of Engineering from Tokyo Institute of Technology in 1997 and is presently a professor in the Graduate School of Information, Production and Systems, Waseda University. His research area includes process control, process modeling and simulation, process analysis and data mining, automotive control and bridge diagnosis. He is also member of SICE, ISCIE, ISIJ, JSAI and IPSJ.

Katsumi Hirai was born in Tokyo, Japan, on February 10th, 1944. He received a Master degree of Engineering from University of Tokyo in 1968 and is presently the president of Hakutsu Technology Corporation.


Tansheng Li was born in Shandong, China, on January 18, 1984. He is acquiring Ph.D. course in engineering in Graduate School of Information, Production and System in Waseda University. He is working at multi-hop wireless sensor network of bridge health monitoring.