

A Study on the Spontaneous Ignition and the Estimation of the Thermal Conductivity of the Coal Stockpile Container Using HMM(Hidden Markov Model)

Min-Seok Jie¹, Seung-Hun Kim², Chan-Syb Yeum³ and Won-Hyuck Choi^{4*}

²Department of Aeronautical Systems Engineering, 236-49, Gomseom-ro, Nam-myeon, Taean-gun., 32158 Chungcheongnam-do, Republic of Korea

³People i Co.,Ltd., 49, Techno 8-ro, Yuseong-gu, Daejeon, Republic of Korea

^{1,4}Department of Avionics Engineering Hanseo University, 236-49, Gomseom-ro, Nam-myeon, Taean-gun, 32158 Chungcheongnam-do, Republic of Korea

²kimsh753@naver.com, ³csyeum@people-i.co.kr,

{¹jiems, ^{4*}choiwh}@hanseo.ac.kr

Abstract

The damage caused by spontaneous combustion of coal stockpiles is proportional to the increase in the use of low carbon in coal-fired power plants. The spontaneous ignition of the coal stockpile causes not only serious economic loss but also environmental impact. In order to prevent spontaneous ignition, a cooling ball device is used and a prediction is made using HMM (Hidden Markov Model) algorithm. To prevent the occurrence of spontaneous ignition in advance.

Keywords: Hidden Markov model, Spontaneous Ignition, Coal Stockpile, Cooling Ball, Nitrogen, Prediction

1. Introduction

The damage caused by spontaneous combustion of coal stockpiles often occurs in proportion to the increase in the amount of low carbon used in coal-fired power plants. It is not only a serious economic damage, but also a typical example of environmental pollution caused by harmful gas power plants.

In the related industry, physical or chemical solutions are proposed to solve the problem of low self-ignition spontaneous ignition, but the cost and effect of expensive facility construction and operation are not ensured and the field is suffering from difficulties. A coal-fired 500-MW thermal power plant will have approximately six 500-ton capacity coal reservoirs, five of which will be supplied with normal coal, and the other will have a reserve coal reserve and is operated with low-fidelity. The problem of spontaneous ignition is getting bigger because coal is put outdoors and dust is blown to pollute the environment of the workplace and a part of raw material is lost by the wind.

In addition, the gas generated in the spontaneous ignition process may cause gas explosion if not released to the outside, and spontaneous ignition in the storage facility may induce dust explosion. In this way, safety management for prevention of spontaneous ignition is very important in the space where coal is stored and measures against fire occurrence are needed.

The spontaneous ignition model of coal stockpile was carried out in Salinger considering the rate of oxidation reaction by natural convection, the compression flow

Received (October 7, 2017), Review Result (December 19, 2017), Accepted (February 26, 2018)

* Corresponding Author

inside and outside the coal reservoir, and the oxygen concentration for 2 - dimensional steady - state analysis. Moghtaderi conducted a two-dimensional transient analysis considering the oxidation rate due to natural convection, the compression flow inside and outside the coal reservoir, and the oxygen concentration [3].

Domestic coal-fired power plants operate coal stockpiles but have not developed or studied coal-fired power plants, and reference reports on coal-fired power plants themselves are all. Based on the Moghtaderi model, HMM algorithm is applied to estimate the coal stockpile temperature after using the cooling ball.

Although the domestic coal-fired power plant operates a coal stockpile, there has been no development or research on the evolution work, and a reference report of the coal-fired power plant itself is all about the report. Based on the Moghtaderi model, we apply the HMM algorithm to predict the temperature of the coal stockpile after using the cooling ball.

2. Design Theory

Figure 1 show an example of the Markov process. The model presented describes a simple model for the stock market index. In this model, the three statuses: Downloads, Height, and Ignition, and the three index observations are unchanged. This model is a finite state automaton, and it has a random transition between states. For instance, we can easily determine whether the condition sequence that generated these observations is Lowness-Height by using the up-down observation sequence. The probability of a sequence is simply the result of a transition. $0.2 \times 0.3 \times 0.3$.

Figure 2 shows an example of how an older model can be extended to HMM. The new model now allows all the observed symbols to be released in each state in a finite probability. This change allows us to express the model more expressive and to express our intuition better. In this case, the bull market can have both good days and bad days, but there is something better. The biggest difference is that the state sequence is now 'hidden' because if the observed sequence goes up and down, we cannot accurately describe the state sequence that generated it. However, you can calculate the probability that the model will create a sequence and what state sequences have generated the readings. The following three sections describe common calculations that can be performed on HMM.

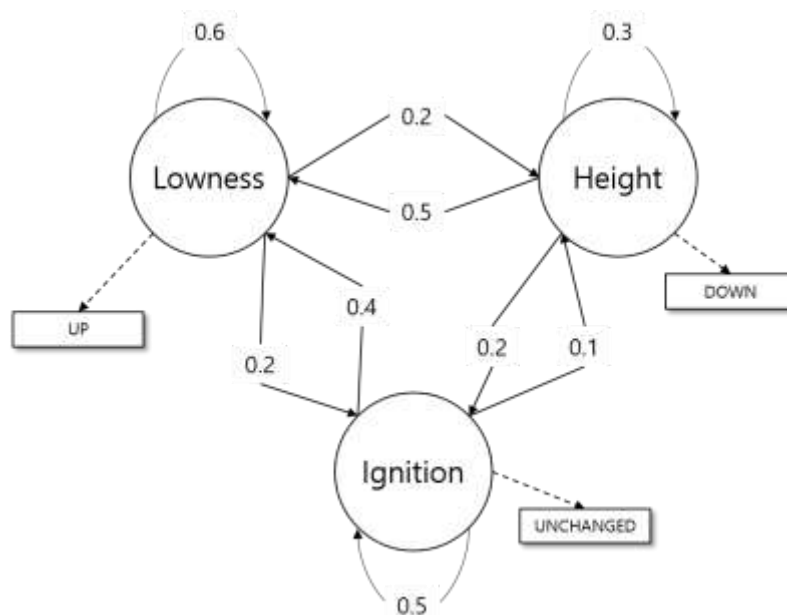


Figure 1. Markov Process

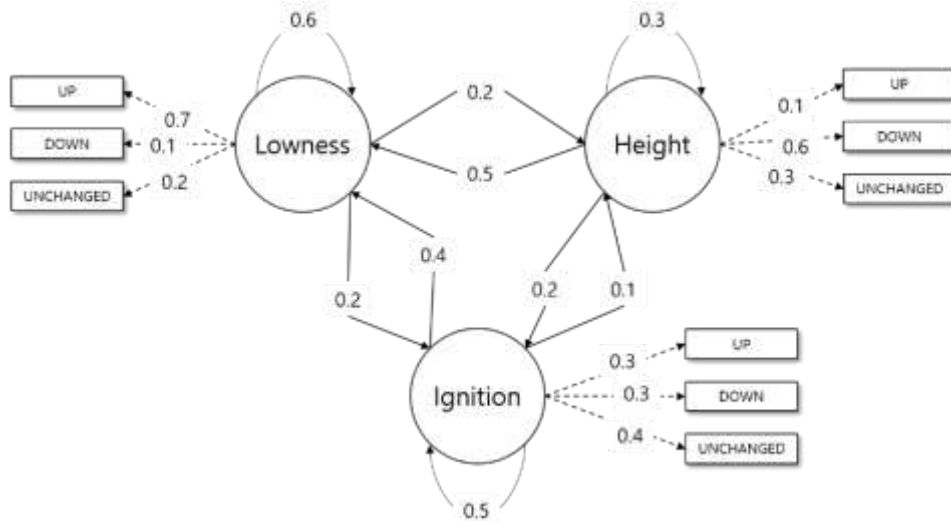


Figure 2. Hidden Markov Model

The official definitions of HMM are as follows:

$$\lambda = (A, B, \Pi) \quad (1)$$

S is the primary set of alphabets, and V is the observation alphabet set.

$$S = (s_1, s_2, \dots, s_N) \quad (2)$$

$$V = (v_1, v_2, \dots, v_M) \quad (3)$$

We define Q to be a fixed state sequence of length T, and corresponding observations O:

$$Q = (q_1, q_2, \dots, q_N) \quad (4)$$

$$O = (o_1, o_2, \dots, o_N) \quad (5)$$

A is a transition array, storing the probability of state j following state i. Note the state transition probabilities are independent of time:

$$A = [a_{ij}], a_{ij} = P(q_t = s_j | q_{t-1} = s_i) \quad (6)$$

B is the observation array, storing the probability of observation k being produced from the state j, independent of t:

$$B = [b_i(k)], b_i(k) = P(X_t = V_k | q_{t-1} = s_i) \quad (7)$$

π is the initial probability array:

$$\Pi = [\Pi_i], \Pi b_i = P(q_1 = s_i) \quad (8)$$

Two assumptions are made by the model. The first, called the Markov assumption, states that the current state is dependent only on the previous state, this represents the memory of the model:

$$P = (q_t | q_1^{t-1}) = P(q_t | q_{t-1}) \quad (9)$$

The independence assumption states that the output observation at time t is dependent only on the current state, it is independent of previous observations and states:

$$P = (o_t | o_1^{t-1}, q_1^t) = P(o_t | o_t) \quad (10)$$

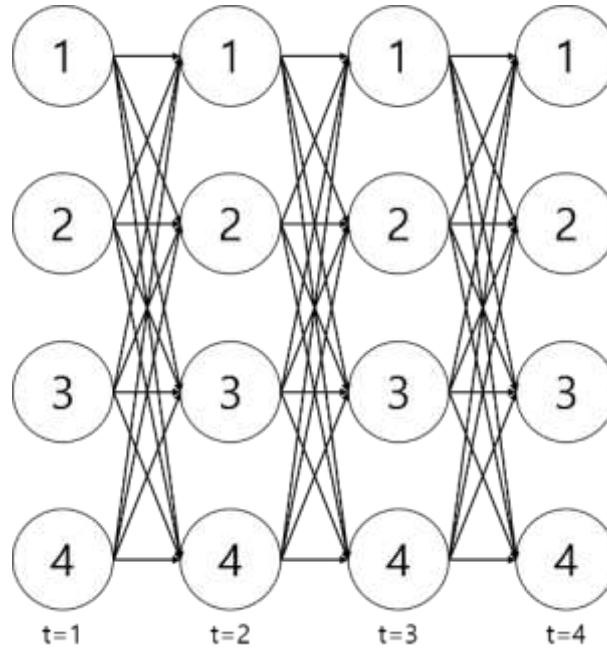


Figure 3. A Trellis Algorithm

2.1. Evaluation

Given a HMM, and a sequence of observations, we'd like to be able to compute $P(O|\lambda)$, the probability of the observation sequence given a model. This problem could be viewed as one of evaluating how well a model predicts a given observation sequence, and thus allow us to choose the most appropriate model from a set.

The probability of the observations O for a specific state sequence Q is:

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t | q_t, \lambda) = b_{q_1}(o_1) \times b_{q_2}(o_2) \cdots b_{q_T}(o_T) \quad (11)$$

and the probability of the state sequence is:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \quad (12)$$

so, we can calculate the probability of the observations given the model as:

$$P(O|\lambda) = \sum_Q P(O|Q, \lambda) P(Q|\lambda) = \sum_{q_1 \cdots q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \cdots a_{q_{T-1} q_T} b_{q_T}(o_T) \quad (13)$$

This result allows the evaluation of the probability of O , but to evaluate it directly would be exponential in T .

A better approach is to recognize that many redundant calculations can be performed by directly assessing Equation 13, therefore reducing the complexity of caching calculations. We implement the cache as a status grid at each stage and compute the cached values (α) for each state as a sum for all states in the previous stage. α is the probability of a state of being in a partial sequence of observations,

with values of s_{i-1} , s_i , and s_{i+1} . It can be visualized as shown in Figure 3. We define the following prospective variables.

$$\alpha_t(i) = P(o_1 o_2 \cdots o_t, q_t = s_i | \lambda) \quad (14)$$

so, if we work through the trellis filling in the values of α the sum of the final column of the trellis will equal the probability of the observation sequence. The algorithm for this process is called the forward algorithm and is as follows:

1. Initialisation:

$$\alpha_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N \quad (15)$$

2. Induction:

$$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(o_{t+1}), 1 \leq t \leq T - 1, 1 \leq j \leq N \quad (16)$$

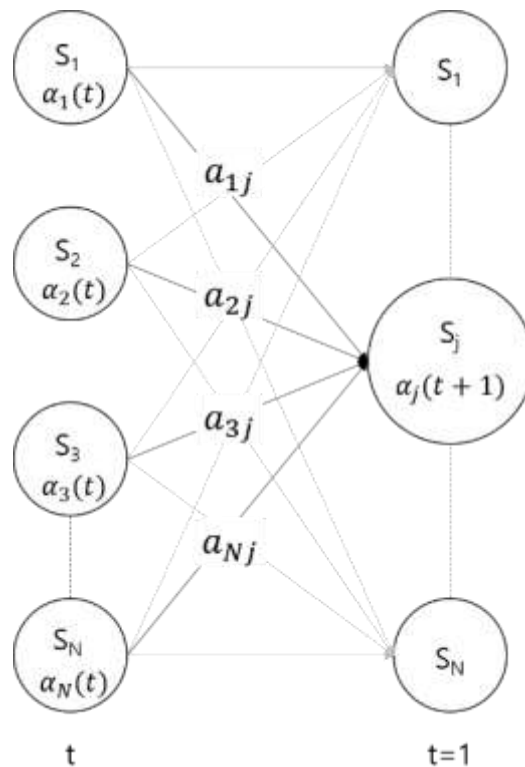


Figure 4. The Induction Step of the Forward Algorithm

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (17)$$

The induction step is the key to the forward algorithm and is depicted in Figure 4. For each state s_j , $\alpha_j(t)$ stores the probability of arriving in that state having observed the observation sequence up until time t .

It is apparent that by caching α values the forward algorithm reduces the complexity of calculations involved to $N^2 T$ rather than $2T N T$. We can also define an analogous

backwards algorithm which is the exact reverse of the forwards algorithm with the backwards variable:

$$\beta_t(i) = P(o_{t+1}o_{t+2} \cdots o_T | q_t = s_i | \lambda) \quad (18)$$

as the probability of the partial observation sequence from $t + 1$ to T , starting in state s_i .

2.2. Decoding

The aim of decoding is to discover the hidden state sequence that was most likely to have produced a given observation sequence. One solution to this problem is to use the Viterbi algorithm to find the single best state sequence for an observation sequence. The Viterbi algorithm is another trellis algorithm which is very similar to the forward algorithm, except that the transition probabilities are maximised at each step, instead of summed. First, we define:

$$\delta_1(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1 q_2 \cdots q_t = s_i, o_1, o_2 \cdots o_t | \lambda) \quad (19)$$

as the probability of the most probable state path for the partial observation sequence.

The Viterbi algorithm and is as follows:

1. Initialisation:

$$\delta_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N, \psi_1(1) = 0 \quad (20)$$

2. Recursion

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) b_{ij}] b_j(o_t), 2 \leq t \leq T, 1 \leq j \leq N \quad (21)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) b_{ij}] b_j(o_t), 2 \leq t \leq T, 1 \leq j \leq N \quad (22)$$

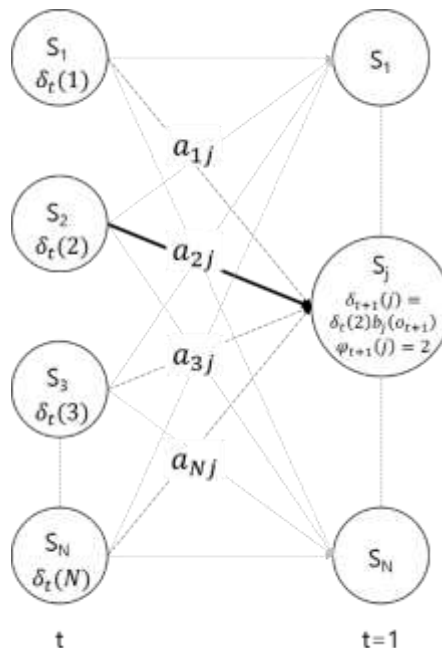


Figure 5. The Recursion Step of the Viterbi Algorithm

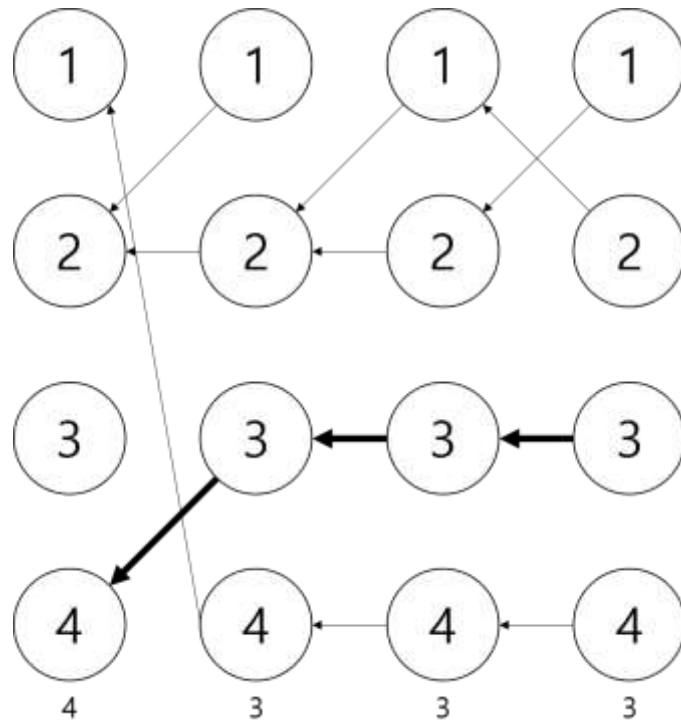


Figure 6. The Backtracking Step of the Viterbi Algorithm

3. Termination

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{23}$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)] \tag{24}$$

4. Optimal state sequence backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T - 1, T - 2, \dots, 1. \tag{25}$$

The recursion step is illustrated in figure 5. The main difference with the forward algorithm in the recursions step is that we are maximizing, rather than summing, and storing the state that was chosen as the maximum for use as a back pointer. The backtracking step is shown in 6. The backtracking allows the best state sequence to be found from the back pointers stored in the recursion step, but it should be noted that there is no easy way to find the second-best state sequence.

2.3. Learning

Given a set of examples from a process, we would like to be able to estimate the model parameters $\lambda = (A, B, \pi)$ that best describe that process. There are two standard approaches to this task, dependent on the form of the examples, which will be referred to here as supervised and unsupervised training. If the training examples contain both the inputs and outputs of a process, we can perform supervised training by equating inputs to observations, and outputs to states, but if only the inputs are provided in the training data then we must use unsupervised training to guess a model that may have produced those observations. In this section we will discuss the supervised approach to training, for a discussion of the Baum-Welch algorithm for unsupervised training see.

3. Structural Analysis Results

Figure 7 is a flow chart of applying heat generated from a coal stockpile to an HMM. Estimates are estimated using the collected data, and learning proceeds through feedback. As shown in the graph, the predicted value of the temperature that occurs after a certain period through the variable values of the coal tar stockpile is considered, considering the temperature characteristics before the spontaneous ignition occurs at the cooling ball position or at each point. In addition, it is expected that the temperature value generated after digestion of natural radiant fire using cooling ball before a certain temperature occurs.

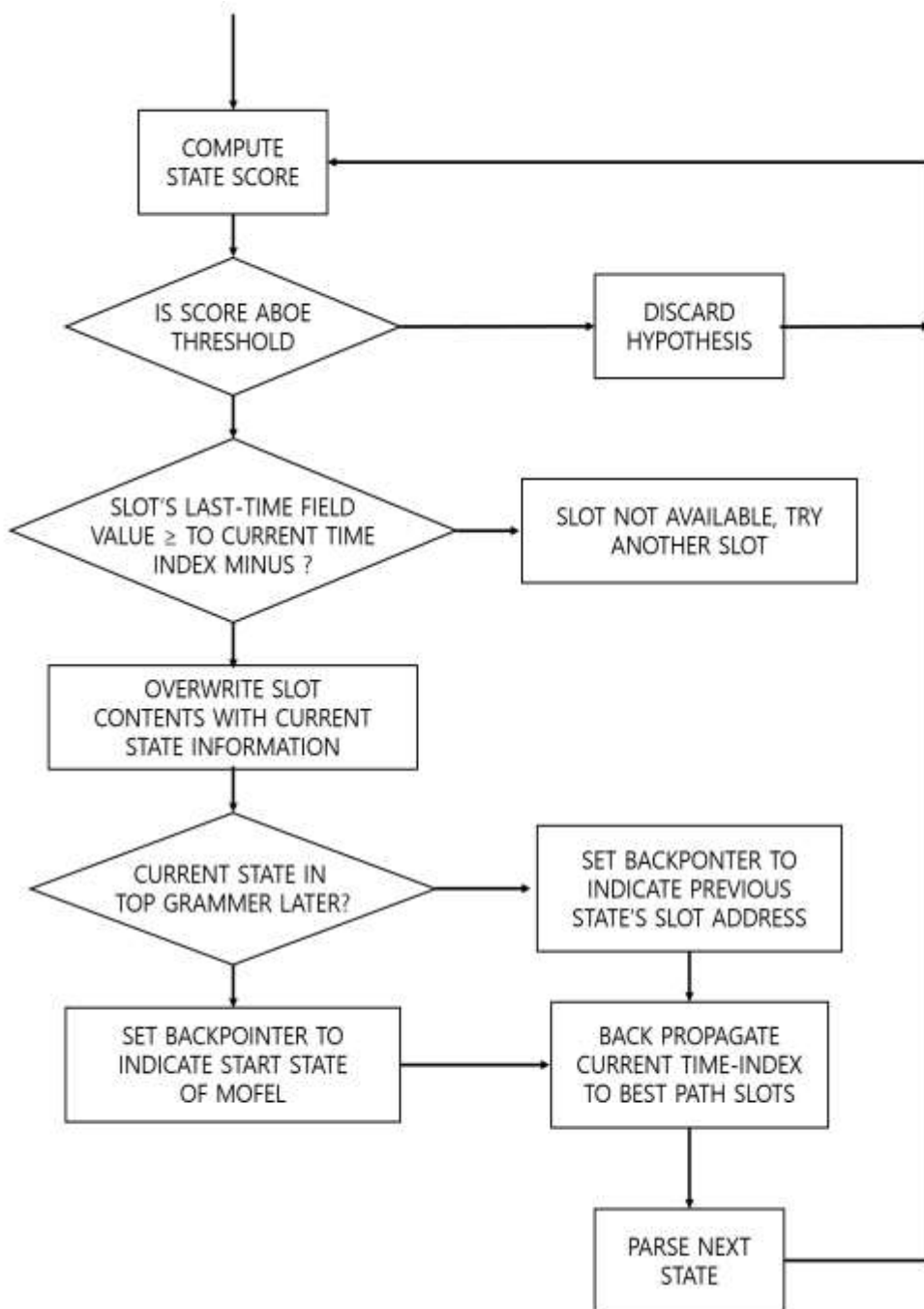


Figure 7. Hidden Markov Model Flowchart

4. Conclusion

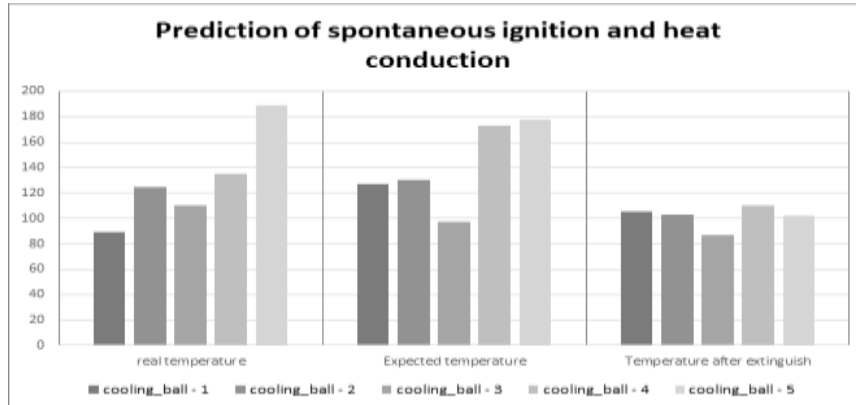


Figure 8. Estimated Temperature of Coal Stockpile and Ambient Temperature of Cooling Ball

In this study, the temperature data of the coal stockpile were based on the values obtained from the simulation values, and it is considered that a lot of data is needed about the point of spontaneous ignition occurring in the domestic thermal power plant. The HMM-based prediction method proposed in this study will be used as an efficient method to prevent spontaneous ignition of coal tar stockpile. Estimation of the prevention of spontaneous ignition is needed in consideration of the future meteorological variability and the surrounding complexity.

References

- [1] Huang et. al. Spoken Language Processing. Prentice Hall PTR.
- [2] L. Baum et. al. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. Annals of Mathematical Statistics, 41:164–171, 1970.
- [3] A. Markov. An example of statistical investigation in the text of Eugene onyegin, illustrating coupling of tests in chains. Proceedings of the Academy of Sciences of St. Petersburg, 1913.
- [4] A. McCallum and K. Nigam. A comparison of event models for naive bayes classification. in AAAI-98 Workshop on Learning for Text Categorization, 1998.
- [5] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of IEEE, 1989.

Authors



Min-Seok Jie, he graduated from the Ph.d degree in the department of avionics Engineering from Korea Aerospace University in 2006. he is Hanseo University Avionics Engineering Professor in to the present in 2008.



Seung-Hun Kim, he is a master's degree in Aeronautical System Engineering, Hanseo University. He graduated from the Department of Aeronautical and Mechanical Engineering, Hanseo University. He is studying Kalman filters, embedded systems, and smart network communications.



Chan-Sub Yeum, he earned an electronics degree from Chungbuk National University in 2008.



Won-Hyuck Choi (Corresponding author), he received the Ph.D. degree in avionics from Korea Aerospace University, Korea. He was a Professor of smart network at Doowon Technical University College, Korea. urrently, he is Professor of avionics at Hanseo University, Korea, where he has been since 2014. His present research interests include embedded system, home network.