A Hidden Semi-Markov Model for Predicting Production Cycle Time Using Bluetooth Low Energy Data

Karishma Agrawal*, Supachai Vorapojpisut

Department of Electrical and Computer Engineering, Faculty of Engineering, Thammasat School of Engineering, Thammasat University, Thailand

Received 02 March 2023; received in revised form 09 June 2023; accepted 10 June 2023

DOI: https://doi.org/10.46604/aiti.2023.11678

Abstract

This study proposes a statistical model to characterize the temporal characteristics of an entire production process. The model utilizes received signal strength indicator (RSSI) data obtained from a Bluetooth low energy (BLE) network. A hidden semi-Markov model (HSMM) is formulated based on the characteristics of the production process, and the forward-backward algorithm is employed to re-estimate the probability distribution of state durations. The proposed method is validated through numerical, simulation, and real-world experiments, yielding promising results. The results show that the Kullback-Leibler divergence (KLD) score of 0.1843, while the simulation achieves an average vector distance score of 0.9740. The real-time experiment also shows a reasonable accuracy, with an average HSMM estimated throughput time of 30.48 epochs, compared to the average real throughput time of 33.99 epochs. Overall, the model serves as a valuable tool for predicting the cycle time and throughput time of a production line.

Keywords: Bluetooth low energy, received signal strength indicator, hidden semi-Markov model, learning problem

1. Introduction

Production is a systematic and organized process that involves a series of interconnected working areas or stages. As shown in Fig. 1, the production process includes a specific set of tasks or operations that transform raw materials into finished products. To ensure that the production process is completed on time, it is critical to have accurate and up-to-date product status as well as location information. Many industries now use barcodes or RFID [1-2] to track products in production areas. However, the main disadvantage of employing RFID or barcodes is the possibility of human errors. For example, data loss or inaccuracy may occur if the RFID tag/barcode is not read correctly or if a worker scans the wrong RFID tag/barcode. This work uses Bluetooth low energy (BLE) technology to collect received signal strength indicator (RSSI) and timestamp data, which provides the proximity of products being manufactured. Even though the scope of BLE detection is larger than that of barcode/RFID, the collected data from BLE devices usually suffers from signal strength fluctuation and noisy environments. Consequently, the application of BLE tracking is much more limited compared to barcode/RFID.

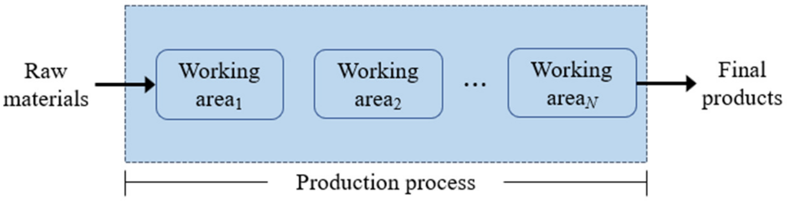


Fig. 1 General example of a production process

[*] Corresponding author. E-mail address: karishma.agra@dome.tu.ac.th

English language proofreader: Chih-Wei Chang

The key performance indicators (KPIs) of a production process can be defined by using temporal behaviors [3-4]. Examples of KPIs for production processes are cycle time and throughput time. Cycle time [3] refers to the time it takes to finish a cycle in a particular operation. In the meantime, throughput time [4] is the time it takes to process a certain material from raw material to a final product. Based on the KPIs, manufacturers can identify areas for improvement, reduce costs, and optimize productivity. Therefore, this information is of great significance for manufacturers who attempt to produce high-quality products in the shortest timeframe to remain competitive.

The traditional method of manually monitoring products to calculate cycle time had various limitations, such as the potential for human errors and limited data collection. At present, manufacturers have adopted a precise and efficient approach to analyzing historical data for cycle time calculation, that is, to use barcode/RFID [1-2] and statistics. The queuing theory is a widely used and traditional approach to analyzing and predicting the temporal properties of production processes. This theory states that production tasks arrive, await maintenance, and then exit once the process is complete. However, queuing models [5] usually consider the timing of each station independently and do not capture the dynamic nature of the whole production process.

The main objective of this study is to predict cycle time and throughput time in a production line using RSSI values. To achieve this, hidden semi-Markov models (HSMM) are used to analyze the RSSI and timestamp data. The paper presents an algorithm to determine the model parameters. The study contributes to estimating both cycle and throughput time by using RSSI values. Furthermore, it is a new approach compared to previous studies [6] that mainly focused on tracking and monitoring activities. This paper provides a more specific and comprehensive version of the work [7] previously presented, including detailed explanations and additional experiments.

The paper is structured as follows: Section 2 provides a background on BLE technology and introduces the concept of HSMM. While section 3 elaborates on the problem and presents mathematical solutions based on RSSI-based data. In Section 4, three experiments are conducted in numerical, simulation, and real-world settings. Section 5 analyzes three case studies to demonstrate the practical application of the proposed method. Finally, Section 6 discusses and concludes how the cycle time of a product can be computed with the mathematical solutions presented in this paper.

2. Backgrounds

This section provides an overview of BLE technology and introduces the concept of HSMM (Hidden Semi-Markov model). It presents the fundamental characteristics of BLE and its practical applications. Furthermore, it discusses the potential use of HSMM as a solution to enhance the capabilities of BLE networks in comprehending the attributes of the production process.

2.1. Bluetooth low energy (BLE) technology

BLE [6-7] technology offers wireless connectivity with low cost, low power consumption, and simple installation. Hence, it’s a popular choice for tracking applications. A BLE network consists of two types of devices: BLE scanners and BLE tags. Typically, BLE tags are battery-powered devices that periodically broadcast their IDs, packet-embedded data, and RSSI data [8]. BLE scanners detect and collect these signals along with a timestamp and send the information to the server. RSSI is a measurement that indicates the strength of the signal between a BLE tag and a scanner. Additionally, higher values represent a stronger and more dependable connection. Decibels (dBm) is the standard unit of measurement for RSSI.

There are two major problems in BLE technology to monitor products during production. Firstly, the RSSI can fluctuate significantly, even if the distance between the BLE scanner and tag remains unchanged, due to interference, random noise, and multipath effects [8]. Secondly, the inconsistent receiver sensitivity of each device [8] limits the communication range and generates random and insufficient RSSI data from the scanner. To illustrate these problems, Fig. 2 demonstrates the correlation between the RSSI values and distance obtained from an experiment conducted with WeMos R32 (ESP32) boards.

Various studies have been conducted to resolve the issue of RSSI fluctuation in object-tracking applications. Two popular methods for indoor localization are trilateration and fingerprinting, which utilize RSSI values obtained from multiple BLE scanners to estimate the 2D coordinates of a BLE tag [8]. However, these methods may not be suitable for production environments due to their large coverage area. Proximity detection presents a simpler approach, which estimates an object's location by detecting the presence of a BLE tag near a BLE scanner [8]. Nonetheless, it may have limitations when multiple BLE scanners give conflicting results.

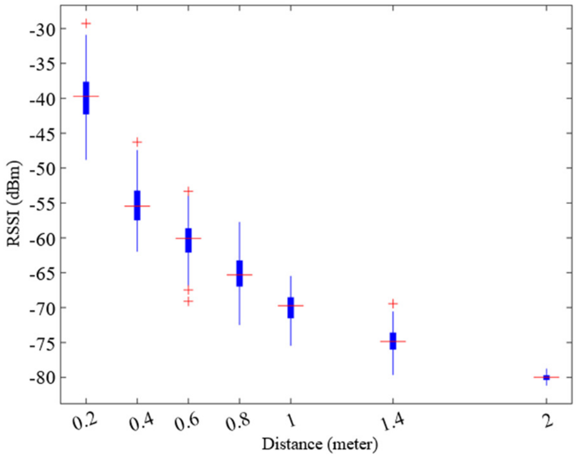


Fig. 2 RSSI vs Distance of BLE devices

Recent studies have shown that the hidden Markov model (HMM) has the potential to explain indoor trajectory with RSSI data. For example, Han et al. [9] used the BLE signal from the smartwatch and implemented an HMM algorithm to determine the user’s indoor location and the most likely sequence based on previous RSSI data. Similarly, Arslan et al. [10] utilized real-time BLE RSSI data from the construction site and HMM to identify semantic trajectories for extracting worker movement awareness to improve safety.

However, conventional HMM approaches do not specify the dwell time of the object in each state, which is not likely to happen in practical scenarios [11-12]. The HSMM has been proposed as an extension of the conventional HMM, including the modeling of the duration of each state. As a result, HSMM has been successfully applied in indoor tracking [11-13] and various other domains, including human-robot collaboration (HRC) [14] in assembly scenarios and machine condition recognition [15]. Despite these advancements, there is insufficient literature specifically focusing on understanding and predicting temporal characteristics in production settings.

2.2. *Hidden semi-Markov model* (*HSMM*)

The HMM [9-11] is a double-layer stochastic process that uses the Markov property to model a sequence of hidden states and observed outputs. The first layer represents the underlying and hidden states of the process, which are not directly observable. The second layer symbolizes the emission states, which are related to the observable and measurable states. The concept of HMM comprises five components, including state (S), initial state probability (π), transition probability matrix (A), observation state (O), and emission probability matrix (B).

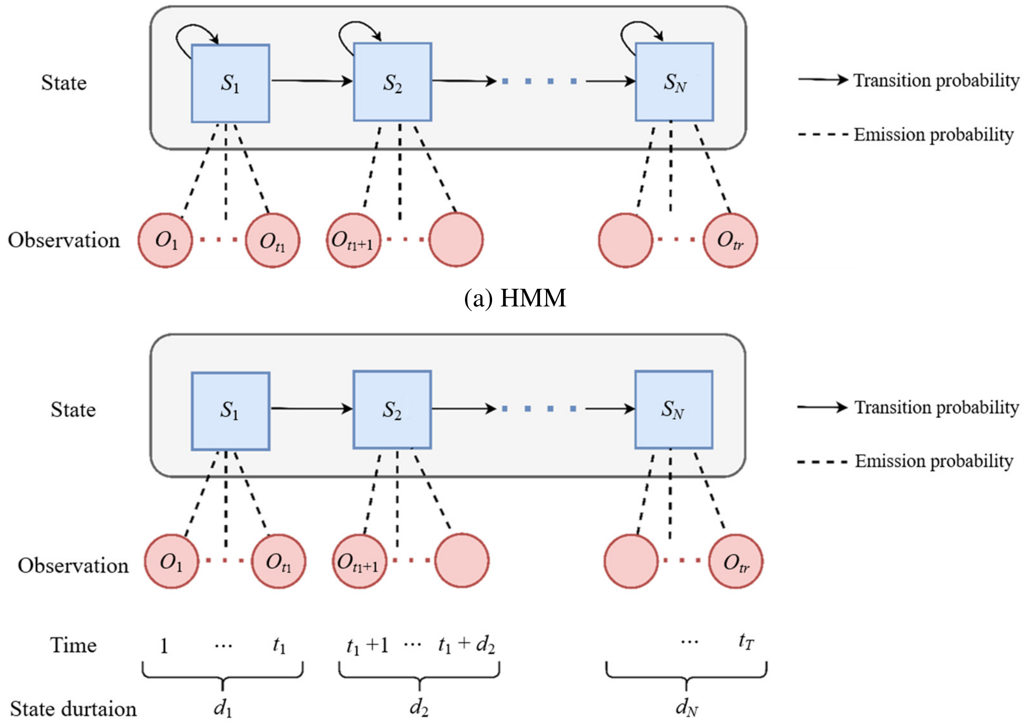
The emission probability matrix (B) defines the probability of an observation O_t at any given time instance t to determine a state $S \in \{1,2,\cdots,N\}$. The transition probability matrix A represents the probability of moving from one hidden state to another. These transition probabilities are to calculate the probability of states given a sequence of observations. The Markov property means that the probability of transitioning from one state to another only depends on the current state. And the HMM uses the Markov property as the prediction of the next state based on the current state. Since RSSI information being used as observation O may not imply the exact location of a device, this paper considers various stages of production, such as material preparation, process, and completion as the set of hidden states S.

The transition probability matrix A in an HMM is time-independent, which is not suitable for modeling temporal behaviors. The HSMM [14-17] addresses this issue by introducing the variable duration d, which signifies the time in each state. Fig. 3 illustrates two primary characteristics of an HSMM that distinguish it from an HMM [13]:

(1) The probability of self-transition is assumed to be zero, i.e., $a_{ii} = 0$ no return to the same state.

(2) The transition from a single state depends upon the sojourn duration or the time spent in that state.

These properties of the HSMM allow for the representation of variable state durations, making it possible to estimate the duration or cycle time of each stage in the production process which conventional HMM fails to achieve.



(a) HMM

(b) HSMM

Fig. 3 Graphical representation of HMM and HSMM

In this study, the BLE scanning pattern is analyzed by using a discrete time approach, where an epoch is defined as $t \in \{1,2,\cdots,\infty\}$. To simplify computations, the duration of each state is treated as a discrete random variable and bounded by an integer value $D > 1$. The probability of duration for state i have given d is represented by $p_i(d)$, where $i \in \{1, N\}$ and $d \in \{1, D\}$.

$$p_i(d) = P_r(d|S_t = i) \tag{1}$$

The constraint for $p_i(d)$ is that the sum of $p_i(d)$ from $d = 1$ to D is equal to 1 for all states i [7]. There are three basic problems with the HSMM framework used in academic works [7, 13]:

(1) Likelihood problem: This involves computing the likelihood of an observed sequence O given an HSMM parameter λ.

(2) Decoding problem: This refers to the problem of finding the optimal state sequence given an observed sequence O and an HSMM parameter λ.

(3) Learning problem: This involves estimating an HSMM parameter λ from an observed sequence O. It re-estimates the distribution of transitions, emission probabilities, and state durations.

The primary objective of this paper is to address the learning problem associated with determining the probability distribution for state durations, which is formulated based on the characteristics of the production process. As for the forward-backward algorithm, it is employed to re-estimate the probability distribution of state durations using a learning problem.

3. Problem Statement

This study considers a BLE system with a BLE tag on each product and a BLE scanner in each working area to collect data on the movement of products. The main objective of this study is to use HSMM concepts to create a mathematical model that can predict the cycle time of production based on the duration characteristics of each working area. This study formulates the product tracking problem as the HSMM model $\lambda = \{A, B, \pi, p\}$, where A represents the transition matrix with duration probability matrix p for each state, B signifies the emission matrix, and π is the initial state probability. The following KPIs are studied in the paper:

(1) Cycle time: Cycle time is the time it takes to complete a single task in a specific working area during the production process. The calculation of the cycle time can be expressed as:

$$CT_i = ET_i - ST_i \tag{2}$$

where CT_i represents the average cycle time for a product to complete its operations in the working area i. ST_i is the time when the operation begins, and ET_i is the time when it ends.

(2) Throughput time: The throughput time is the total time required for a product to move through the production process, and it represents:

$$\text{Throughput time} = \sum_{i=1}^{N} CT_i \tag{3}$$

where CT_i represents the cycle time for a product to complete its operations in the working area i.

The following assumptions are made for a production process based on how raw materials move in the production line to produce finished products [7].

(1) The working areas (stations) are represented as the hidden states $S \in \{1, 2, \cdots, N\}$. Each manufactured product is equipped with a specific tag ID, and a BLE tag is attached to it. BLE scanners scan these BLE tags and capture RSSI values. These RSSI values at time t are collected by multiple scanners and used as an observation $O_t = [RSSI_{1,t}, RSSI_{2,t}, \cdots, RSSI_{N,t}]$.

(2) The emission probability B consists of $b_i(O_t) = P(O_t | S_t = i, \mu_i, \sum_i)$ are assumed to be fixed but may be updated periodically based on the collected RSSI data [11-12]:

$$b_i(O_t) = \sum_{m=1}^{M} \mathcal{N}\left(O_t \middle| \mu_i^m, \Sigma_i^m\right) \tag{4}$$

where μ_i^m and $\sum_i^m$ are the mean and covariance for the m^{th} Gaussian component for observed RSSI values at the state i, respectively. The Gaussian distribution is denoted by $\mathcal{N}$. To simplify the calculation, the assumption is made that the observations are independent across time [12-13]. Therefore, $b_i(O_{t1:t2})$ can be expressed as the product of the probabilities of observing each RSSI data point from epoch $t1$ to $t2$, and it can be expressed as:

$$b_i(O_{t1:t2}) = \prod_{t=t1}^{t2} b_i(O_t) \tag{5}$$

(3) Define $A(i, j)$ to represent the probability of moving from area i to area j as the BLE tag moves through areas sequentially from 1 to *N*. The calculation of $A(i, j)$ can be simplified as:

$$A(i, j) = \begin{cases} 1 \; ; j = i + 1 \\ 0 \; ; \, elsewhere \end{cases} \tag{6}$$

(4) The initial state probability π_i is set to 1, indicating that every sequence will always begin at state 1.

This study focuses on the learning problem of the HSMM framework to capture the cycle time of the production process. To re-estimate the duration probability of each state, it is necessary to solve likelihood problems using a forward-backward procedure $P(O|\lambda)$ with two subdivided variables: forward and backward. In HSMM, the forward variable $\alpha_t(i)$ represents the likelihood of being in state i given the partial observations $O_{1:t}$ up to time t, for a given HSMM parameter λ [13].

$$\alpha_t(i) = P\left(O_{1:t}, S_i \ ends \ at \ t \middle| \lambda\right) = \sum\nolimits_{d=1}^{\min(t,D)} \alpha^*_{t-d}(i)\, p_i(d)\, b_i\left(O_{t-d+1:t}\right) \tag{7}$$

$$\alpha^*_t(i) = P\left(O_{1:t}, S_i \ begins \ at \ t+1 \middle| \lambda\right) = \sum\nolimits_{i=1}^{N} \alpha_i(i)\, \alpha_{j,i} \tag{8}$$

where $t = 1, \cdots, T$, $i = 1, \cdots, N$, $\alpha^*_t(i)$ is the joint probability of obtaining the observation sequence up to time t, and entry to state i begins at the next time $t + 1$, given the model λ. The calculation of the proximity likelihood for a location is performed using the following formula:

$$\widehat{S_t} = arg \max_{1<i<N} \alpha_t(i) \tag{9}$$

The definition of the HSMM backward variable $\beta^*_t(i)$ can be found in formula [13]:

$$\beta_t(i) = P\left(O_{t:T} \middle| S_i \ begins \ at \ t, \lambda\right) = \sum\nolimits_{d=1}^{T-t,D} \beta^*_{t+d}(i)\, p_i(d)\, b_i\left(O_{t:t+d-1}\right) \tag{10}$$

$$\beta^*_t(i) = P\left(O_{t:T} \middle| S_i \ ends \ at \ t-1, \lambda\right) = \sum\nolimits_{j=1}^{N} \alpha_{i,j}\, \beta_t(j) \tag{11}$$

where $t = T, \cdots, 1$, $i = 1, \cdots, N$, $\beta^*_t(i)$ is the probability of the partial observation sequence from time t to the end, given that the system leaves the state i at the previous time $t - 1$ and with a model λ. The re-estimation of the duration probability $\hat{p}_i(d)$ matrix, based on the forward and backward variables, is expressed as [13]:

$$\hat{p}_i(d) = \frac{\sum_{t=1}^{T-d+1} \alpha^*_{t-1}(i)\, p_i(d)\, b_i\left(O_{t:t+d-1}\right) \beta^*_{t+d}(i)}{\sum_{d=1}^{D} \sum_{t=1}^{T-d+1} \alpha^*_{t-1}(i)\, p_i(d)\, b_i\left(O_{t:t+d-1}\right) \beta^*_{t+d}(i)} \tag{12}$$

The outline of the learning problem used in the paper is as follows [7].

Algorithm 1: re-estimation of $p_i(d)$

```
Input: $A$, $B$, coupled sequences $(S^k, O^k)$; $k = 1,2, \cdots, K$
Training:
Set the values of $p_i(d)$ with uniform distribution
Initialize the $\hat{P}_i(d)$ matrix
Loop for $k = 1, \cdots, K$
    Loop for $t = 1, \cdots, T$
        Calculate $\alpha_t(i), \alpha^*_t(t), \beta_t(i), \beta^*_t(t)$
        Compute the $\hat{P}_i(d)$
        $\hat{P}_i(d) = \hat{P}_i(d) + \hat{p}_i(d)$
    $p_i(d) = \hat{P}_i(d)/T$
Output: $p_i(d)$
```

The $p_i(d)$ parameter in the HSMM represents the probability of the system staying in the state i for a duration of d time units. In other words, the $p_i(d)$ distribution of state i governs the cycle time for that specific stage of the production process. It is important to note that the $\hat{p}_i(d)$ Eq. (12) is computed from the entire sequence. To predict temporal properties, a sequence is constructed by sampling durations from the transition probabilities and $\hat{p}_i(d)$. This allows for the estimation of cycle times and throughput time. Overall, the HSMM can be modeled from collected RSSI and timestamp data in a production process, and the $\hat{p}_i(d)$ parameter plays a key role in estimating these KPIs.

4. Experiments

Three studies were conducted to evaluate the performance of the proposed algorithm based on numerical, simulated, and real-world experiments. Furthermore, the evaluation aimed to validate the effectiveness of the algorithm across various scenarios and provide empirical evidence of its capabilities.

4.1. Numerical experiment

In this experiment, 1000 state/observation sequences were based on given HSMM parameters (π, A, B, P). Each coupled sequence contains two types of data: observed (RSSI) signals and state sequences. HSMM parameters were defined as follows:

(1) The number of working areas N is 7.

(2) The maximum duration $D = 8$ epochs.

(3) Probability distribution matrix $P = \{p_i(d)\}$ for the system to leave the state i after d duration is given by.

$$P = \begin{bmatrix} 0 & 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0.1 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0.1 & 0.2 \\ 0 & 0 & 0.3 & 0.4 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.2 & 0.2 & 0.2 & 0 \\ 0 & 0.4 & 0.4 & 0.2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13}$$

(4) The values of RSSI are computed using a Gaussian mixture model with mean and variance:

$$\mu = \begin{bmatrix} -50 & -75 & -100 & -100 & -100 & -100 & -100 \\ -78 & -55 & -80 & -100 & -100 & -100 & -100 \\ -95 & -78 & -56 & -87 & -100 & -100 & -100 \\ -100 & -100 & -70 & -50 & -79 & -100 & -100 \\ -100 & -100 & -100 & -78 & -54 & -80 & -100 \\ -100 & -100 & -100 & -100 & -79 & -50 & -80 \\ -100 & -100 & -100 & -100 & -100 & -100 & -53 \end{bmatrix} \tag{14}$$

$$\sigma = \begin{bmatrix} 5 & 2 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 2 & 1 & 1 & 1 \\ 1 & 1 & 3 & 5 & 2 & 1 & 1 \\ 1 & 1 & 1 & 2 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 4 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 3 \end{bmatrix} \tag{15}$$

The sequence generation is outlined using Algorithm 2. Then, Algorithm 1 is used with the generated data to estimate the probability distribution matrix $\hat{p}$.

Algorithm 2: sequence generation

```
    **Input:** $\lambda = A, \mu, \sigma, \pi, P, N$ is the number of states and $M$ is RSSI observations
```

```
Output: Hidden state $S$ and RSSI observations $O$
for $i = 1 \; to \; N$
    if $i == 1$
        $S_{selected}$ = Select a state based on the given $\pi$ initial probabilities
    else
        $S_{selected}$ = Select a state using the previous state and $A$ probability
    end
        $durtaion_{selected}$ = Based on the $S_{selected}$ and $P$ probabilities, choose the duration
    for $k = 1 \; to \; durtaion_{selected}$
        for $m = 1 \; to \; M$
            $O(m, k)$ = Generate random value from $\mu$ and $\sigma$ with $S_{selected}$
        end
    end
    $S_{duration}(1: durtaion_{selected}) = S_{selected}$
    $S = [S, S_{duration}]$
    $O = [O, o]$
end
```

4.2. Simulated experiment

The MATLAB Bluetooth toolbox enables users to create a simulation environment of BLE devices. Fig. 4 shows the 2-D coordinates of the BLE scanners with moving BLE tags. This study uses MATLAB Bluetooth Toolbox to simulate BLE transmission to find RSSI values when tags are moved along the given trajectory.

One thousand state/observation sequences are generated from the movement of BLE tags in the simulated environment. The BLE tags are advanced randomly according to the selected duration. Fig. 5 provides an example of the observed RSSI values and state in a simulated sequence.

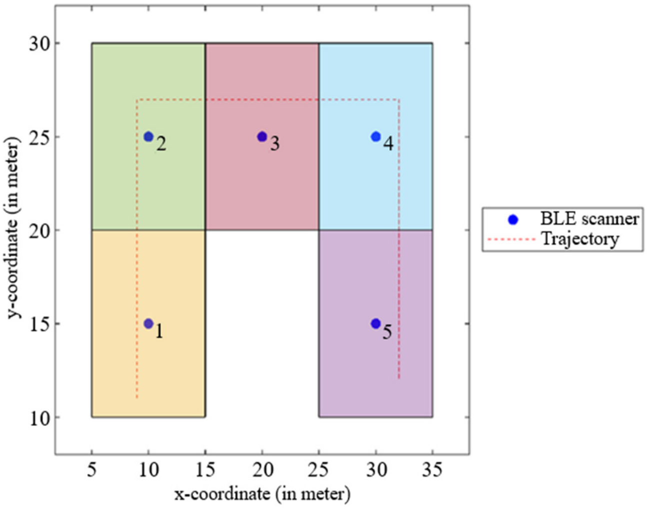


Fig. 4 Layout with BLE 2-D scanner coordinates

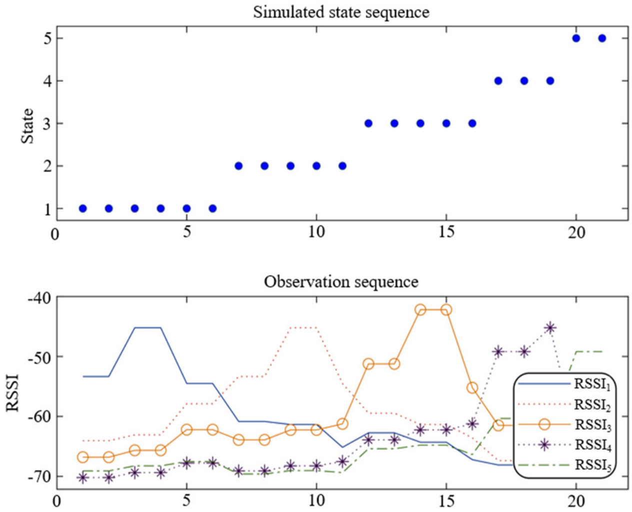


Fig. 5 Example of simulated data

4.3. Real-world experiment

In a real-world scenario, different stations were prepared to collect RSSI data from BLE devices using the components: WeMos R32 boards as BLE tags and scanners, Wi-Fi access points, and the Firebase database. The data collection was performed in a U-shaped area of 7 × 4 m with approximately 2 meters between each scanner as shown in Fig. 6.

(1) WeMos R32 boards are programmed as BLE scanners that connect with the WiFi access point and report ID/RSSI/timestamp to the Firebase database.

(2) WeMos R32 boards are programmed as BLE tags were moved from one working area to another according to a predetermined sequence.

(3) Each BLE tag is reset at the endpoint to renew the tag ID.

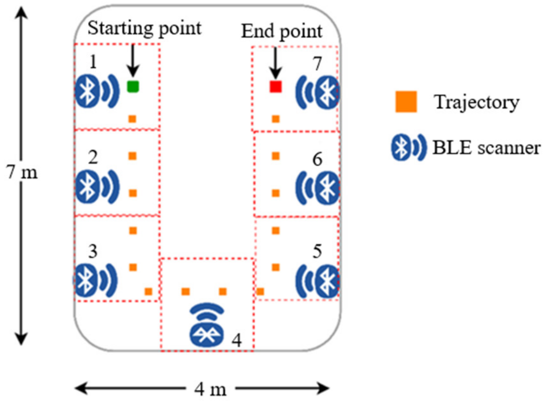


Fig. 6 BLE scanner allocation in a real-time experiment

The scanning period for all BLE scanners is set at 5 seconds. 25 sets of data were collected by moving the BLE tags through a pre-determined sequence from one working area to the next. As illustrated in Fig. 7, the results show that not all scanners can receive the RSSI signal. The occupation of the tag in each BLE scanner area is determined using the relationship between distance and RSSI in Fig. 2.

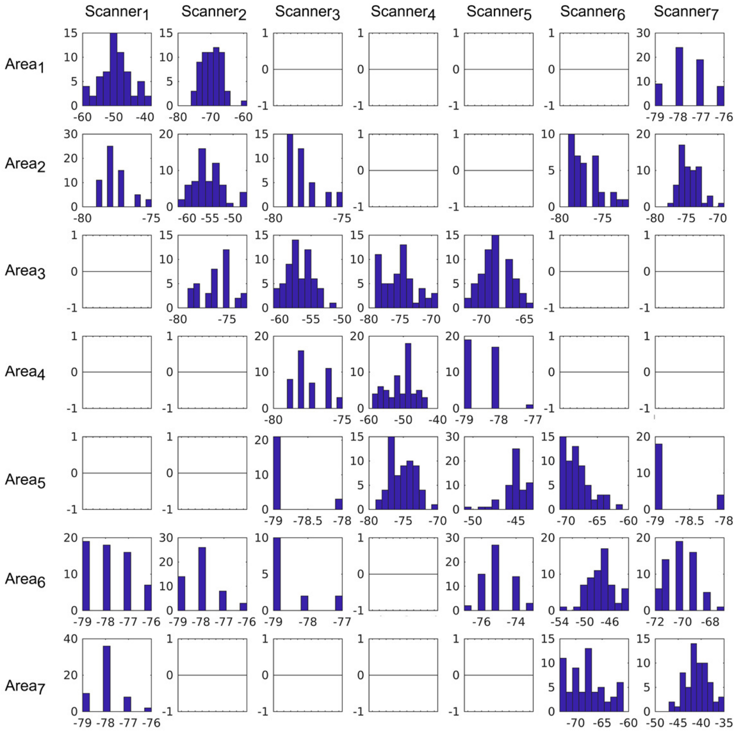


Fig. 7 RSSI measurements in real-world environments

5. Discussion

This section presents the results of three studies conducted to assess the effectiveness of the algorithm in different scenarios. The objective is to validate its capability in comprehending the characteristics of the production process through empirical evidence.

5.1. Analysis of numerical experiment

In this study, the Kullback-Leibler divergence (KLD) [13] is utilized as the distance metric to determine the similarity between the original p and the re-estimated $\hat{p}$.

$$D_{KL}\left(p_i\left(d\right)\middle\|\widehat{p_i}\left(d\right)\right)=\sum_{d=1}^{D}p_i\left(d\right)ln\frac{p_i\left(d\right)}{\widehat{p_i}\left(d\right)} \tag{16}$$

The average KLD value for the entire system approximately stands at 0.1843. This value is close to zero, indicating that the estimated $\widehat{p_i}(d)$ is nearly equivalent to the actual $p_i(d)$. To show the similarities, Fig. 8 displays a comparison between the predicted duration using HSMM and the actual duration with a range represented by minimum and maximum values. The average throughput time based on numerical data is 28.51 epochs, while the HSMM estimated average throughput time is 27.69 epochs.

Fig. 9 shows the trend of the average KLD values concerning the number of coupled sequences used in the training phase. It demonstrates that the proposed algorithm converges quickly with a reasonable amount of data. As the number of samples increased from 100 to 1000, the KLD value decreased from 1.7281 to 0.1843. Moreover, the evidence supports that Algorithm 1 is a reliable and accurate method for estimating the duration probability.

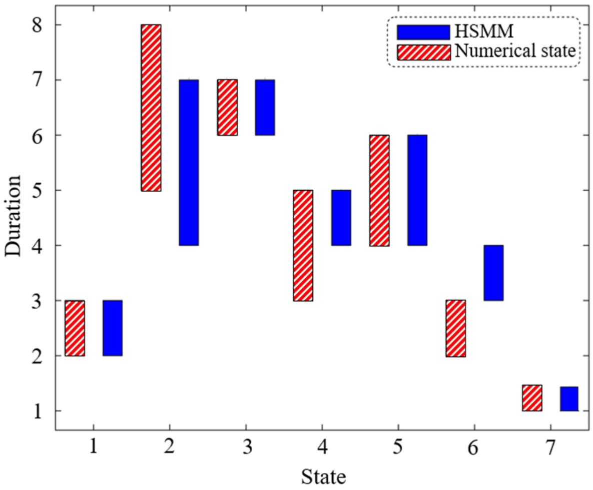


Fig. 8 Predicted vs actual duration comparison

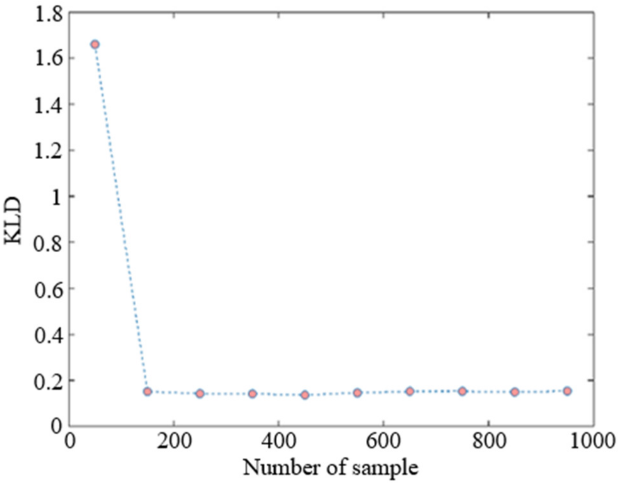


Fig. 9 Trend of average KLD value in the numerical experiment

5.2. Analysis of the simulation experiment

In the simulated experiment, 1000 sequences were generated, and then Algorithm 1 was employed to parameterize an HSMM model. The average throughput time for the simulated sequences was 20.53 epochs, while the estimated throughput time using the HSMM model was 19.89 epochs. This result is promising because it shows that the estimated throughput time is close to the simulated sequences. The study was continued by running simulations and estimations for another 500 state sequences as part of the forecast set.

The similarity between the forecasted and simulated sequences is evaluated with run-length encoding (RLE) and vector distance. RLE is a simple data compression technique that works well for sequences where the same value occurs in many consecutive manners. It encodes the sequence by replacing consecutive repeating values with a single value and its count. For example, the RLE input presents in “aaabbcccc” and the output is “3a2b4c”. The vector distance is used to measure the similarity of RLE repetitive counts as a vector. The vector distance is calculated by:

$$Vector\ distance=\sqrt{\left(\mathrm{X}-\mathrm{Y}\right)^2} \tag{17}$$

where X represents the simulated data, and Y is the forecasted data. For example, the RLE of the simulated data X becomes "3a2b4c" and the RLE of the forecasted data Y becomes "3a2b4c". In this case, the vector distance between the RLE encoded sequences can be calculated using the following formula.

$$\sqrt{(3-3)^2+(2-2)^2+(4-3)^2}=\sqrt{0+0+1^2}=1 \tag{18}$$

The lower the vector distance it becomes, the greater similarity it results in. A value of zero indicates that two sequences are identical.

The comparison of vector distances between one sample to 500 samples is depicted in Fig. 10. The average vector distance decreased from 3.8531 for one sample to 0.9740 for 500 samples. Thus, it indicates that the forecasted and simulated sequences had become more similar. This result shows that the estimated duration becomes more precise as the number of samples increases.

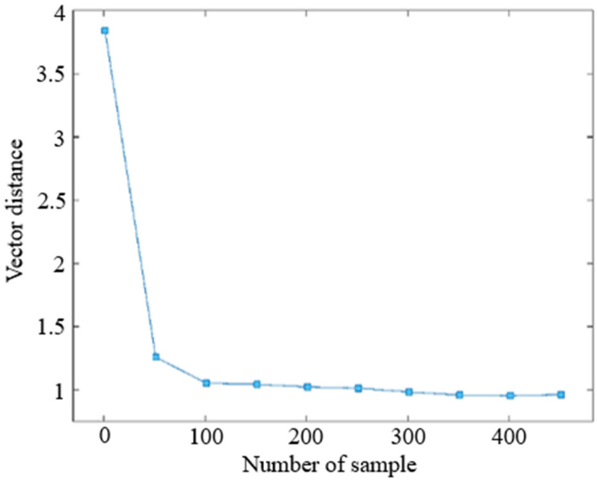


Fig. 10 Vector distance for simulated and forecasted sequences

5.3. Analysis of real-world experiment

The objective of the real experiment is to demonstrate the effectiveness of Algorithm 1 with realistic data. To achieve this, data were collected from 7 BLE scanners for 25 BLE tags. The dataset was filtered based on tag ID to retain the RSSI values associated with the scanner number. The filtered data was input into Algorithm 1, which is used to re-estimate parameters $\hat{p}$. Fig. 11 compares the state sequences generated by the trained HSMM model to the actual data collected during the experiments.

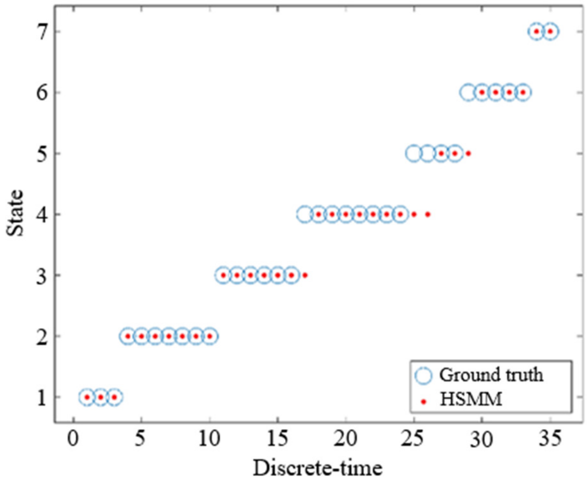


Fig. 11 Hidden state alignment example

Table 1 State durations (unit of 5 secs.)

Area (state)	Average cycle time (in epochs)	
	Real-world	HSMM
#1	3.28	2.88
#2	6.68	5.88
#3	5.84	6.16
#4	8.76	8.56
#5	3.36	1.52
#6	4.60	4.44
#7	1.48	1.04
Throughput time (in epochs)	33.99	30.48

Table 1 presents a comparison of the average state duration values between the collected data and those generated by the proposed HSMM model. The average throughput time for the realistic data is 33.99 epochs (169.95 seconds), while the average throughput time for the HSMM model is 30.48 epochs (152.4 seconds), with a difference of only 3.51 epochs. The difference highlights the accuracy of the HSMM model in capturing the dynamics of the system. The close match between the realistic data and the HSMM model suggests that the model is capable of accurately predicting throughput time and state sequences based on the RSSI data.

6. Conclusions

This study aims to overcome the limitations of previous research by uncovering temporal behavior and identifying potential correlations in the HSMM with RSSI data from the production line. To achieve this, the proposed method utilized two key steps. First, the method involved updating duration probabilities using forward and backward procedures. It allows probable determination of observed sequences in an HSMM, which is a crucial aspect of understanding temporal behavior. Second, learning procedures were employed to update the duration parameter, enabling the model to adapt and improve its predictions based on the collected data. The proximity between the actual and predicted values in experiments highlights the reasonable accuracy of the HSMM model. It indicates that the proposed method effectively predicted temporal behavior.

Future work could involve further refinement of the method to include the condition when there is no RSSI available. Additionally, it is important to note that the limitations of this work include the dependence on accurate and consistent RSSI data and the assumption of stationary behavior in the production process. Despite this limitation, this study has shown the potential of RSSI and timestamp data for predicting production metrics in a manufacturing setting.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] W. C. Tan and M. S. Sidhu, “Review of RFID and IoT Integration in Supply Chain Management,” Operations Research Perspectives, vol. 9, article no. 100229, January 2022.

[2] W. C. Chen, M. H. Nguyen, and P. H. Tai, “An Intelligent Manufacturing System for Injection Molding,” Proceedings of Engineering and Technology Innovation, vol. 8, pp. 09-14, April 2018.

[3] J. V. Tébar-Rubio, F. J. Ramírez, and M. J. Ruiz-Ortega, “Conducting Action Research to Improve Operational Efficiency in Manufacturing: The Case of a First-Tier Automotive Supplier,” Systemic Practice and Action Research, vol. 36, no. 3, pp. 427-459, June 2023.

[4] R. Joppen, S. Von Enzberg, J. Gundlach, A. Kühn, and R. Dumitrescu, “Key Performance Indicators in the Production of the Future,” Procedia CIRP, vol. 81, pp. 759-764, 2019.

[5] S. Gao, J. I. U. Rubrico, T. Higashi, T. Kobayashi, K. Taneda, and J. Ota, “Efficient Throughput Analysis of Production Lines Based on Modular Queues,” IEEE Access, vol. 7, pp. 95314-95326, July 2019.

[6] P. Bencak, D. Hercog, and T. Lerher, “Indoor Positioning System Based on Bluetooth Low Energy Technology and a Nature-Inspired Optimization Algorithm,” Electronics, vol. 11, no. 3, article no. 308, February 2022.

[7] S. Vorapojpisut and K. Agrawal, “Modeling of Manufacturing Processes Using Hidden Semi-Markov Model and RSSI Data,” 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing, November 2022.

[8] T. Kluge, C. Groba, and T. Springer, “Trilateration, Fingerprinting, and Centroid: Taking Indoor Positioning with Bluetooth LE to the Wild,” 21st International Symposium on “A World of Wireless, Mobile and Multimedia Networks”, August-September 2020.

[9] D. Han, H. Rho, and S. Lim, “HMM-Based Indoor Localization Using Smart Watches’ BLE Signals,” 6th International Conference on Future Internet of Things and Cloud, pp. 296-302, August 2018.

[10] M. Arslan, C. Cruz, and D. Ginhac, “Semantic Trajectory Insights for Worker Safety in Dynamic Environments,” Automation in Construction, vol. 106, article no. 102854, October 2019.

[11] S. Sun, Y. Li, W. S. T. Rowe, X. Wang, A. Kealy, and B. Moran, “Practical Evaluation of a Crowdsourcing Indoor Localization System Using Hidden Markov Models,” IEEE Sensors Journal, vol. 19, no. 20, pp. 9332-9340, October 2019.

[12] S. Sun, X. Wang, B. Moran, A. Al-Hourani, and W. S. T. Rowe, “Radio Source Localization Using Received Signal Strength in a Multipath Environment,” 22nd International Conference on Information Fusion, pp. 1-6, July 2019.

[13] S. Sun, X. Wang, B. Moran, and W. S. T. Rowe, “A Hidden Semi-Markov Model for Indoor Radio Source Localization Using Received Signal Strength,” Signal Processing, vol. 166, article no. 107230, January 2020.

[14] C. H. Lin, K. J. Wang, A. A. Tadesse, and B. H. Woldegiorgis, “Human-Robot Collaboration Empowered by Hidden Semi-Markov Model for Operator Behaviour Prediction in a Smart Assembly System,” Journal of Manufacturing Systems, vol. 62, pp. 317-333, January 2022.

[15] W. Yang and L. Chen, “Machine Condition Recognition via Hidden Semi-Markov Model,” Computers & Industrial Engineering, vol. 158, article no. 107430, August 2021.

[16] B. Mor, S. Garhwal, and A. Kumar, “A Systematic Review of Hidden Markov Models and Their Applications,” Archives of Computational Methods in Engineering, vol. 28, no. 3, pp. 1429-1448, May 2021.

[17] K. Li, C. Qiu, X. Zhou, M. Chen, Y. Lin, X. Jia, et al., “Modeling and Tagging of Time Sequence Signals in the Milling Process Based on an Improved Hidden Semi-Markov Model,” Expert Systems with Applications, vol. 205, article no. 117758, November 2022.