# INDOOR TRACKING IN FACTORY ENVIRONMENT

**BY**

**KARISHMA AGRAWAL**

**A THESIS OR DISSERTATION SUBMITTED IN**
**PARTIAL FULFILLMENT OF THE REQUIREMENTS**
**FOR THE DEGREE OF MASTER OF ENGINEERING**
**DEPARTMENT OF ELECTRICAL AND COMPUTER**
**THAMMASAT SCHOOL OF ENGINEERING**
**THAMMASAT UNIVERSITY**
**ACADEMIC YEAR 2023**

THAMMASAT UNIVERSITY
THAMMASAT SCHOOL OF ENGINEERING

THESIS

BY

KARISHMA AGRAWAL

ENTITLED

INDOOR TRACKING IN FACTORY ENVIRONMENT

was approved as partial fulfillment of the requirements for
the degree of Master of Engineering

on Approval date June 29, 2023

Chairman

Professor Dr. Sanya Mitaim

Member and Advisor

Asst. Prof. Dr. Supachai Vorapojpisut

Member

Assoc. Prof. Dr. Chowarit Mitsantisuk

Member

Asst. Prof. Dr. Dahmmaet Bunnjaweht

| | |
|---|---|
| Thesis Title | Indoor Tracking in Factory Environment |
| Author | Karishma Agrawal |
| Degree | Master Of Engineering |
| Major Field/Faculty/University | Electrical and Computer Engineering<br>Thammasat School Of Engineering<br>Thammasat University |
| Thesis Advisor | Asst. Prof. Dr. Supachai Vorapojpisut |
| Academic Year | 2023 |

# ABSTRACT

This study focuses on the application of an Indoor Positioning System (IPS) using Bluetooth Low Energy (BLE) network for estimating key manufacturing metrics, specifically cycle time. The manufacturing flow is represented as a Hidden Semi Markov Model (HSMM) problem, and a model is trained using RSSI data from the BLE network. An algorithm is proposed to determine the duration probability distribution, a critical parameter of the model. The study also addresses scenarios where no RSSI data is available, such as when products are stored away from BLE scanners. This problem is formulated as a Duration and Interval Hidden Markov Model (DI-HMM), and a combined architecture of a classification tree and HSMM is proposed. The algorithms are extended to handle the DI-HMM problem. RSSI observation sequences are generated using software, real-world experiments, and simulations. The estimated HSMM duration probability is evaluated using the Kullback-Leibler Divergence (KLD), resulting in a score of 0.0573. The estimated duration and interval of the DI-HMM are compared to the actual values using a vector distance score of 0.4717.

**Keywords:** Manufacturing process, Bluetooth Low Energy, Duration and Interval Hidden Markov Model, Classification Tree, Hidden Semi Markov Model

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Supachai Vorapojpisut, for his invaluable guidance and support throughout the completion of my master's thesis. His expertise, encouragement, and insightful feedback have greatly contributed to the success of this research. I would also like to thank Thammasat University for providing the necessary resources and environment for conducting this study. Thank you for the opportunity to pursue my academic goals and for the enriching experience gained during my time at the university.

Karishma Agrawal

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

The Global Positioning System (GPS) is widely used in outdoor navigation but fails inside an indoor environment due to inadequate GPS signals. The Indoor Positioning System (IPS) is a technology used to track down objects or people inside an indoor environment. IPS had become an essential tool to provide better services in several domains such as commercial, hospitals, airports, shopping malls, manufacturing areas, and supply chain.

A factory is a combination of various buildings to conduct several processes to produce final products to be stored and sent to customers. Generally, the factory area has two sections based on their operations: manufacturing (production) area and warehouse area. A production area is a sequence of works to manufacture final products, and a warehouse is an area to store these products until customers demand them. The production area consists of multiple working areas and storage areas. There are usually storage areas within production area is to temporarily keep intermediate parts among working areas.

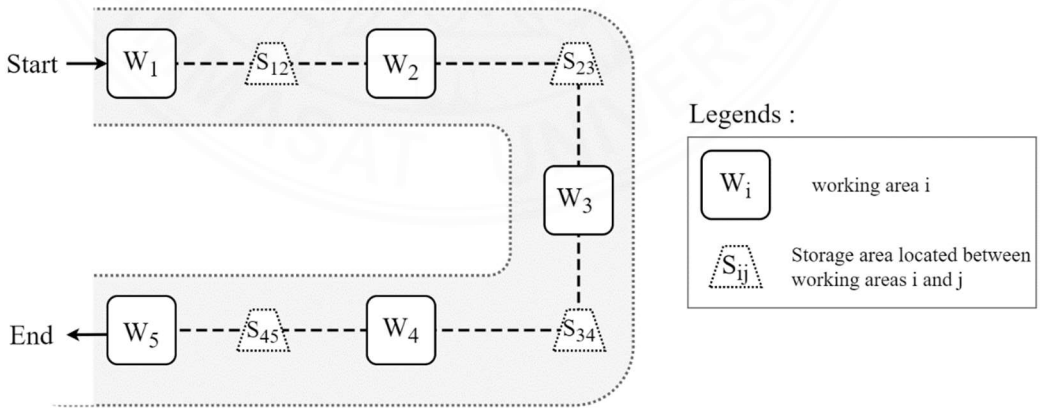



Fig. 1.1 Flow within of production area

In the production area, the sequence of works from initial to final working areas are usually pre-defined for a product. Production flow refers to how raw materials move in sequential order from one working area to another to get final products. Working areas

complete their specific individual task and move finished parts to the next working area. If the next working area occupies some work, then the parts will be shifted into a temporary storage area. This production flow can be understood by the tracking of changes from raw material to parts then products in production area.

Indoor Positioning Systems use sensing devices based on radio-communication technology to track an object. The most used technology is Radio-Frequency IDentification (RFID). RFID is a technology with two main components: reader and tag. Tag is attached to a tracking object and communicates to the reader when it is closed to tags thus defines the object's location. There are two types of RFID tags: passive RFID and active RFID tags. Passive RFID tags obtain power from the reader; therefore, the reader can receive a signal from a tag only when a tag is near the reader. The passive RFID reader tends to be bulky in size and covers less distance. As an example, RFID reader gates are bulky, and tags are only detected when the pass through the RFID reader gate. Hence, the implementation of passive RFID system requires modification in the pre-existing building that is not flexible for industries as shown in the below figure.

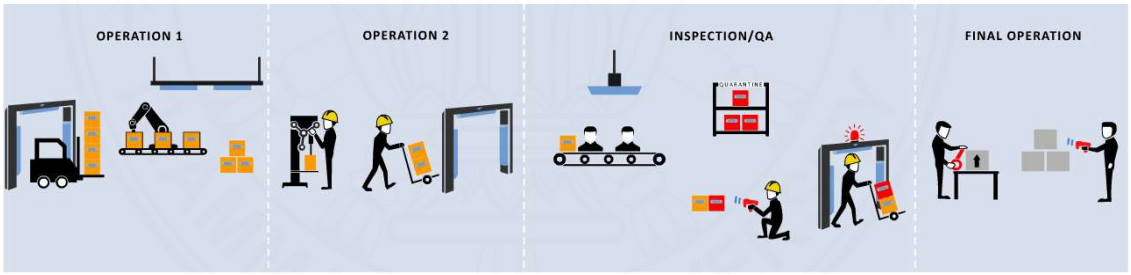



Fig. 1.2 Passive RFID asset [adopted from www.entigral.com]

Active RFID tags unite battery power to broadcast signals that lead to reducing the reader size and increasing the range of communication. Related wireless technologies that can be used with the concept of active RFID are Bluetooth Low Energy (BLE), WiFi, and LoRaWAN (Long Range Wide Area Network). Thus, this technology is easy to implement, and no reconstruction of buildings required. In this study, we use BLE-based indoor positioning because of its high (RSSI) accuracy over a small area and can use globally without license.

The indoor Positioning System does not use information about the physical arrangement and flow between working areas. We use the Hidden Semi Markov Model

(HSMM) to define physical arrangement and flow between working areas, where the object's location is considered a hidden state and the Received Signal Strength Indicator (RSSI) values are an observation value. The main advantage of using HSMM is that we can create a reasonable production flow based on historical data that can be easily modified in the future.

### 1.2 Objective

The main target of this thesis is to study Indoor Positioning System within factory environment using active RFID technologies. The objectives of this thesis are decomposed into two sub-objectives:

1) Mathematical model of production flow using RSSI data from BLE devices.
2) Estimate manufacturing KPIs such as cycle time and throughput using a mathematical model.

### 1.3 Scope

This study will apply math notions from the HSMM framework with RSSI data acquired from BLE devices with the assumptions of factory environment. The following tools will be used in the study:

1) Hardware:
    a) Wemos D1 ESP32 – The ESP32 can work as a BLE beacon (tag) and BLE scanner for BLE tracking. The BLE beacon broadcasts its existence so that the BLE scanner can scan. Whereas BLE scanner will send all scanned data to the server via Wi-Fi.
2) Software:
    a) Arduino IDE 1.0.4 – Arduino IDE is open-source software, which allows us to write code and upload to Arduino-compatible boards and other boards like ESP32 using third-party Arduino cores. Libraries used - ESP32 BLE Arduino, ESP32 WiFi, and Firebase-ESP32.
    b) MATLAB R2022a – MATLAB compute and simulate with Bluetooth toolbox.
    c) Firebase – Firebase is a type of NoSQL database program, which collects RSSI data from BLE devices in JSON-like documents.

## 1.4 Terminology

1) RFID
    a) **RFID reader** is a device that transmits and receives radio waves to communicate with RFID tags.
    b) **RFID tag** is a device to be attached to a tracking object. The tag broadcast signals and sends signals to RFID readers.
2) BLE
    a) **Scanner** is a device that receive data from beacon.
    b) **Beacon** is a device that only advertise information in form of radio signal.
    c) **Received Signal Strength Indicator (RSSI)** measure of how much power a scanner receives from a beacon.
3) Manufacturing
    a) **Working area** is the fixed location where a unique operation is performed.
    b) **Storage area** is used to temporarily store intermediate parts between working areas.
    c) **Production flow** is the movement of products in the manufacturing area.
4) Hidden Semi Markov Model (HSMM)
    a) **State** is a symbol representing working area (Scanner)
    b) **State duration** refers to the length of time spent in a particular state within the model (Cycle time and throughput).
    c) **Observation** is the term for data that can be seen (RSSI).

## 1.5 Literature Review

Academic problems arising in the Indoor Positioning System can be classified into localization and tracking problems. Localization describes the position of stationary object. Tracking problem is about the location of the moving object over time.

### 1.5.1 RFID (Radio-frequency Identification)

RFID is a radio-based technology used for wireless communication over a fixed distance. RFID is the oldest technology used for tracking and localization techniques. RFID can communicate and store information of multiple objects at the same time.

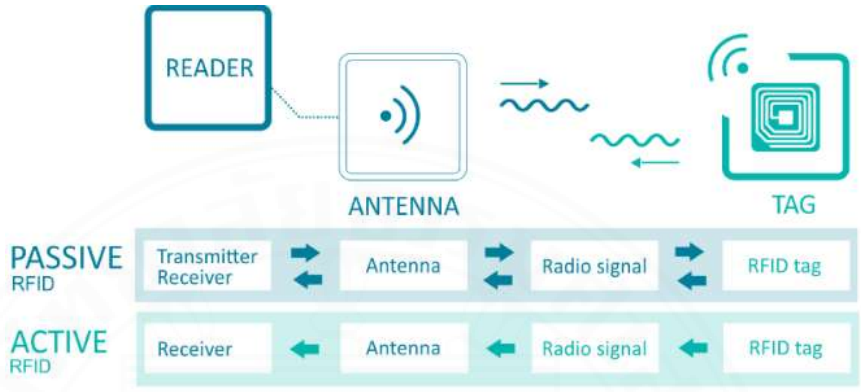



Fig. 1.3 Passive and active RFID systems [adopted from www.aucxis.com ]

Ail Monstaser and Osama Moselhi (2014) have discussed the indoor location tracking for construction sites using RFID for collecting essential data related to the locations of workers, materials, and equipment's to see the progress of construction. The methodology had applied RFID and localization methods using a two-step algorithm assisted with relational databases. Authors resolved the location of workers by moving mobile RFID reader to find materials and equipment's and nearby workers. The location and material identification uses two methods triangulation and proximity. In the construction site, passive RFID tags at areas known as reference points. A worker carries a mobile RFID reader, which is used to read RFID tags. The worker location is determined using a Received Signal Strength Indicator (RSSI) approach a database stores location values in 2-D coordinates. The location of workers and materials placed nearby workers can be located with respect to the mobile reader. Bayesian probability and K-Nearest Neighbor (BKNN) applied to resolve issues of are noisy signal and path loss affected the visibility of materials. The results show a mean error of 1.0 m and 1.9 m for user location identification and material tracking using the triangulation method, respectively. The results also show a mean error of 1.9 m and 2.6 m for location identification of the worker and for material tracking using the proximity method,

respectively. The proposed system identifies the areas of worker and material location with 100% accuracy.

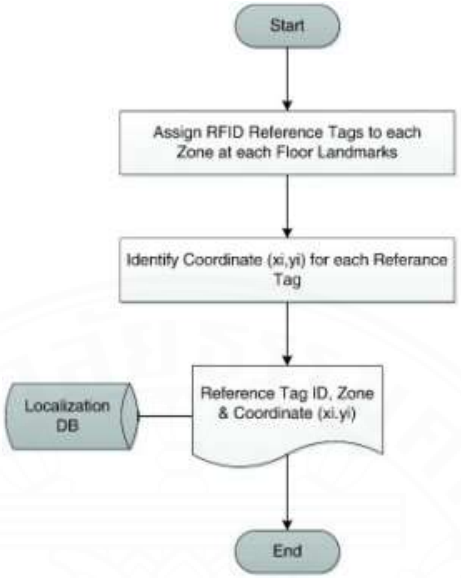



Fig. 1.4 The process of deploying RFID reference tags.

Xu et al. (2017) described the problem regarding low accuracy in indoor localization. The proposed system applied Bayesian probability and K-Nearest Neighbor (BKNN) algorithms with data from RFID. BKNN implemented three implemented algorithms are the K-Nearest Neighbor (KNN), the Gaussian filter, and the Bayesian estimation. The KNN algorithm selects three reference tags to know object location. Then the Gaussian filter and the Bayesian estimation algorithm are applied to remove the signal noise. As a result, the BKNN system has approximately a 15cm error and nearly a 38cm error in the Gaussian-based system.

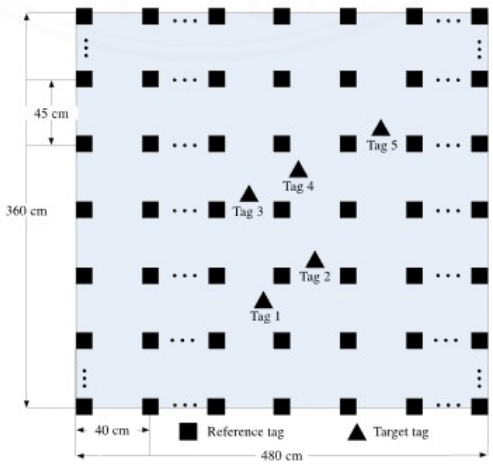



Fig. 1.5 BKNN experimental environment

Zhang et al. (2014) outlined the problem of indoor localization is not suitable for a large range. It proposed RTLS using active RFID for IoT (iLocate). The localization accuracy improved by the concept of virtual reference tags and tag-tag communication protocol. To remove the RFID RSSI noise, iLocate applied the frequency-hopping technique. To serve a wide-range RFID network used Zigbee. An object detection accuracy of the proposed iLocate system is up to 30 cm with ultralong distance transmission. The proposed system has large computation and complexity, which is not suitable for industries.

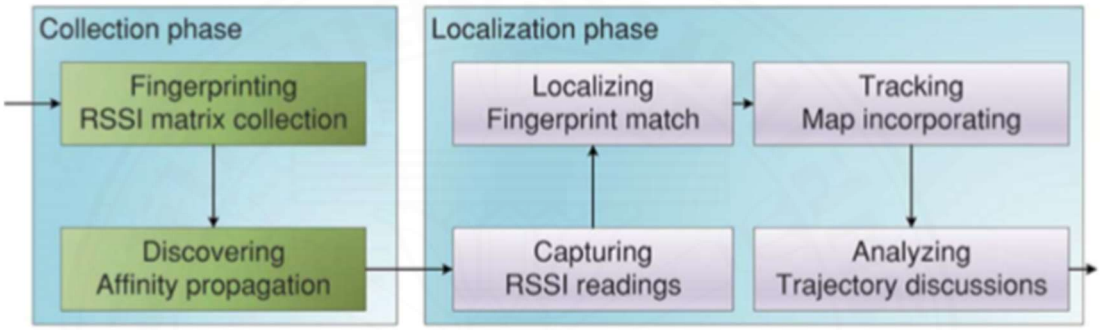



Fig 1.6 Architecture of the iLocate system

### 1.5.2 Bluetooth

A beacon is a small Bluetooth Low Energy (BLE) device that broadcasts a continuous radio frequency signal and its unique ID number (like RFID Tag). Nearby smart devices can receive beacon broadcast signals. Nowadays, shopping-malls and airports utilize beacon technology for offline advertisements.

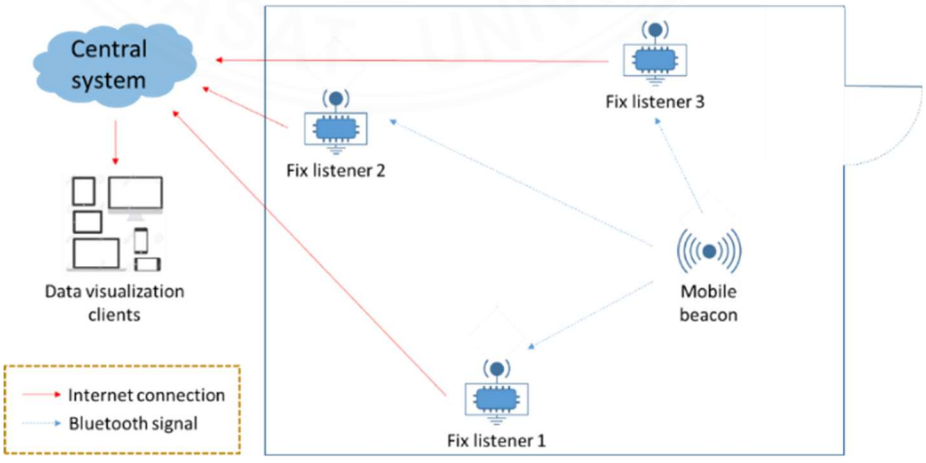



Fig 1.7 Beacon Asset Tracking [adopted from www.todaysoftmag.com]

Jun-Ho Huh and Kuyungryong Seo (2017) proposed indoor localization using Bluetooth beacons. They designed a system that tracks the activity of a user when it enters its vicinity. The applied location techniques include spatial partitioning, Received Signal Strength Indication (RSSI), trilateration, and Centre of Gravity (CoG). The proposed system is comprised of three main parts: location-based server, service-provision client, and user application. The location-based server controls the access of the client's smartphone then predicts its location and maintains the indoor map. Service-provision client provides indoor map information by the location-based server. The User Application an Android applications that performs socket communication with a server for sending RSSI values of beacons. The Android application displays the 2-D RSSI coordinates of location-targeted. As a result, the system provided an 74% accuracy with margin of error is 1 meter.

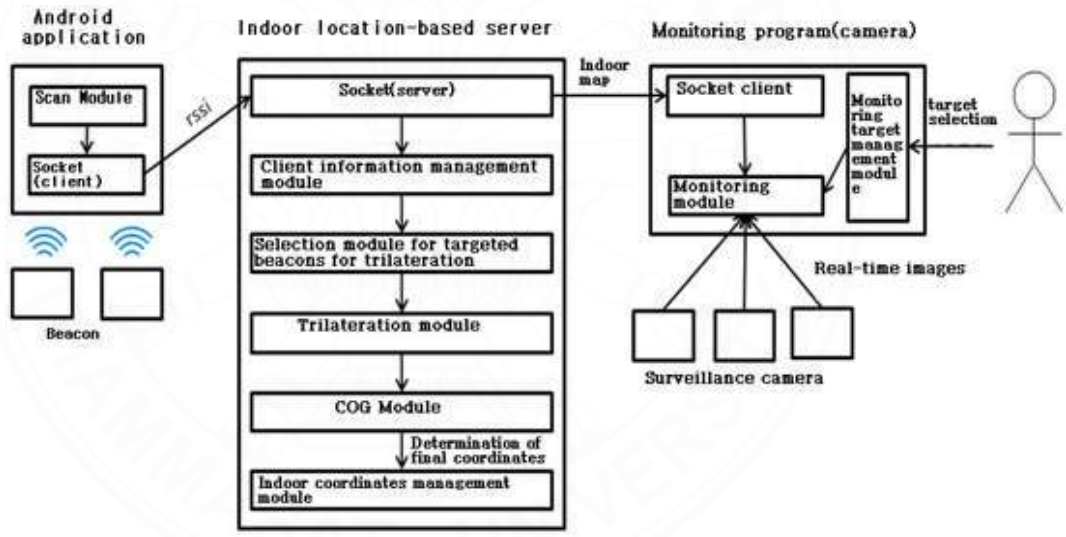



Fig. 1.8 Indoor location-based system

Sou et al. (2019) described the problem regarding line-of-sight in indoor location tracking. The proposed system implemented the Markov chain model in wireless fingerprinting that improves the visibility of a targeted object. Nine beacons are set on the top of a bookshelf in the library to test the proposed system. The location of a child is tracked using smart-phone Bluetooth services. As a result, the estimated position of the child is displayed on a mobile application with 34.29% ~ 69.92% accuracy and the mean distance error is 1.31 ~ 2.34 m.

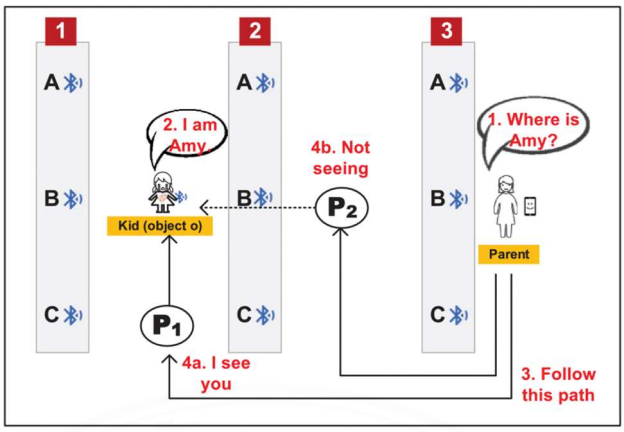



Fig. 1.9 Basic concept of kid tracking on the bases of visibility.

### 1.5.3 Hidden Markov Model (HMM)

A Hidden Markov Model is a double-layered finite-state process, with a hidden Markovian process that controls the selection of states based on an observation process. HMM parameters have the conditional probability distribution upon such observation (output parameter) (Rabiner, L. R. (1989), Nguyen, L. (2016)).

Young Ho Lee and Ivan Marsic (2018) highlighted the problem of motion detection of medical objects based on UHF RFID technology and proposed a solution using the HMM with RSSI collected with tags information. The RSSI values are analysed by Simplified Augmented HMM (SAHMM) of frame-based segments to detect a motion-transition with the exact time movement. SAHMM observe a change in position by comparing the present and previous data without using all history of data shown. Then supervised learning is implemented on the bases of emission probability. As a result, the passive UHF RFID system can provide information about the movement and location of a medical object.

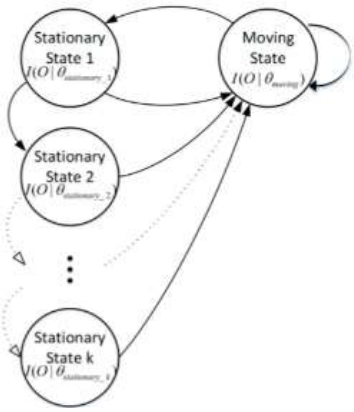



Fig 1.10 State of SHMM

SUN et al. (2019) proposed crowdsourcing based indoor mobile localization by combining Wi-Fi RSSI data with HMM framework. The HMM-based model extended to a Hidden Semi Markov Model (HSMM) to solve sojourn time. The author focused on the Viterbi algorithm to find the most likely sequence of hidden states. The first stage of the system used to analyse the state of the mobile user by applying the Ground Truth State Generation algorithm. Then RSSI observation sequence is sampled from the collected RSSI database. At each Monte-Carlo simulation, the error rate is computed for each algorithm (RSSI fingerprinting, HMM-Viterbi, and HSMM-Viterbi). Based on 10000 Monte-Carlo runs, the error rates are 10.42%, 7.82% and 15.11% for HMM-Viterbi, HSMM-Viterbi and RSSI fingerprinting, respectively. The result shows that the HMM solution is more reliable than the pure RSSI fingerprinting method.

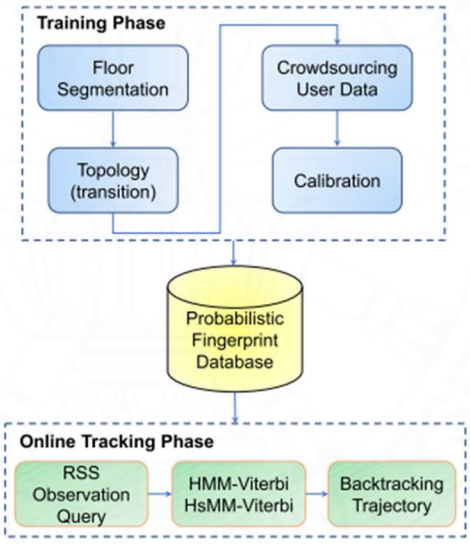



Fig 1.11 Framework of Crowdsourcing Localization System

SUN et al. (2020) highlight that HSMM are more flexible than HMM for modelling radio source dynamic properties and introduce observation missing mode in HSMM to deal with the missing RSSI measurement. The author focused on all three HSMM problems: forward-backwards, Viterbi, and leaning algorithm. The author evaluates the similarity between the estimated duration probability and the ground truth using the Kullback-Leibler divergence (KLD), concluding that HSMM requires more training data and demonstrating its impact by examining different data lengths.

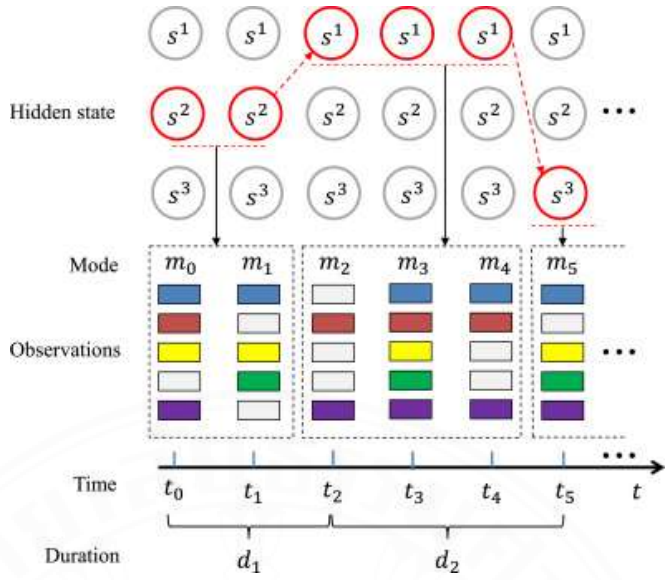



Fig 1.12 Example of using the HSMM to model RSSI measurements with varying missing entries.

Narimatsu et al. (2015) proposed the Duration and Interval Hidden Markov Model (DI-HMM), a new sequential data model that effectively represents the "state duration" and "state interval" of data events. This has significant implications for how time-series sequential data is represented in practice. The author first explains why support for "state duration" and "state interval" is essential for representing practical sequential data. Then, to explain "state duration" and "state interval," extend the HSMM by newly introducing state interval probability to each transition probability between two states. This present paper calls this new extended model as Duration and Interval Hidden Markov Model (DI-HMM).

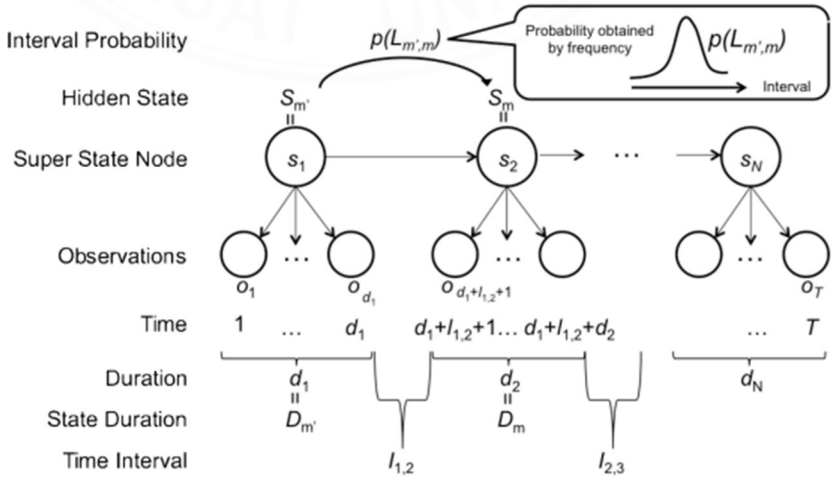



Fig 1.13 Concept of DI-HMM using interval probability.

The above Literature reviews on indoor tracking system gathered from the following:

1) IEEE (Institute of Electrical and Electronics Engineers)
2) Elsevier
3) MDPI sensors
4) Google Scholar

Table 1.1 Literature Review Summary

| **References** | **Survey focus** | **Method** | **Device** | **IPS Type** |
|---|---|---|---|---|
| Montaser et al. (2014) | Determine location of workers in construction site and materials placed nearby workers. | - RSSI<br>- Triangulation<br>- Proximity | RFID | Localization |
| Xu et al. (2017) | Improve localization by introducing Bayesian probability and K-Nearest Neighbor (BKNN) algorithms. | - K-Nearest (KNN)<br>- Gaussian filter<br>- Bayesian estimation | RFID | |
| SUN et al. (2020) | The main goal is to show HSMM is a better option for dealing with missing data issues because of its flexibility in modeling state sojourn time. Also, show that HSMM is better than HMM. | - RSSI<br>- HSMM and HMM<br>- KLD | HackRF One software | |

| References | Survey focus | Method | Device | IPS Type |
|---|---|---|---|---|
| Zhang et al. (2014) | The goal of the iLocate system is to locate objects accurately and quickly in large-scale networks. | - RSSI<br>- K-nearest neighbors (KNN) | RFID | Tracking |
| Huh et al. (2017) | The activity tracking of a user when user enters in a room. | - Spatial Partitioning,<br>- RSSI<br>- Trilateration, and Centre of Gravity | Bluetooth | |
| Sou et al. (2019) | Resolve line-of-sight in indoor location tracking. | - Markov chain<br>- fingerprinting | Bluetooth | |
| Lee et al. (2018) | Detection of medical objects. | - RSSI<br>- Hidden Markov model | RFID | |
| SUN et al. (2019) | Tracking location in crowdsourcing. | - Hidden Semi Markov model<br>- Fingerprinting, Bayesian | Wi-Fi | |

## 1.6 Contribution

The main contributions of this study include the development and evaluation of an IPS based on a BLE network for estimating manufacturing Key Performance Indicators (KPIs), such as cycle time. The study formulates the manufacturing flow as a HSMM problem and proposes algorithms for training and parameterizing the model using RSSI data. Additionally, the study addresses scenarios where RSSI data is not available by formulating the problem as a DI-HMM and proposing a combined architecture of a

classification tree and HSMM. Experimental validation is performed using simulated, real-world, and software generated RSSI observation sequences, demonstrating the accuracy and effectiveness of the proposed methods in estimating durations, intervals, and forecasting. Overall, this research contributes to the advancement of IPS technology in manufacturing industries and provides valuable insights for improving manufacturing processes and productivity.

The research contributions of this study have been recognized and accepted for publication in reputable journals and presented at a conference. The following papers have been accepted:

1) Conference Paper:
   Title: "Modeling of Manufacturing Processes using Hidden Semi Markov Model and RSSI Data"
   Conference: iSAI-NLP-AIoT 2022 Conference
   Date of Conference: November 5-7, 2022
   Date Added to IEEE Xplore: November 30, 2022
2) Journal Paper:
   Title: "Production flow modeling based on BLE-based RSSI data with non-detectable areas"
   Journal: Songklanakarin Journal of Science and Technology
   Acceptance Date: April 21, 2023
3) Journal Paper:
   Title: "A Hidden Semi-Markov Model for Predicting Production Cycle Time using Bluetooth Low Energy Data"
   Journal: Advances in Technology Innovation
   Acceptance Date: June 10, 2023

# CHAPTER 2
# RELATED THEORIES

## 2.1 Model of Manufacturing Process

Manufacturing systems are usually modelled by their temporal describes the arrival and departure time of the product at the system. The manufacturing system breaks into several working areas being arranged sequentially to produce final products.

### 2.1.1 Key Performance Indicators

Key Performance Indicators (KPI) is a method to measure the system performance concerning a time for a specific job. Manufacturing KPI computes tracking progress in the various working area with specific goals. Moreover, the sum of various working KPIs reports the performance of the production area. Below mentioned are the basic manufacturing KPIs used by manufacturers to track the process in the production area:

1) **Cycle-time (CT)** defined as the time-spend in each working area as well as total manufacturing time.
2) **Utilization** is the total of product generate as a ratio of total possible product.
3) **Work-in-process (WIP)** represents the partially completed product in each working area.
4) **Throughput** is the number of products that pass through each working area and production area.

KPIs that manufacturing systems tend to monitor are:

1) Average cycle-time at each working area
2) Average factory cycle-time
3) Throughput rate

### 2.1.2 Queuing Theory

Queueing theory (Ghimire et al. 2017) is a traditional way that describes the manufacturing system's behavior and performance. The basic definition of queueing system is that a client arrives, waits for service, and then exits after finishing the process. Kendall’s notation was adopted to describe various behaviors of the queuing

systems and represented as **A/S/n,** where **A** is arrival process, **S** is service process and **n** is number of servers. Queuing system comprises three main parts:

1) **Server** is a processing area of queue system.
2) **Buffer** is a waiting part in system, which is interconnected with server and used when sever is busy.
3) **Event** arrival $\boldsymbol{\lambda}$ and departure rate

To avoid complexity, assume a finite capacity of buffer, and inter-arrival time is negligible. The sum of arrival rate $\boldsymbol{\lambda}$ and service rate $\mu$ is response time. Queuing system used to define two main qualities $N$ and $T$, where $N$ is the average number of jobs in a system (can be at buffer or server) and $T$ is the average delay in a job.

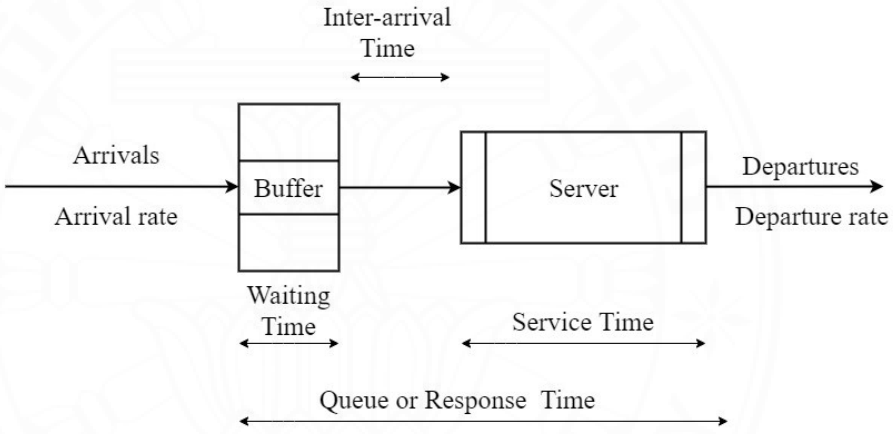



Fig 2.1 Basic Queueing system

The Little's law determines a throughput rate or the average number of jobs in a system. Little's law correlates the long-flowing jobs in a system $L_q$ is equal to the average arrival rate $\lambda$ multiply by the average amount of time spent $W$ in a system, i.e.,

$$L_q = \lambda W \tag{1}$$

Little's law is based on the black box because only input and output are visible and, the internal process of the model is invisible. Another parameter is utilization, defined as the ratio of average arrival rate $\lambda$ with respect to service rate $\mu$, i.e.,

$$\rho = \frac{\lambda}{\mu} \tag{2}$$

Using the utilization factor, average number of customers $N$ in a system can be determined, i.e.,

$$N = \frac{\rho}{1-\rho} \tag{3}$$

Cycle-time (CT) and work-in-process (WIP) of the manufacturing system can be determined by applying queuing theory. The Manufacturing system modelling (2010) divides cycle-time into two-part for analysing: average Factory Cycle-time and average Cycle-time at $i^{th}$ working area. Cycle-time consists of two components: processing time and waiting time. Thus, the average cycle time at $i^{th}$ working area is given as the sum of two components.

$$CT(i) = CT_q\ (i) + E[T_s\ (i)] \tag{4}$$

$CT_q(i)$ is the average time a job spends in the queue in front of the working area and $T_s\ (i)$ is the service time (or processing time) at $i^{th}$ working area. Assume manufacturing system is 'steady state' i.e., obey Little's law and use ergodic property then only queuing theory is applied. Notations used for explaining manufacturing system into mathematical form are:

1) $T_i^a$ à arrive time for the $i^{th}$ jobs.
2) $T_i^d$ à departure time for $i^{th}$ jobs
3) $A\ (\cdot)$ à number of jobs that arrived with time-interval $[0, t]$
4) $D\ (\cdot)$ à number of jobs departed with time-interval $[0, t]$
5) $N(t)\ =\ A(t) - D(t)$ à number of jobs in the system at time $t$
6) $N_{ab}$ à number of jobs with time-interval $[a, b]$
7) $\lambda$à average arrival rate $\left(\frac{Nab}{b-a}\right)$
8) $\mu$ à average service rate

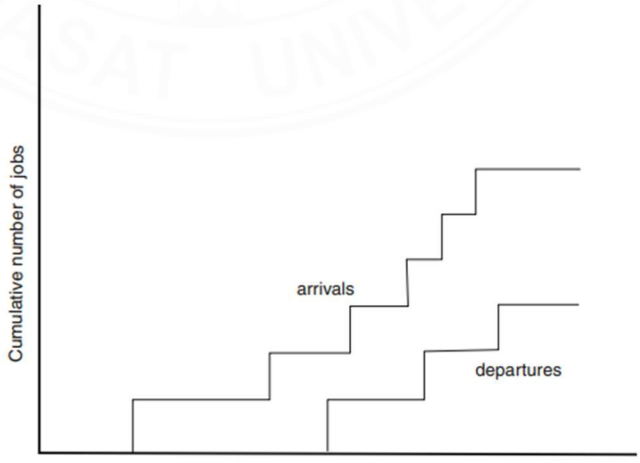



Fig 2.2 Arrival $A(\cdot)$ and departure $D(\cdot)$ functions realization

The mean waiting time CT during jobs considering for the period of $(a, b)$ period, given as:

$$\mathrm{CT(a,b)} = \frac{1}{N_{ab}} \sum_{i=1}^{N_{ab}} (T_i^d - T_i^a) \tag{5}$$

In Figure 2.2 assume $AB$ is an area between curve $A(t)$ and $D(t)$, where $t$ is between $(a, b)$ interval. To estimated $AB$ area standard integration approach is used.

$$AB = \int_a^b (A(t) - D(t))\, dt \tag{6}$$

An average work-in-process in the system with a time interval $(a, b)$ is defined as:

$$WIP(a,b) = \frac{1}{b-a} \int_a^b (A(t) - D(t))\, dt$$

$$WIP(a,b) = \frac{1}{b-a}\, AB \tag{7}$$

Cycle-time in the system with time-interval $(a, b)$ can be correlated with the work-in-progress.

$$CT(a,b) = \frac{1}{N_{ab}}\, AB \tag{8}$$

Thus, the following relationship is found.

$$WIP(a,b) = \frac{N_{ab}}{b-a}\, CT(a,b) \tag{9}$$

Since $\frac{N_{ab}}{b-a} = \lambda$ then,

$$WIP(a,b) = \lambda\, CT(a,b) \tag{10}$$

The above equation follows Little's law. Assume average arrival rate and average service rate is in exponential form and the probability of steady state system with n number of jobs.

$$\Pr\{N = n\} = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^n \quad \text{for n} = 0,1,\cdots,\infty, \tag{11}$$

In the above equation, *N* denotes the jobs number in the system.

Assume $E$ denote Erlang formula in below equation. Average work-in-processes (WIPs) of the system is expressed in a discrete probability distribution as:

$$WIP_s = E[N] = \sum_{n=0}^{\infty} np_n \tag{12}$$

$$WIP_s = \frac{\lambda/\mu}{1-\lambda/\mu} \quad , given\ that\ \lambda < \mu \tag{13}$$

Note that queuing model is implemented only when $\lambda < \mu$ is satisfied, i.e., the system is at equilibrium state. The expected time in the system or cycle time ($CT_s$) computed using Little's law, $CT_s = \frac{1}{\mu - \lambda}$.

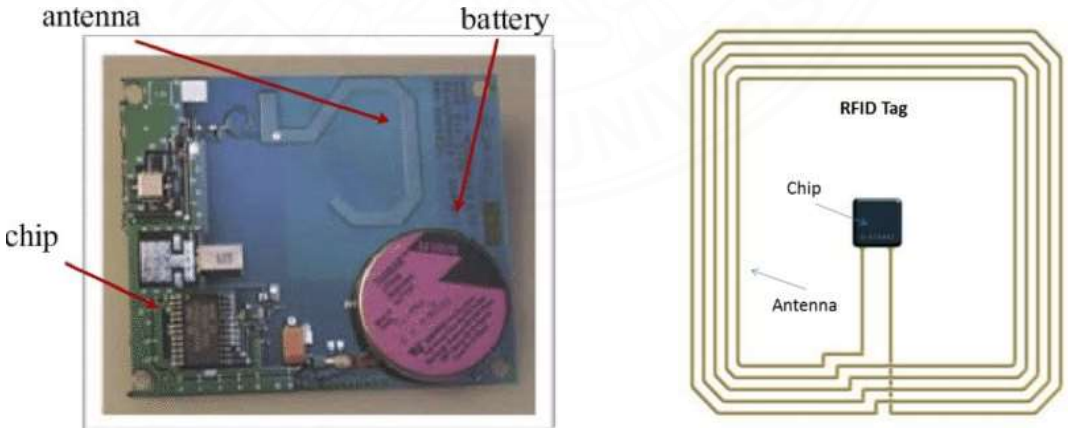
The queueing theory has some disadvantages to implement due to the following reasons.

1) Queueing system provides no visibility to any stage other than input and output.
2) In equation (11) system is in a steady state, which is valid only for long-term analysis. Queuing systems rarely reach to steady state. Therefore, implementation of queuing system in real-world is impossible.

## 2.2 Indoor localization

### 2.2.1 RFID (Radio-frequency Identification)

Radio-frequency Identification (RFID) technology works on tracking objects using radio waves. RFID has two main components: RFID tags and RFID reader. RFID tags are utilizing radiofrequency. These tags are manufactured using an antenna (receiving and transmitting) and microchip (store unique id). These tags are of two types: active and passive tags. Active tags have a power supply (battery) to drive the broadcast of signals, whereas passive tags are battery-less and have short-range than active tags.




(a) Active tags (b) Passive tags

Fig. 2.3 Types of RFID Tags

The RFID reader generates radio waves at a fixed frequency, which transmits and receives back from RFID tags. RFID tags are attached to objects for auto-identification.

RFID reader identifies the location when RFID tags are under a range of radio-wave. Hence, the RFID reader communicates with RFID tags without physical contact.

In our project, we use Beacons that is a type of active RFID that uses BLE. BLE (or active RFID) systems operate by a tag “beaconing,” or sending out transmission, to a reader, and then transmitting that location to the cloud. Beacons are a type of active RFID that use BLE.

### 2.2.2 Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) (Surian et al. 2019) is low power and short-range wireless technology that can be utilized for indoor localization. Received Signal Strength Indicator (RSSI) is the fundamental value in wireless localization system, which calculates a distance between the BLE scanner and BLE tag by inverse-square law and propagation model, hence estimate the location of the tracking object.

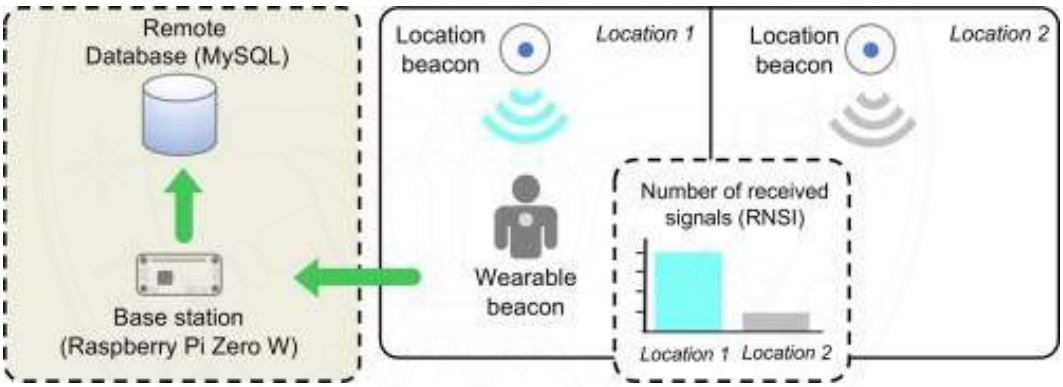



Fig 2.4 BLE Localization [Surian et al. (2019)]

In general, RSSI has three main issues: Signal Strength Attenuation, Signal Interference, and Multipath Propagation. Mostly, these issues resolve via fingerprinting and trilateration method. In the fingerprint method, gradually locate BLE tag near multiple fixed BLE scanners to collected RSSI values for creating a database (offline phase), then compare with a real-time (online) RSSI value and estimate the location. Trilateration is a method used to determine the location of an object using RSSI distance relationship. Localization of the BLE tag using trilateration technique assumes three or more BLE scanners as a specific fixed location-point. Each BLE scanner receives different RSSI values from the same BLE tag and then converted each RSSI value into

the distance using the path-loss model. Hence by comparing a distance value of individual BLE scanner location is confirmed.

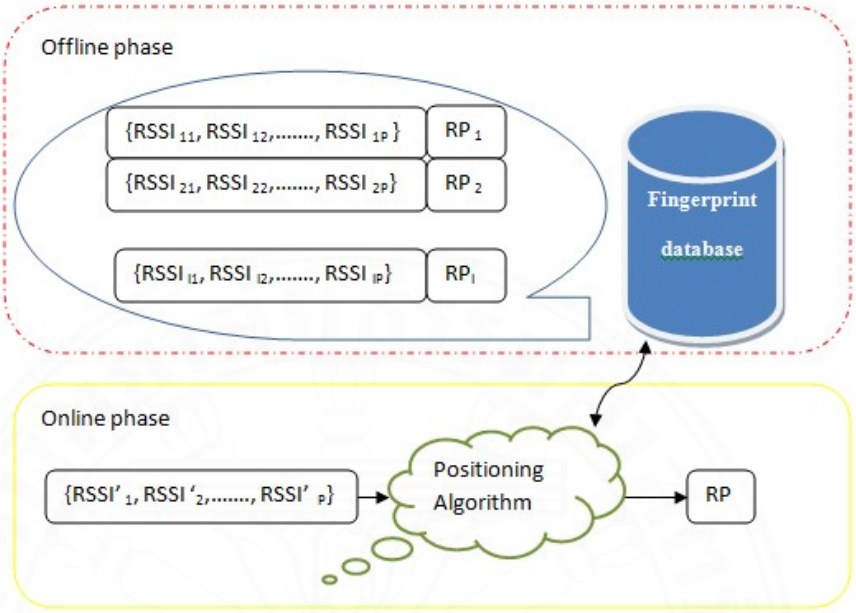



Fig 2.5 Fingerprint positioning scheme [Yasmine et al. (2016)]

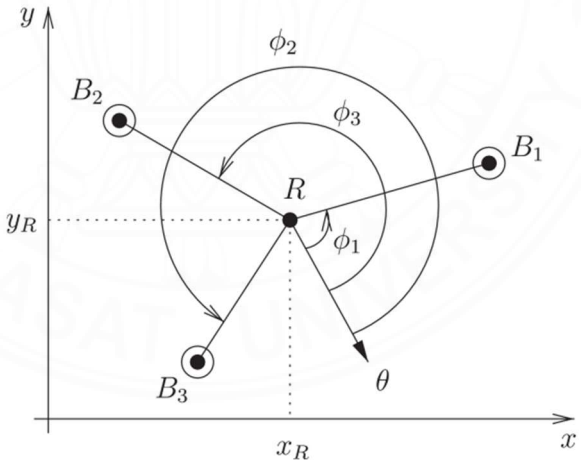



Fig 2.6 Triangulation setup in the 2-D plane [V. Pierlot et al. (2019)]

The main advantage of using the BLE is that it is a freely available (or license-free) wireless technology. The operating frequency range of BLE is 2.402 to 2.483 GHz ISM band. Gaussian Frequency Shift Keying (GFSK) is a modulation technique used by BLE which is easy to actualize. The usage of power consumption of BLE reduces due to the modulation index range 0.45 to 0.55.

### 2.2.3 Received Signal Strength Indicator

Received Signal Strength Indicator (RSSI) estimates the power level that a radio-frequency scanning device received from a broadcasting device considering a distance. RSSI values directly depend on the distance between a signal transmitted and received, and usually varies over time, called non-stationary signals (Jung et al. 2013). When RSSI strikes with any object reduces the original RSSI value, and distance calculation has an error. This error compensates through the path-loss model. RSSI is measured as dBm (decibel-milliwatts), which is a unit that indicates a signal power in decibels (dB) with reference to one milliwatt (mW). Figure. 2.3 shows RSSI variation with respect to the distance of Bluetooth device. This demonstrates characteristic of Bluetooth RSSI and distance in real environment.

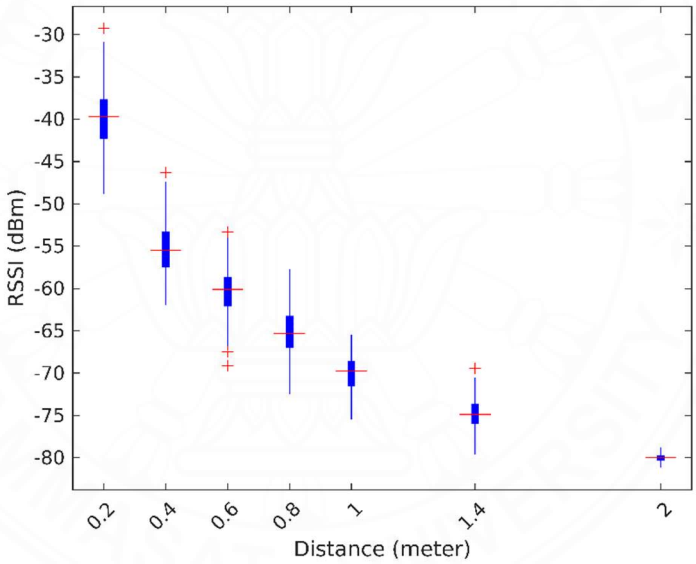



Fig 2.7 RSSI vs distance

Path loss models estimate the decrease in radio signal power as it propagates away from the transmitter. Mostly two types of path loss models are used in RSSI ranging: free space propagation and shadowing model. The path loss model of free space propagation calculates power signal loss along with a single line-of-sight propagation path only.

$$Loss = 32.44 + 10nlog(d) + 10nlog(f) \quad (14)$$

Where $Loss$ is path loss of signal power, $d$ is signal transmission distance (in meter), $f$ is wireless signal frequency and $n$ is path attenuation factor in real-world. When it

comes to real-world scenarios such as industrial production areas, it needs to assume shadowing caused by various barriers, thus requires the shadowing model. The shadowing model (logarithmic) is a stochastic path loss model where power levels obey a lognormal distribution.

$$P_L\,(d) = P_L\,(d_0\,) + 10nlog(d/d_0) + X_\sigma \tag{15}$$

Where: - $P_L(d)$ is absolute power value and describe receiving signal pathloss with distance $d$ ($d$ in meter), $P_L(d_0\,)$ is path loss of receiving signal with reference distance $d_0$ , $n$ is path loss index in particular conditions also speed of pathloss, and $X_\sigma$ is normal distributed random value used to model shadowing effects. General formula for calculating RSSI value of receiving node is:

$$RSSI = P_t - P_L(d) \tag{16}$$

Where, $P_L(d)$ is path loss with distance $d$, $P_t$ is signal transmission power, and $RSSI$ Received Signal Strength Indicator (in dBm). Assume $A$ is received signal strength from reference nodes with reference distance $d_0$.

$$A = P_t - P_L(d) \tag{17}$$

The path loss model measured at the real distance $d$ (m):

$$P(d) = P(d_0) - 10_n log\left(\frac{d}{d_0}\right) - X_\sigma \tag{18}$$

Where, $P(d)$ is received signal strength with real distance is $d$ (m) and $d_0$ = 1 meter (reference distance)

$$RSSI = A - 10nlog\left(\frac{d}{d_0}\right) - X_\sigma \tag{19}$$

$\overline{RSSI}$ is defined as $RSSI$ value multiple times of measurement.

$$\overline{RSSI} = A - 10nlog(d) \tag{20}$$

$d$ is defined as the undetermined distance.

$$d = 10^{(A-RSSI)/10n} \tag{21}$$

The above equation shows the relationship between distance and RSSI i.e., $RSSI \propto 1/distance$.

## 2.3 Hidden Markov Model (HMM)

### 2.3.1 Hidden Markov Model (HMM)

Hidden Markov Model (HMM) is a double-layer stochastic process with some specific hidden state based on the Markov process that relies upon observation states.

Utilizing Markov's property, the next state probability is predicted by considering the present state of the model. Hidden Markov Model consists of 5-elements:

1) State ($S$) is a stationary subprocess of the hidden non-stationary process.

$$S = \{S_1, S_2, S_3, \dots \dots, S_N\} \tag{22}$$

2) Initial state probability ($\pi$) represents the starting of a stochastic process.

$$\pi = \{\pi_1, \pi_2, \pi_3, \dots \dots, \pi_N\}$$
$$\pi = \sum_{S_i \epsilon S} \pi_i = 1 \tag{23}$$

3) Transition matrix ($A$) is a probability that varies from state to state.

$$A = \{a_{ij}\} = P(S_j|S_i) \tag{24}$$

where $a_{ij}$ refer to the probability that transition from present state $S_i$ to next state $S_j$ with time.

4) Observation state ($O$) is a secondary stochastic process that correlates with hidden states.

$$O = \{O_1, O_2, O_3, \dots \dots, O_M\} \tag{25}$$

5) Emission matrix ($B$) represents the probabilities of one observation state to another.

$$B = \{b_{ik}\} = b_i(O_k) = P(O_k|S_i) \tag{26}$$

The above equation represents the probability distribution for observation state ($O_k$) concerning each state ($S_i$) where, $i \neq k$, $1 \leq k \leq M$ and $1 \leq i \leq N$. Assume $k$ be any observation state for any state $i$ concerning any time. A sum of all observation probabilities of any state should be equal to 1.

$$\forall\, S_i, \sum_{O_k} b_i(k) = 1 \tag{27}$$

We consider the HMM structure known as the left-right model, as depicted in Figure 2.8, in which state transition starts at the left and proceeds to the right. For the left-right HMM, the initial probability distribution is 1 for the initial state and 0 for the others. The left-to-right HMM is a suitable model for manufacturing processes because products move in manufacturing flow in a specific order, and the cycle does not repeat.

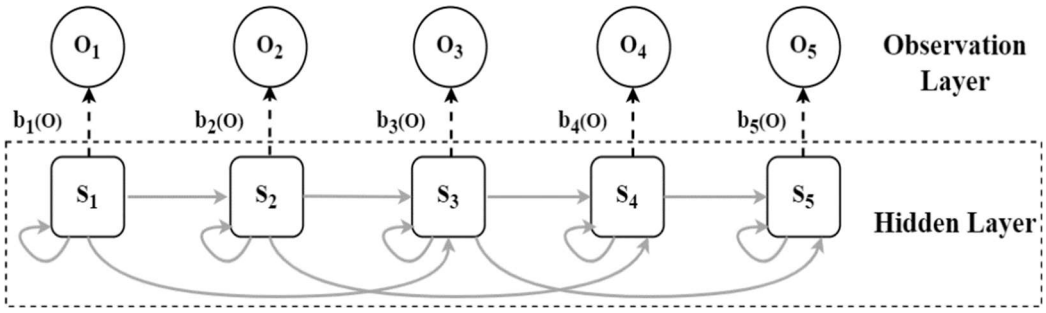



Fig 2.8 Left-right Hidden Markov Model

Formulation of HMM requires the following three conditions to be satisfied.

1) **The Markov assumption**: The transition probabilities of HMM define by the Markov assumption, which bases on the first-order Markov chain property. Hence, the first-order Markov chain property says that the HMM's next state is depends on the just previous state.
2) **Independent assumption**: emission probability of current observation depends only on the current state, consequently independent from other states and another-observation.
3) **The stationarity assumption**: state transition probabilities do not consider an actual time at which the transformation of state performs.

There are three problems of Hidden Markov Model (L. Nguyen 2016) are:

1) **Evaluation or Explanation Problem –** The evaluation problem determines the way to compute the probability $P(O|\Delta)$ based on observation sequence $O$ and HMM model Δ. A probability $P(O|\Delta)$ can determine through a forward-backward procedure. The forward-backward has two sub-divided variables: forward variable and backward variable.
    a) **Forward variable –**

$$\alpha_t(i) = (O_1, O_2, \dots, O_t, x_t = S_i \,|\Delta) \tag{28}$$

Where joint probability $\alpha_t(i)$ is forward variable of partial observation sequence ($O_1$, $O_2$, …, $O_t$) = O and state $x_t = S_i$ where $1 \le t \le T$, assume $n$ possible states of $x_T$.

**Step1 Initialization:** Initializing for all $1 \le i \le n$.

$$\alpha_i(i) = b_i(O_1)\pi_i \tag{29}$$

**Step 2 Recurrence:** Calculating all $\alpha_t$+1(j) for all $1 \le j \le n$ and $1 \le t \le T - 1$.

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{n} \alpha_t(i) a_{ij}\right) b_j(O_{t+1}) \quad (30)$$

Where $b_j(O_{t+1})$is the probability of observation $(O_{t+1})$ when the state stochastic process is in state $S_j$.

**Step 3 Evaluation:** Probability $P(O|\Delta)$ based on forward variable.

$$P(O|\Delta) = P(O_1, O_2, \dots, O_t)$$

$$P(O|\Delta) = \sum_{i=1}^{n} P(O_1, O_2, \dots, O_t, x_t = S_i|\Delta)$$

$$P(O|\Delta) = \sum_{i=1}^{n} \alpha_T(i) \quad (31)$$

b) **Backward variable –**

$$\beta_t(i) = (O_{t+1}, O_{t+2}, \dots, O_T | x_t = S_i, \Delta) \quad (32)$$

Where conditional probability $\beta_t(i)$ is forward variable of partial observation sequence ($O_{t+1}$, $O_{t+2}$, …, $O_T$) = O and state $x_t = S_i$ where $1 \leq t \leq T$.

**Step 1 Initialization:** Initializing for all $1 \leq i \leq n$.

$$\beta_T(i) = 1 \quad (33)$$

**Step 2 Recurrence:** Calculating all $\beta_t(i)$ for all $1 \leq i \leq n$ and $t = T-1, t = T-2, \dots, t = 1$.

$$\beta_t(j) = \left(\sum_{j=1}^{n} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)\right) \quad (34)$$

**Step 3 Evaluation:** Probability $P(O|\Delta)$ based on backward variable.

$$P(O|\Delta) = \sum_{i=1}^{n} \pi_i \, b_j(O_1) \beta_1(i) \quad (35)$$

2) **Uncovering Problem –** The uncovering problem determines the hidden state sequence by observation sequence. In other words, the simple criterion is the conditional probability of sequence X concerning sequence $O$ and model $\Delta$, denote $P(X|O, \Delta)$

$$X = argmax_X \, P(X|O, \Delta) \quad (36)$$

Assume $\gamma_t(i)$ be a joint probability (individually optimal criterion) that the arbitrary process in state $S_i$ at time point $t$ with observation sequence O = {$O_1$, $O_2$, …, $O_T$}. By reference equations (30) and (34), the forward variable $\alpha_t$ and backward variable $\beta_t$ can be calculated.

$$\gamma_t(i) = P(O_1, O_2, \dots, O_t, x_t = S_i|\Delta) = \alpha_t(i) \beta_t(i) \quad (37)$$

The probability $(O_1, O_2, \ldots, O_t \,|\Delta)$ is unrelated to state sequence X. The below formula defines how to determine the optimal state $x_t$ of X at time point t.

$$x_t = \text{argmax}_i \gamma_t(i) = \text{argmax}_i \alpha_t(i)\beta_t(i) \quad (38)$$

In the above formula, $i$ denote as state $S_i \in S$. The ideal state $x_t$ of $X$ for time $t$ is equal to the estimated product of $\alpha_t(i)\beta_t(i)$ for an overall value of $S_i$. Finally, state sequence $X = \{x_1, x_2 \ldots, x_T\}$ is determined when its partial states $x_t(S)$ where $1 \leq t \leq T$.

3) **Learning Problem –** The learning problem defines how to adjust a parameter of the model so that initial state distribution π, transition probability matrix A, and observation probability matrix B improve the quality and determine the maximum likelihood of the HMM Δ.

$$\theta = (A, B, \pi) = \text{argmax}_{A,B,\pi} P(O|\Delta) \quad (39)$$

The Expectation-Maximization (EM) or Baum-welch algorithm can resolve the HMM learning problem. Generally, the EM algorithm obtains a maximum-likelihood estimation of observed values with the help of the Gaussian distribution method from raw observed values. EM algorithm sub-divided into two parts: - Expectation step (E-step) and Maximization step (M-step). The main aim of EM algorithm is to estimate $\theta$ by finding out which $\hat{\theta}$ maximizes the likelihood function $L(\theta) = P(O|\theta)$.

$$\hat{\theta} = argmax_\theta L(\theta) = argmax_\theta P(O|\theta) \quad (40)$$

$$\hat{\theta} = argmax_\theta \sum_x P(X|O,\theta) \ln\big(P(O,X|\theta)\big) \quad (41)$$

Where: - $\hat{\theta}$ – parameter estimate, and "ln"- natural logarithm function.

Let us assume $\Delta = (a_{ij}, b_j(k), \pi_i)$ be the current parameter and $\hat{\Delta} = (\hat{a}_{ij}, \widehat{b_j}(k), \hat{\pi}_i)$ be the next parameter.

**E-step:** To begin E-step, we need two predefined forward variable $(\alpha_t)$ and a backward variable $(\beta_t)$ from the forward-backward algorithm. Assume we need to calculate $\gamma_t$ posteriori probability and $\xi_t(i,j)$ joint probability that the stochastic process receives state $S_i$ at time point $t-1$ and state $S_j$at the time point $t$ of given observation sequence $O$.

$$\xi_t(i,j) = P\big(O, x_{t-1} = S_i, x_t = S_j \,\big|\Delta\big)$$

$$\xi_t\,(\mathrm{i},\mathrm{j}) = \alpha_{t-1}(\mathrm{i})\;\; a_{ij}\;\; \mathrm{b_j}\,(\mathrm{O_t})\;\; \beta_t(\mathrm{j}) \tag{42}$$

When $t \geq 2$

$$\gamma_t\,(\mathrm{j}) = \mathrm{P}\big(\mathrm{O}, \mathrm{x}_t = \mathrm{S}_j \,\big|\Delta\big) = \alpha_{\,t}(\mathrm{j})\,\beta_t(\mathrm{j}) \tag{43}$$

Where forward variables $\alpha_{\,t}$ and backward variables $\beta_t$ are can be calculated by the previous equation (31) and (35), respectively.

**M-step:** Estimating next parameter $\hat{\Delta} = (\hat{a}_{ij}, \widehat{b_j}(k), \hat{\pi}_i)$ on the joint probabilities $(i, j)$ and $\gamma_t\,(\mathrm{j})$ determined at E-step.

$$\hat{a}_{ij} = \frac{\sum_{t=2}^{T} \xi_t\,(\mathrm{i,j})}{\sum_{t=2}^{T}\sum_{l=1}^{n} \xi_t\,(\mathrm{i,l})} \tag{44}$$

$$\widehat{b_j}(k) = \frac{\sum^{T}{}_{t=1}{}^{\;\;\gamma_t(\mathrm{j})}_{o_t=\varphi_k}}{\sum_{t=1}^{T} \gamma_t(\mathrm{j})} \tag{45}$$

$$\hat{\pi}_i = \frac{\gamma_1(\mathrm{j})}{\sum_{t=1}^{n} \gamma_1(\mathrm{j})} \tag{46}$$

The estimate $\hat{\Delta}$ becomes the current parameter for next iteration.

### 2.3.2 Hidden Semi Markov model (HSMM)

The HMM explained above has some limitations for scenarios when the data of a system is dependent upon time. To resolve this problem, the Hidden semi-Markov model (Sun et al. (2019)) introduces the variable duration $D = \{d_1, d_2, \ldots, d_N\}$ that relies on each state, also controls transition probabilities. Model accuracy increases if the system is dependent on time in comparison with HMM. Three main properties that distinguish HSMM from HMM are:

1) The self-transition probability is assumed zero, i.e., no return to the same state.
2) The duration of a state determines whether it will remain in that state or move to the next. The duration for state $i$ is expressed as $p_i(d) \triangleq Pr\,(d|\,s_t = i)$ ,where $d = 1,2,\ldots,D$ and $D$ is maximum duration, with condition that $\sum_{d=1}^{D} p_i(d) = 1$.
3) The observation sequence between successive time instants is assumed to be independent in the emission probability, i.e., $b_i(O_{t:t+d-1}) = \prod_{k=t}^{t+d-1} b_i(O_k)$.

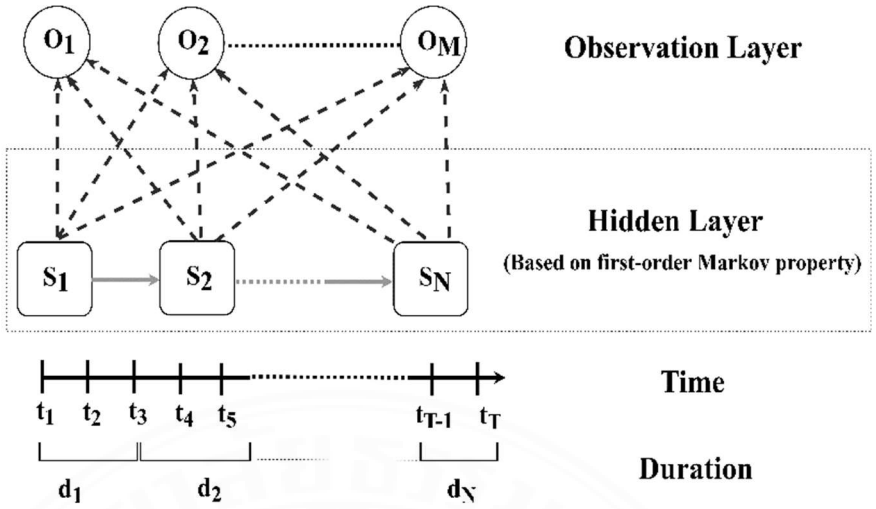



Fig 2.9 Hidden Semi Markov Model

HSMM $\Delta_{HSMM}$ consist of four mains of parameters (Sun et al. (2020)).

$$\Delta_{HSMM} = \{A, B, \pi, p\} \tag{47}$$

Where, $A$ represents transition matrix with duration probability $p$ for each state, $B$ signifies emission matrix, and $\pi$ is initial state probability. The sequential likelihood of the HSMM denoted as $P(O|\Delta_{HSMM})$ can determine through a forward-backward procedure. The forward-backward has two sub-divided variables: forward variable and backward variable. The HSMM forward-backward algorithm is defined as follows:

$$\alpha_t(i) = P(\, O_{1:t},\, S_i \; ends \; at \; t \mid \lambda) \; = \sum_{d=1}^{min\,(t,D)} \alpha^*_{t-d}(i) \; p_i(d) b_i(O_{t-d+1:t}) \tag{48}$$

$$\alpha^*_t(i) = P(O_{1:t}\,, S_i \; begins \; at \; t+1 \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i) \; a_{j,i}$$

$$t \; = \; 1, \ldots, T \qquad i \; = \; 1, \ldots, N \tag{49}$$

$$\beta_t(i) = P(O_{t:T} |\, S_i \; begins \; at \; t\,, \lambda) = \sum_{d=1}^{(T-t,D)} \beta^*_{t+d}(i) \; p_i(d) b_i(O_{t:t+d-1}) \tag{50}$$

$$\beta^*_t(i) = P(O_{t:T} \mid S_i \; ends \; at \; t-1, \lambda) = \sum_{j=1}^{N} a_{i,j} \; \beta_t(j)$$

$$t \; = \; T, \ldots, 1 \qquad i \; = \; 1, \ldots, N \tag{51}$$

To estimating the next parameters of the model i.e., $\hat{\pi}_i$, $\hat{a}_{ij}$, $\hat{b}_i(k)$, and $\hat{p}_i(d)$

$$\hat{\pi}_i = \frac{\pi_i \beta_1(i)}{\sum_{i=1}^{N} \pi_i \beta_1(i)} \qquad for\ t \leq 0 \tag{52}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta^*_{t+1}(i)} \quad 1 \leq i \neq j \leq N \tag{53}$$

$$\hat{a}_{ii} = 0 \quad i = 1,2,..,N \tag{54}$$

$$\hat{b}_i(k) = \frac{\sum_{t=1(s.t.O_t=k)}^{T} [\sum_{\tau=min\,(t,D)} \alpha^*_\tau . \beta_\tau - \sum_{\tau=min\,(t,D)} \alpha_\tau . \beta^*_\tau]}{\sum_{k=1}^{M} \sum_{t=1(s.t.O_t=k)}^{T} [\sum_{\tau=min\,(t,D)} \alpha^*_\tau . \beta_\tau - \sum_{\tau=min\,(t,D)} \alpha_\tau . \beta^*_\tau]} \tag{55}$$

$$\hat{p}_i(d) = \frac{\sum_{t=1}^{T-d+1} \alpha^*_{t-1}(i)\ \mathrm{p}_i(d)\ b_i(O_{t:t+d-1}) \beta^*_{t+d}(i)}{\sum_{d=1}^{D} \sum_{t=1}^{T-d+1} \alpha^*_{t-1}(i)\ \mathrm{p}_i(d)\ b_i(O_{t:t+d-1}) \beta^*_{t+d}(i)} \tag{56}$$

## 2.4 Tools

### 2.4.1 Hardware tools

Devices selected to accomplish the thesis goals is Wemos D1 ESP32. The Wemos D1 ESP32 board uses ESP32 module in the form factor of the Arduino Uno and can communicate with Wi-Fi and Bluetooth (BLE) network.

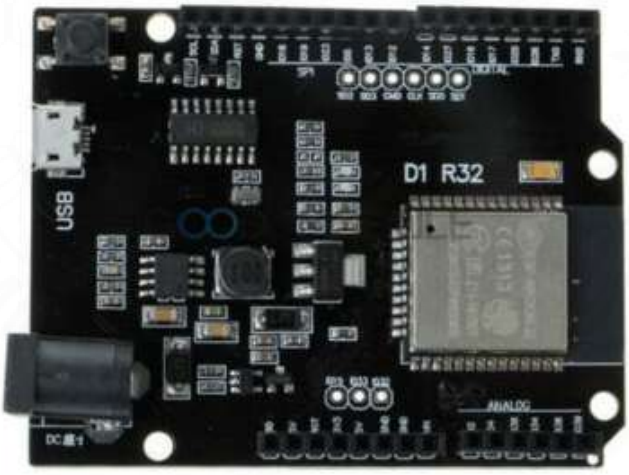



Fig.2.10 Wemos D1 ESP32

### 2.4.2 Software tools

The MATLAB software is used for extracting and analysing data using following functions:

1) Web access — *webread* function is used to read the data from RESTful web service.
2) Preprocessing data — *readtable* function to create table from extracted data, *table* array with named variables that can contain different types, *cell2table*

convert cell array to table, *findgroups* used to find groups and return group numbers, *discretize* function used group data into categories, and *groupcounts* function used number of group elements.

3) 2-D plots — *plot* creates a 2-D line plot of the data in y versus the corresponding values in x and *stairs* used to represent stairstep graph based on categories data.
4) Bluetooth Toolbox — simulates, analyses, and tests Bluetooth communication systems by simulating both links and networks. Run bit error rate and packet error rate simulations on those links with the toolbox. Set up piconet and mesh networks and test their performance in the presence of WLAN interference. Create localization scenarios and test with various impairments.
5) Classification tree — *fitctree* classify multiple classes, make a binary decision tree.
*6)* Normal distribution — *fitdist* fit the data to the probability distribution object.

# CHAPTER 3

# METHODOLOGY

## 3.1 Problem definition

Manufacturing are a complex process where products go through different working areas and operations, including picking, scheduling, and forwarding. The initial and final working areas are usually pre-defined for each manufactured product. Production flow refers to how raw materials are moved and processed in sequential order from one working area to the next until the final area. Workers or machines in each working area will spend time to complete their tasks, known as Cycle time. Cycle time is a critical KPI in the manufacturing industry because it describes temporal charatersitics of working areas which can be used in the forecasting of production efficiency.This study used a BLE-based indoor positioning system to track and collect product movement information to estimate the total cycle time of products. A BLE-based indoor positioning system uses a BLE tags being attached to products, and BLE scanners being strategically placed in working areas to detect BLE tags. However, the coverage area of the BLE scanner in the working area is limited. As a result, when the product is between two working areas, the BLE scanner may not be able to receive precise RSSI data, and the product location becomes unknown.

We compute the total cycle time of a product in terms of duration and interval in each area. Duration determines how long a product stays in a working area, whereas the interval determines how long the product stays between two working areas, such as the product is being transferred or put in a storage area. The HMM approach is used to achieve accurate indoor positioning. HMM combines fingerprints and motion sensors to solve missing data issues and overcome the high-dimensions classification challenge posed by a large number of scanners.The conventional HMM model is a time-independent model, i.e., the state transitions are time-independent, such that the cycle time of products can not be defined directly. Hence, this study formulates the product tracking problem as a HSMM model $\Delta_{HSMM} = \{A, B, \pi, p\}$ that defines its state transistion probability to depend on the sojour time, that is the time for workers or machines to complete their task.

A HSMM model can represent duration time in its context, but cannot capture the transition period between working areas HSMM. The Duration and Interval Hidden Markov Model (DI-HMM) $\Delta_{DI-H} = \{A, B, \pi, p, L\}$ defines the periods without observations as state intervals. In HSMM, the next state $S_j$ starts immediately after ending the present state $S_i$, but, next state $S_j$ in DI-HMM starts after a state interval that occurs after ending its previous state $S_i$. A state interval $I_{i,j}$ between two consecutive states where $i, j \in \{1,2, \dots, N\}$ is expressed by interval length probability $P(L_{i,j})$ assumed to be the Gaussian distribution. A time length of observation of the DI-HMM is $\sum(d_i + I_{i,j})$, where $d_i \in D$ and $I_{i,i+1}$stands for the time difference between ending-time of $S_i$ and stating-time of $S_j$.

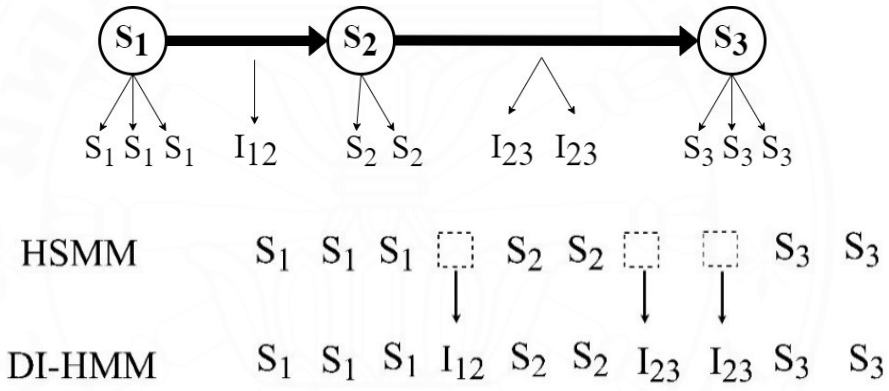



Fig 3.1 Sequence of DI-HMM (Narimatsu et al. 2017)

The objective of our work can be separated into two goals: first to develop a mathematical model that estimates the duration of each working area using collected RSSI data. Second, to further consider the interval concept in the mathematical model that reflects when products are kept temporarily between two working areas.

### 3.2 Research Methodology

This study aims to answer the following question: *how can we estimate the total Cycle Time using data acquired from a BLE-based indoor positioning system?* and is divided into two parts:

1) Modelling of duration using HSMM
2) Modelling of duration and interval using DI-HMM.

We assume no periods without observations in the first study, and we consider the situation of interval, i.e, no RSSI data for some period, in the second study.

### 3.2.1 Duration estimation

The main goal of the first study is to develop a mathematical model that uses RSSI data to estimate how long each working area takes to complete a particular operation. This RSSI data is analyzed using the concept of HSMM that characterizes the sequence in the production area statistically. Then the expected duration of product for each working area is predicted from the obtained HSMM model. The physical setting of our study is depicted in Fig 3.2.

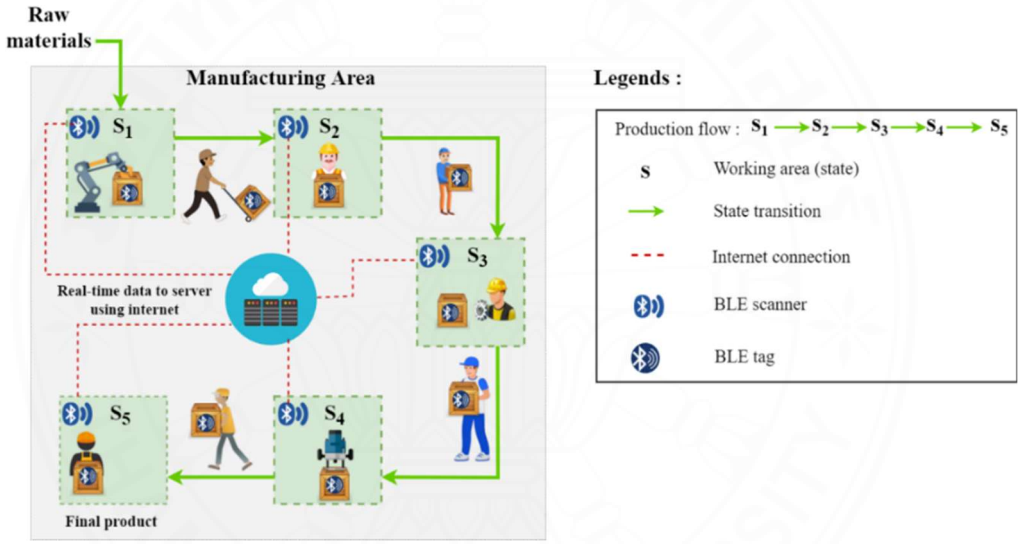



Fig. 3.2. Information flow for duration estimation

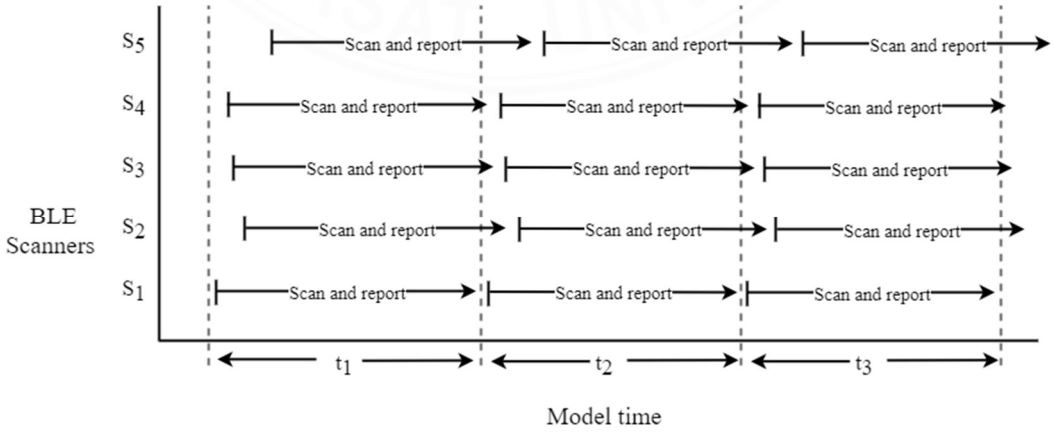



Fig.3.3. BLE scanning and sending data to server.

To model the production flow in the HSMM framework, each working area with BLE scanner represents state $S_i$ , where $i = 1,2,3,\ldots.N$ and $N$ denotes the number of working areas. Let all BLE scanners start independently with the same scanning/sending period. The RSSI data collected from all scanners are related to each other based on their scanning/reporting period. As shown in Figure 3.3, the data timing are allocated as discrete steps based on the scanning/reporting periods.

For a timing step $t$, multiple RSSI values observed at BLE scanners represent a group of observations $O = \{O_1^{m_1}, O_2^{m_2}, \ldots, O_t^{m_t}\}$ with $m_t$ observable mode. Let observable mode $m_t = (m_t^1, m_t^2, \ldots, m_t^M)$ represents the status of BLE scanners that report RSSI value. The observable mode $m_t^k = 0$ indicates that BLE scanner $k$ has not received signal at the time $t$, while $m_t^k = 1$ indicates that BLE scanner $k$ has received signal. This study assumes the following conditions that correspond to characteristics of manufacturing process.

1) Each working area with BLE scanner represents state $S_i$ , where $i = 1,2,3,\ldots.N$ and $N$ denotes the number of working areas. In this case, $M$ observed (RSSI) signals are generated based on the states, here $M = N$.
2) The transition matrix $A$ is fixed to reflect that the manufacturing process of similar products is sequential and identical. Since all product manufacturing begins in the $S_1$ working area, the initial state probability for the $S_1$ working area is 1 and 0 for the others.
3) The emission probabilities $B$ are assumed to be fixed, but may be updated periodically based on the collected RSSI data (Sun et al. 2020), as follows:

$$b_i\left(O_t^{m_t}\right) \triangleq P\left(S_t = i|\mu_i^{m_t}, \sum_i^{m_t}\right) \quad (57)$$

$$b_i\left(O_{t:t+d-1}^{m_t:m_{t+d-1}}\right) = \prod_{k=t}^{t+d-1} b_i(O_k^{m_k}) \quad (58)$$

The mean and covariance of the Gaussian component at state $i$ is $\mu_i^{m_t}$ and $\sum_i^{m_t}$, respectively with $m_t = (m_t^1, m_t^2, \ldots, m_t^M)$ observable mode.

For an HSMM model, $\alpha_t(i)$ and $\alpha_t^*(i)$ represent forward variables, while $\beta_t(i)$ and $\beta_t^*(i)$ represent backward variables. The forward algorithm estimates state $S_t$ by recursively computing the forward variable $\alpha_t(i)$ using only observation sequences

until time-length $t$. In the same way, we can compute the backward variable $\beta_t(i)$. Observable mode is used to define the HSMM forward-backward algorithm as follows:

$$\alpha_t(i) = P\left(O_{1:t}^{m_1:m_t}, S_i \text{ ends at } t|\lambda\right) = \sum_{d=1}^{min\,(t,D)} \alpha_{t-d}^*(i)\ p_i(d) b_i\left(O_{t-d+1:t}^{m_{t-d+1}:\,m_t}\right) \quad (59)$$

$$\alpha_t^*(i) = P\left(O_{1:t}^{m_1:m_t}, S_i \text{ begins at } t\ +1\ |\lambda\right) = \sum_{i=1}^{N} \alpha_t(i)\ a_{j,i} \quad (60)$$

where $t\ =\ 1, \dots, T \quad i\ =\ 1, \dots, N$

$$\beta_t(i) = P\left(O_{t:T}^{m_t:m_T} | S_i \text{ begins at } t, \lambda\right) = \sum_{d=1}^{(T-t,D)} \beta_{t+d}^*(i)\ p_i(d) b_i\left(O_{t:t+d-1}^{m_t:\,m_{t+d-1}}\right) \quad (61)$$

$$\beta_t^*(i) = P\left(O_{t:T}^{m_t:m_T} | S_i \text{ ends at } t-1, \lambda\right) = \sum_{j=1}^{N} a_{i,j}\ \beta_t(j) \quad (62)$$

where $t\ =\ T, \dots, 1 \quad i\ =\ 1, \dots, N$

We replace the $p$ with a re-estimated $\hat{p}$ for each area's cycle time. The duration probability $\hat{p}_i(d)$ matrix is re-estimated based on the forward-backward algorithm as:

$$\hat{p}_i(d) = \frac{\sum_{t=1}^{T-d+1} \alpha_t^*(i) p_i(d) \beta_{t+d}^*(i) b_i\left(O_{t+1:t+d}^{m_{t+1}:\,m_{t+d}}\right)}{\sum_{d=1}^{D} \sum_{t=1}^{T-d+1} \alpha_t^*(i) p_i(d) \beta_{t+d}^*(i)\ b_i\left(O_{t+1:t+d}^{m_{t+1}:\,m_{t+d}}\right)} \quad (63)$$

As mentioned in the above assumption, manufacturing process transitions $A$ occur in a set order from left to right, although the duration $p$ of each product varies. To find this hidden duration $p$ pattern where the sequence does not need to be estimated, use Algorithm 1.

**Input:** $O_{1:T}^z\ =\ \{o_1^z, o_2^z, \dots\dots o_T^z\}$ where $Z$ is number of training sequences

**Training phase:**

1. $p_{sum}\ =\ 0$
2. **for** $z$ = 1 to $Z$
3. Assign the HSMM parameters $\{\pi, A, B\}$
4. $z$ == 1 assign $p$ as uniformly distributed probability
5. **for** $t$ =1 to $T$
6. Calculate $\alpha_t$ , $\alpha_t^*, \beta_t$, and $\beta_t^*$ using (59), (60), (61), and (62)
7. **end**
8. Update parameter $\hat{p}$ using (63)
9. $p_{sum}\ =\ p_{sum}\ +\hat{p}$
10. $p_{avg}\ =\ {}^{p_{sum}}/_{Z}$
11. $p\ =\ p_{avg}$
12. **end**
13. **Output:** $\hat{p}$

Algorithm 1. Re-estimate the original probability of duration.

### 3.2.2 Duration and interval estimation

In this section, we will examine when a product may be either in a working area or a storage area and estimate how much time products spend in each using two different terms: duration and interval. The working area contains BLE scanners that collect RSSI data from BLE tags and determine the location of BLE tags based on strong RSSI values. While the movement and storage of products outside working areas will lead to weak and noisy RSSI values such that product location becomes unknown.

To model duration and interval with incomplete RSSI data we propose a classification tree to identify inside/outside scenarios based on BLE data, as shown in Figure 3.4. When RSSI data is classified as a positive case (working area), the training algorithm of HSMM model $\Delta_{HSMM} = \{A, B, \pi, p\}$ will proceed normally. When RSSI data is classified as a negative case (outside working area), the traning algorithm of HSMM model $\Delta_{HSMM} = \{A, B, \pi, p\}$ is paused and the counting of the timing steps will be used for the modeling of interval probability $L_{i,j}$. That is, the classification tree model supervises the flow of RSSI data to two training algorithms.

$$L_{i,j} = p(I_{i,j}) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(I_{i,j}-\mu)^2}{2\sigma^2}} \tag{64}$$

where $i = 1,2,..N-1$ and $j = 1,2,..N$

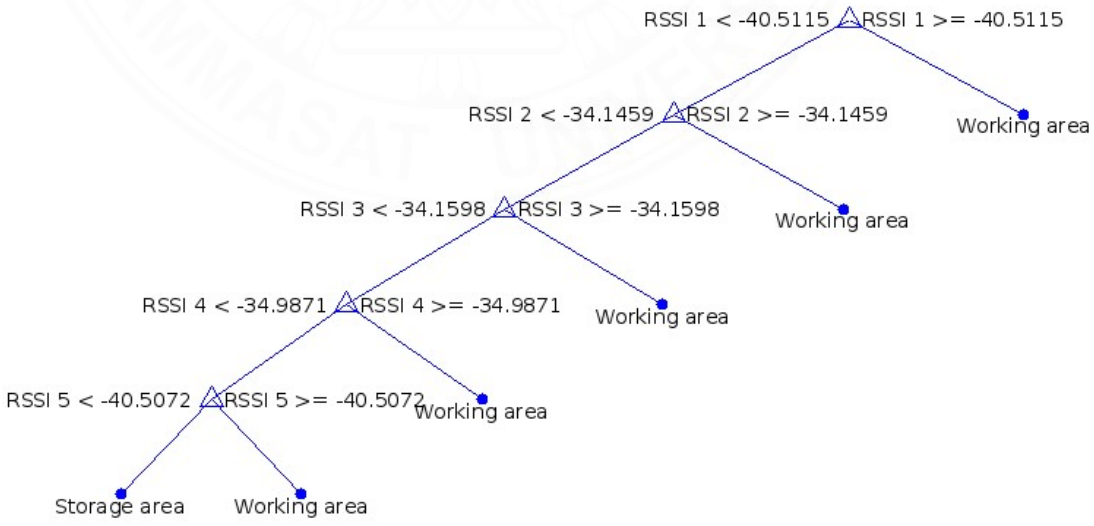



Fig 3.4. Classification tree split data into two area

**Input:** $O_{1:T}^{z} = \{o_1^z, o_2^z, \ldots\ldots o_T^z\}$ where $Z$ is number of training sequences

$ctree$ Classification tree model

**Training phase:**

1. $p_{sum} = 0$
2. $I_{sum} = 0$
3. **for** $z$ = 1 to $Z$
4. Assign the HSMM parameters $\{\pi, A, B\}$
5. $z$ == 1 assign $p$ as uniformly distributed probability
6. **for** $t$ =1 to $T$
7. $label = predict(ctree, o_t)$
8. **if** $label == \{working\ area\}$
9. Calculate $\alpha_t$ , $\alpha_t^*$, $\beta_t$, and $\beta_t^*$ using (59), (60), (61), and (62)
10. **else**
11. Calculate interval $I$
12. **end**
13. **end**
14. Update parameter $\hat{p}$ using (63)
15. Calculate $L$ using (64)
16. $p_{sum} = p_{sum} + \hat{p}$
17. $p_{avg} = {p_{sum}}/{Z}$
18. $p = p_{avg}$
19. **end**
20. **Output:** $\hat{p}$ and $L$

Algorithm 2. Re-estimate the duration and interval probabilities

# CHAPTER 4

# EXPERIMENT AND RESULTS

## 4.1 Experiment setting

In this study, we used three different settings to generate a sequence of RSSI observations:

1) Software
2) Real-world
3) Simulation.

HSMM will use software and real-world settings to generate a sequence of RSSI observations, whereas DI-HMM will use simulation based on the distance factor.

### 4.1.1 Software-based sequence generation

We use Algorithm 3 to generate the software-based sequence using the HSMM parameters $\boldsymbol{\pi}, \boldsymbol{A}$ and $\boldsymbol{p}$ as probability distributions, and multiple $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ for RSSI observations.

The duration probability $\boldsymbol{p} = \boldsymbol{Prob}(\boldsymbol{State}|\boldsymbol{duration})$ is based on a probability mass distribution in which the state is specified by the row and the duration by the column, with the maximum duration value of any state denoted to $\boldsymbol{D}$ and the sum of each row must equal one.

**Input:** $\Delta_{HSMM} = \{\boldsymbol{A}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi}, \boldsymbol{p}\}$, $\boldsymbol{N}$ is number of states and $\boldsymbol{M}$ is RSSI observations
**Output:** Hidden state $\boldsymbol{S}$ and RSSI observations $\boldsymbol{O}$

1. **for** $\boldsymbol{i} = \boldsymbol{1}$ to $\boldsymbol{N}$
2. **if** $\boldsymbol{i} == \boldsymbol{1}$
3. $\boldsymbol{S_{selected}}$ = Using the $\boldsymbol{\pi}$ initial probability, randomly select a state.
4. **else**
5. $\boldsymbol{S_{selected}}$ = randomly select state using previous state and $\boldsymbol{A}$ probability
6. **end**
7. $\boldsymbol{duration_{selected}}$ = Based on the $\boldsymbol{S_{selected}}$ and $\boldsymbol{p}$ probability, choose the duration at random.
8. **for** $\boldsymbol{k}$ = 1 to $\boldsymbol{duration_{selected}}$

9. **for** $\boldsymbol{m = 1}$ to $\boldsymbol{M}$
10. $\boldsymbol{o(m,k)}$ = Generate random value from $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ with $\boldsymbol{S_{selected}}$
11. **end**
12. **end**
13. $\boldsymbol{S_{duration}}$ (1: $\boldsymbol{duration_{selected}}$) = $\boldsymbol{S_{selected}}$
14. $\boldsymbol{S = [S, S_{duration}]}$
15. $\boldsymbol{O = [O, o]}$
16. **end**

Algorithm 3 Sequence generation using software.

We can see the variation in duration with respect to each state in fig 4.1.

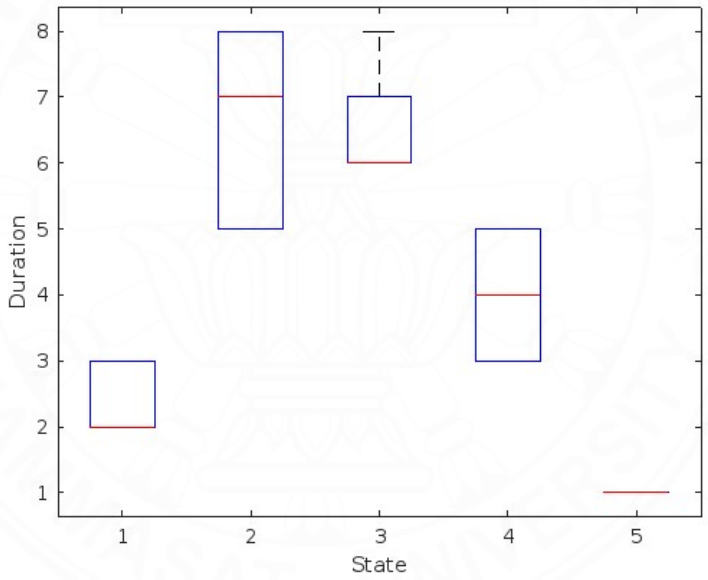



Fig 4.1. Duration variation for each state

### 4.1.2 Real-world

Physical experiments consist of three components: ESP32 boards as BLE tags and scanners, Wi-Fi access points as network, and Firebase as the database. The ESP32 boards are programmed differently to function as BLE scanners and BLE tags, respectively. The scanning time for all BLE scanners is 5 seconds. Figure 4.2 depicts the fundamental operation of a BLE scanner, which includes scanning, storing, and sending data to the Firebase's real-time database.

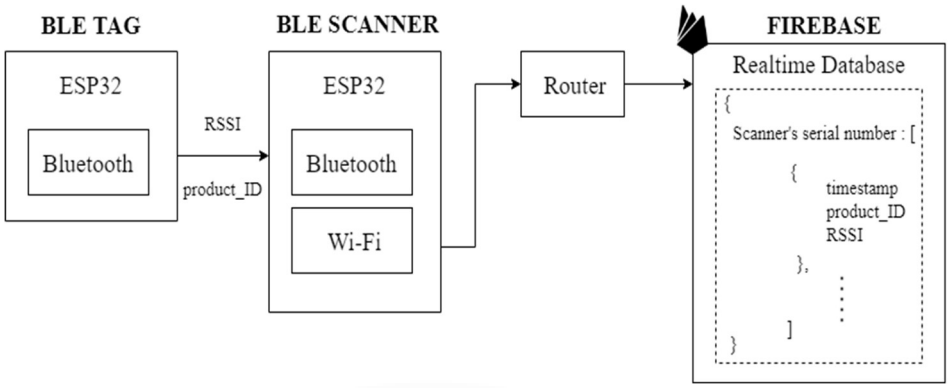



Fig.4.2. Fundamental operation of a BLE scanner

RSSI values from BLE scanners are collected and computed as patterns in fig 4.3 based on the placement of BLE tags at specific positions.

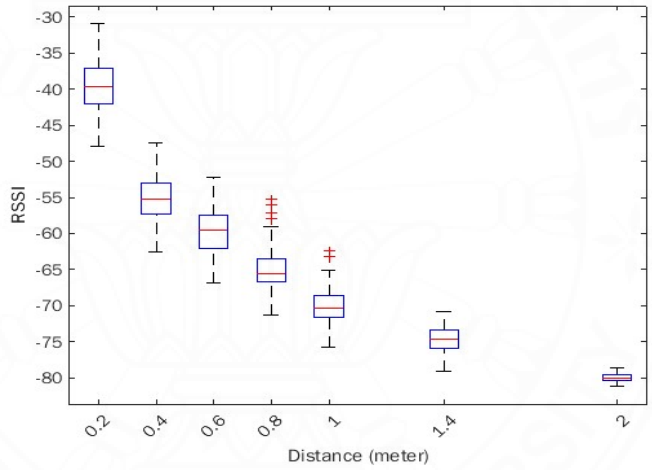



Fig. 4.3. Experiment BLE RSSI vs Distance

### 4.1.3 Bluetooth toolbox based BLE network.

MATLAB Bluetooth® Toolbox is used in this study to simulate end-to-end Bluetooth low energy (LE) transmission modes in the presence of a path loss model, RF impairments, and additive white Gaussian noise (AWGN). The bit error rate (BER), path loss, and distance between transmitter and receiver are all estimated using the simulation results. To run an end-to-end simulation with path loss, we use these steps:

1) Generate random bits.
2) Generate a Bluetooth BR/EDR waveform.
3) Add impairments.

4) Attenuate the waveform based on the path loss.
5) Add AWGN
6) Show the spectrum of transmitted and received waveforms.

The end-to-end example chain is depicted in Figure 4.4. Algorithm 4 show the process how to estimate the RSSI based on the scanner position with motion of tags.

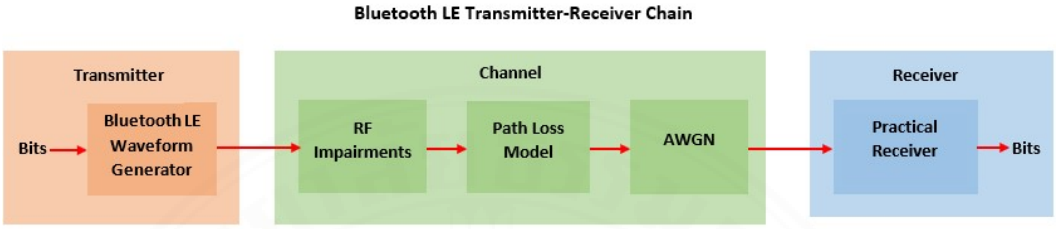



Fig. 4.4 Summary of the end-to-end chain.

**Input:** Environment = Industrial, $P_t = 0\ dB$ Tx power
$Scanner_{pos} = [x_1\ y_1; x_2\ y_3;\ ..; x_N\ y_N]$ , where $N$ is number of scanners.
$Tag_{Motion} = [x_1\ y_1; x_2\ y_3;\ ..; x_L\ y_L]$, where $L$ is total path steps.
$Tag_{at\ scanner} = [pos_1; pos_2;\ ..; pos_N]$ , $Interval = [I_{min}\ I_{max}]$
$Duration = [d_{1_{min}}\ d_{1_{max}}; d_{2_{min}}\ d_{2_{max}};\ ..; d_{N_{min}}\ d_{N_{max}}]$

**Output:** Hidden state $\boldsymbol{S}$ and RSSI observations $\boldsymbol{O}$

1. **for** $trajectory = 1$ to $L$
2. Tag position = $Tag_{Motion}(trajectory, :)$
3. Nearest scanner = $find\ (Tag_{at\ scanner} == trajectory)$
4. **if** isempty(Nearest scanner) == 1
5. t = randomly select $Interval$.
6. $S_{selected}$ = 'Storage area'
7. **else**
8. t = Select a random $Duration$ based on the Nearest scanner
9. $S_{selected}$ = Nearest scanner
10. **end**
11. $S_{time}$ (1: t) = $S_{selected}$
17. $\boldsymbol{S} = [\boldsymbol{S}, \boldsymbol{S_{time}}]$
12. **for** $scan = 1$ to $N$

13. Scanner position = $Scanner_{pos}\,(scan,:)$
14. Distance real = distance between the Tag position and Scanner position
15. **for** $k$ = 1 to t
16. Distance = Distance real ± randomly generated minor error
17. $PL_{dB}$ = End-to-End Bluetooth BR/EDR Simulation Procedure was used to calculate path loss. (Shown in figure 4.4)
18. $RSSI = P_t - PL_{dB}$
19. $o(scan,k) = RSSI$
20. **end**
21. $O = [O\ ,\ o]$
22. **end**
23. **end**

Algorithm 4 Generate RSSI using MATLAB Bluetooth toolbox.

## 4.2 Duration estimation

In the duration estimation, the performance of the HSMM model is evaluated based on generated and real-world data.

### 4.2.1 Numerical result of generated data

The generated data consists of 1000 data sequences. Each sequence contains two types of data: observed (RSSI) signals and state sequences. The data generateion steps assume the following parameters:

1) The number of working areas $N = 5$, and state $S_5$ is an absorbing (end) state. Based on our problem defined in section 3.2.1, $\pi$ and $A$ are declared as:

$$\pi = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{65}$$

2) The duration probability $p$ for state $i$ is specified using a probability mass distribution with duration $d = 1,2,..,8$, indicating that state $i$ will stay in the same state until duration $d$.

$$p = \begin{bmatrix} 0 & 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0.1 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0.1 & 0.2 \\ 0 & 0 & 0.3 & 0.4 & 0.3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{66}$$

3) RSSI observations are generated for each state using normal distribution probabilities with $\mu$ and $\sigma$ as shown in (67) and (68). In this case, $M$ observed (RSSI) signals are generated based on the states, here $M = N$. The Gaussian mixture model defined by (65)- (66) is used as the emission probability $b_i(O_t^{m_t})$ for state $i$.

$$\mu = \begin{bmatrix} -50 & -75 & -100 & -100 & -100 \\ -78 & -55 & -100 & -100 & -100 \\ -85 & -67 & -56 & -70 & -100 \\ -100 & -100 & -70 & -50 & -70 \\ -100 & -100 & -100 & -100 & -50 \end{bmatrix} \tag{67}$$

$$\sigma = \begin{bmatrix} 5 & 3 & 1 & 1 & 1 \\ 1 & 3 & 2 & 1 & 1 \\ 1 & 3 & 2 & 3 & 1 \\ 1 & 1 & 2 & 5 & 3 \\ 1 & 1 & 1 & 1 & 3 \end{bmatrix} \tag{68}$$

Use Algorithm 3 with the parameters listed above generate the state/observation sequences. A sample of generated data based on duration is shown in Figure 4.5.

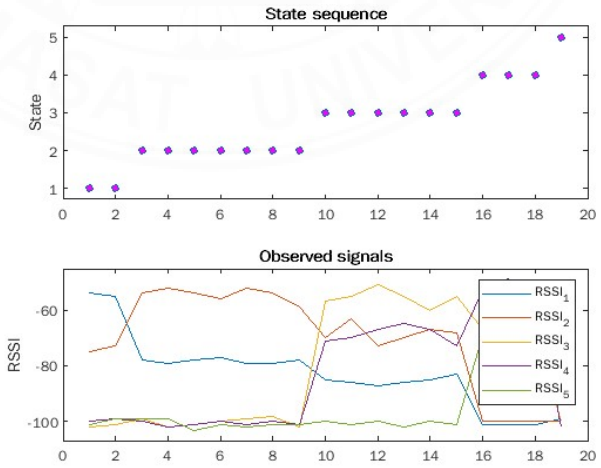



Fig. 4.5. Example of duration based generated data.

For each coupled state/observation sequence, the forward-backward algorithm (58)-(61) is applied with predefined HSMM parameters $\{\pi, A, B\}$ described above. Assume

that HSMM duration probabilities $\widehat{p_i}(d)$ for each state $i$ are bounded with $D \leq 8$. Then, the duration probabilities of each state are re-estimated using Algorithm 1. The comparison between $p$ and $\hat{p}$ is shown in figure 4.6.

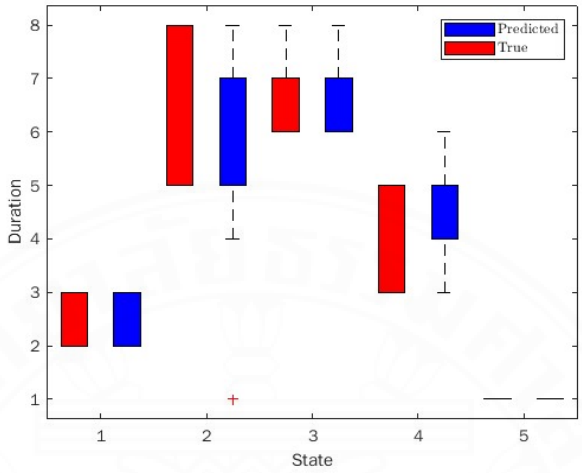



Fig. 4.6. Re-estimated duration parameter based on generated sequences.

To compare the similarity of two probability distribution functions $p$ and $\hat{p}$ this paper considers the Kullback–Leibler divergence (KLD) (Sun et al. 2020).

$$D_{KL}(p\|\hat{p}) = \sum_{i=1}^{N}\sum_{d=1}^{D} p_i(d)\ ln\ \frac{p_i(d)}{\widehat{p_i}(d)} \tag{69}$$

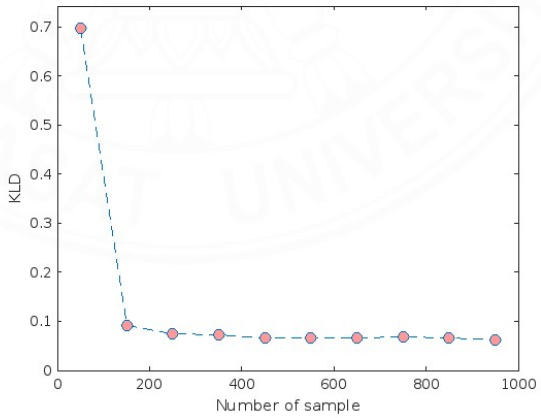



Fig. 4.7. $\hat{p}$ and the updated $p$ KLD value

The average KLD value for the entire system is approximately 0.0573, which is close to zero and proves that $\hat{p}$ is nearly equal to $p$. Figure 4.7 depicts the re-estimated $\hat{p}$ KLD value after each sequence versus different training data lengths. This resulted uses the Algorithm 1 and estimate the KLD based $p$ declared in (66).

### 4.2.2 Numerical results of simulated data

The MATLAB Bluetooth Toolbox is a useful tool for creating simulations of Bluetooth Low Energy (BLE) devices. By using this toolbox, we can simulate the movement of BLE tags and generate RSSI values based on their trajectory. In this study, we utilized the MATLAB Bluetooth Toolbox to simulate BLE transmission and obtain RSSI values while the tags were moved along a specified path. Fig. 4.8 illustrates the 2D coordinates of the BLE scanners and the moving BLE tags in the simulation environment.

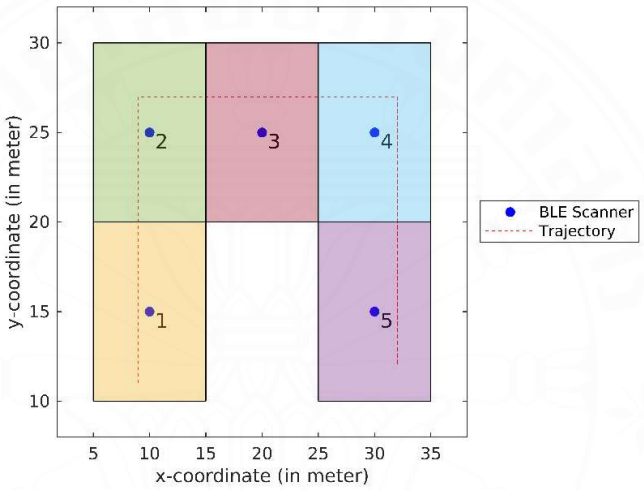



Fig. 4.8. 2-D Scanner Coordinates Layout for BLE

To generate state/observation sequences, we randomly advanced the BLE tags according to predefined durations, resulting in one thousand sequences. An example of observed RSSI values and states in a simulated sequence is shown in Fig. 4.9.

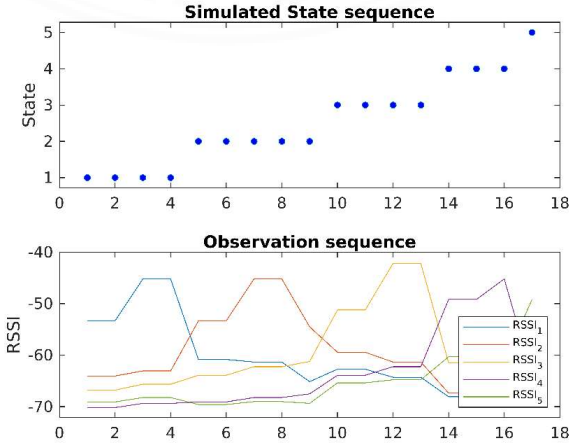



Fig. 4.9. Simulated Data Example.

In the simulated experiment, we generated 1000 sequences and used Algorithm 1 to parameterize an HSMM model. The average throughput time for the simulated sequences was 20.53 epochs, and the estimated throughput time using the HSMM model was 19.89 epochs. This shows that the estimated throughput time closely matches the simulated sequences. To assess the forecast accuracy, we conducted simulations and estimations for an additional 500 state sequences. We compared the forecasted and simulated sequences using run-length encoding (RLE) and vector distance. As we increased the number of samples from one to 500, the average vector distance decreased from 3.8531 to 0.9740. This reduction indicates that the forecasted and simulated sequences became more similar, indicating improved precision in the estimated duration as the number of samples increased.

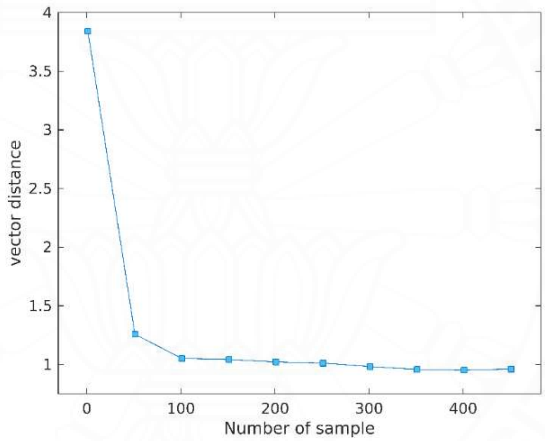



Fig. 4.10. Vector Distance Comparison: Simulated vs. Forecasted Durations

### 4.2.3 Numerical results of real-world data

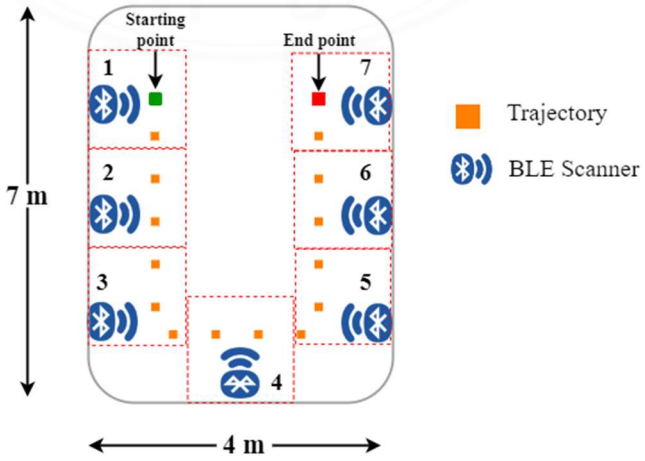



Fig. 4.11. Indoor localization with BLE

To collect the real-world data, seven BLE scanners are placed in separate areas and arranged in a U-shape over a $7 \times 4\,m$ an area so that only a few BLE scanners can scan BLE tags, as shown in figure 4.11. A total of 25 datasets are collected while moving BLE tags from one working area to the next in a predefined sequence. At the final area (BLE Scanner 7), the BLE tag will be shut down at the ending of the process. Based on the RSSI values collected, Figure 4.12 shows the pattern of estimating the object location.

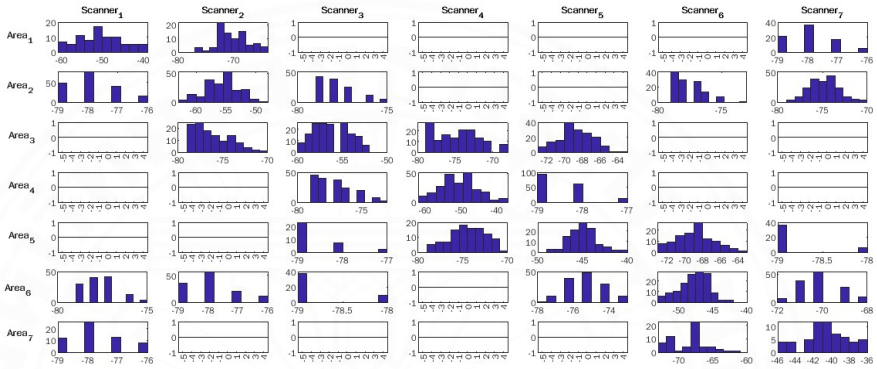



Fig. 4.12. RSSI pattern based on experimental data.

The procedure to process collected data is as follows:

1) Sort the data collected by each scanner using a timestamp, and then index to add a discrete-time column.
2) Retrieve discrete-time and RSSI values based on product_ID and rename the RSSI column to represent the scanner. Then, join the retrieved data based on discrete-time and product_ID.
3) Split the retrieved data based on product_ID and then join the based-on product_ID with a discrete-time column and multiple RSSI based on a scanner.

As described in the simulation, these data are used to train and re-estimate duration probabilities using HSMM. Figure 4.13 and table 4.1 depicts the performance of HSMM approaches based on one of the indoor localizations.

Table 4.1 Average Cycle Time

| **Working area** | **#1** | **#2** | **#3** | **#4** | **#5** | **#6** | **#7** | **Total Cycle Time** |
|---|---|---|---|---|---|---|---|---|
| **Real World average duration** | 3 | 6 | 6 | 8 | 4 | 4 | 1 | 32 |
| **HSMM average duration** | 4 | 5 | 6 | 7 | 4 | 4 | 1 | 31 |

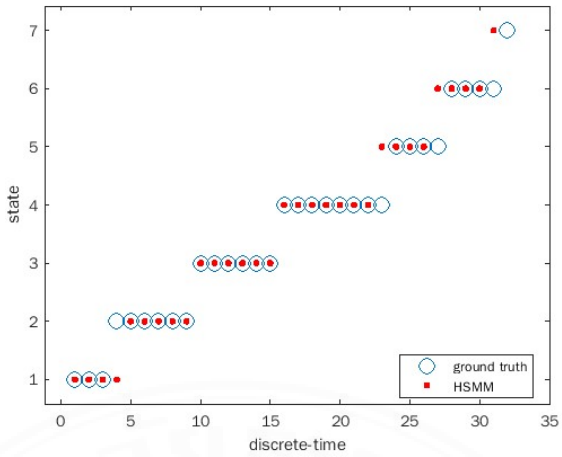



Fig. 4.13. Experimental performance comparison

### 4.3 Duration and interval estimation

The second objective of this study focuses on evaluating the performance of the DI-HMM model using RSSI sequences obtained from both simulated and real-world data. The aim is to assess the effectiveness of the DI-HMM in capturing and study the RSSI patterns in different scenarios.

#### 4.3.1 Numerical results of simulated data

The MATLAB Bluetooth toolbox is used to generate simulated data by providing the 2-D coordinates of the BLE scanner and moving points for BLE tags. The RSSI data is simulated using the MATLAB Bluetooth® Toolbox, which simulates and evaluates localization performance in the presence of channel and radio frequency (RF) impairments. The RSSI observation sequence is generated using algorithm 4 based on the position of a BLE scanner and the trajectory of a BLE tag setting, as shown in Figure 4.14.

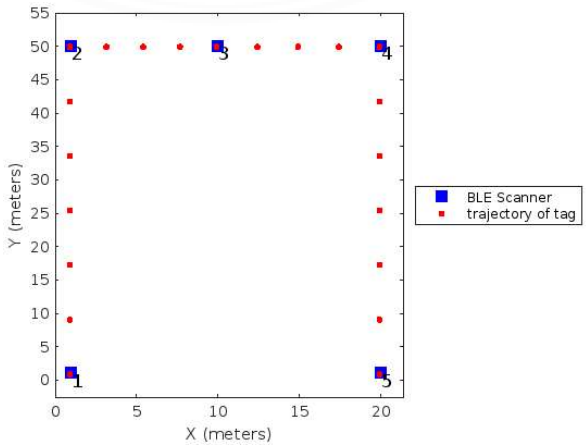



Fig. 4.14 BLE scanner position and BLE tag trajectory.

The MATLAB Bluetooth Toolbox is used to generate 500 data sequences based on the setup of the BLE scanners and BLE tag trajectory shown in Figure 4.14. Each simulated data sequence contains two types of information: observed (RSSI) signals and state sequences, as illustrated in Figure 4.15.

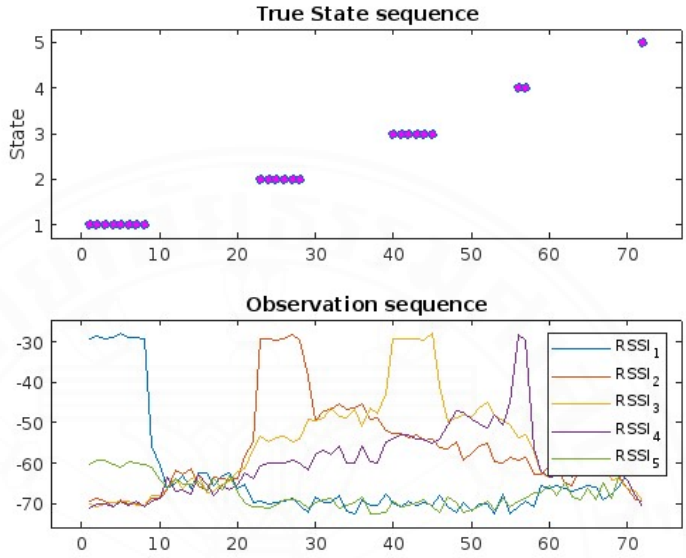



Fig. 4.15. Example of duration and intervals based simulated data

#### 4.3.1.1 Runtime estimation scenario

In runtime scenario, we use the RSSI signals and classify the activity at each time step using the following procedures:

1) Randomly select 100 observed signal sequences and estimate their duration and interval using Algorithm 2.
2) Classify the state sequence corresponding to the observed signal at each time step based on the estimated duration and intervals.

Figure 4.16 shows the comparison of the estimated and original states, and the accuracy of each sequence is determined using the multiclass accuracy formula. The multiclass accuracy formula considers the sum of True Positive and True Negative elements in the numerator and the sum of all confusion matrix entries in the denominator. The average accuracy of the 100 test sequences was approximately 95%.

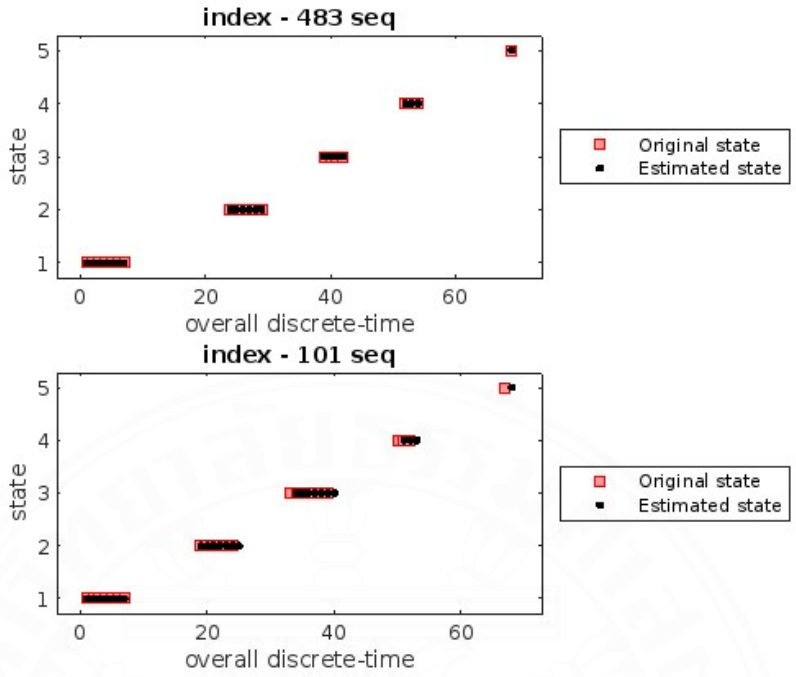



Fig 4.16 Comparison of the original and estimated states

#### 4.3.1.2 Offline forecast scenario

In offline forecast parameter $\hat{\boldsymbol{p}}$ and $\boldsymbol{I}$ obtained from algorithm 2 is used for forecasting the state sequences. We then, use Run-Length Encoding (RLE) to encode the original/predicted state sequence and then compare it using the vector distance. RLE is a simple lossless data compression algorithm for sequences that contain the same value multiple times. After RLE has encoded the input sequence, only its unique sequence number and count in series are stored as output. For example, RLE's input is "AAABBCCCC," and the output is "3A2B4C". After that, the vector distance is used to see how similar two or more RLE vectors are in multi-dimensional space. The following procedures are used in the offline estimation scenario:

1) The 200 state sequences are forecasted using estimated duration and interval probability distributions, and 200 original sequences are randomly selected.
2) The forecasted sequence and selected original sequences are encoded using Run-Length Encoding.
3) Based on the number of test sequences, calculate the average of the forecasted and original sequences with the encoded value.

4) After comparing the average encoded value of the forecast sequence and the original sequence using vector distance, shown in figure 4.17.

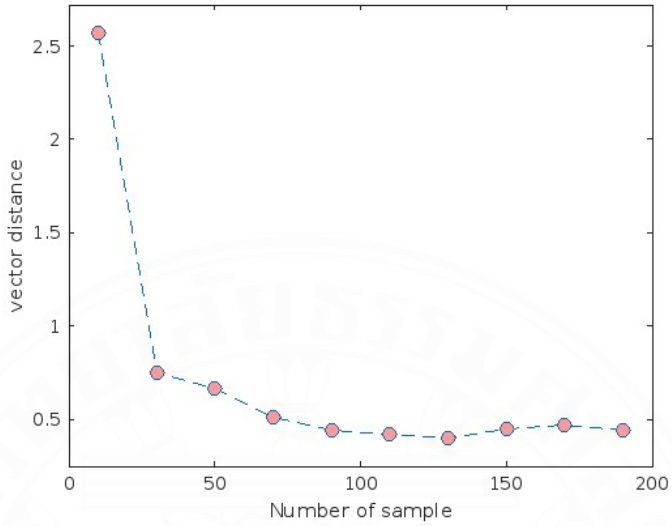



Fig 4.17 Vector distance for simulated and forecasted sequences.

The average vector distance value for the entire system is approximately 0.4717, which is close to zero and demonstrates that the estimated duration and interval are close to the original sequences used for the training of duration and interval parameters.

### 4.3.2 Numerical results of real-world data

In a real-world scenario, RSSI data from BLE devices is collected using various stations that perform different tasks. The experiment comprises three components: Wemos R32 boards serving as BLE tags and scanners, Wi-Fi access points, and the Firebase database. To conduct the experiment, four BLE scanners are strategically positioned in a U shape across a 12×5m area, as depicted in Figure 4.18.

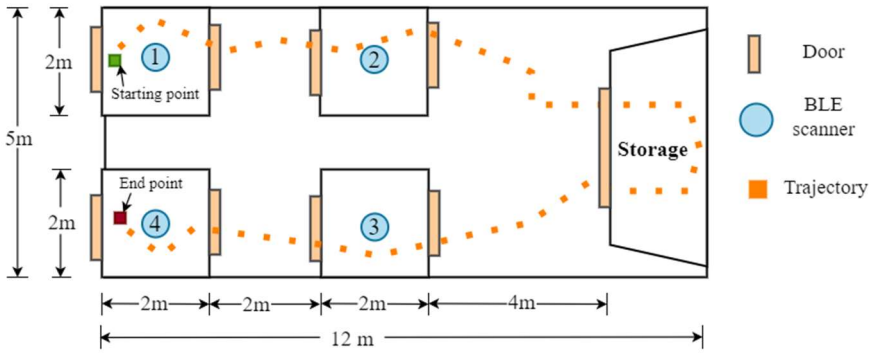



Fig 4.18. Scanner location in the experiment

During the experiment, data collection occurred every 5 seconds as a timing unit. Each BLE scanner scanned the surroundings, stored the collected data, and transmitted it to Firebase's real-time database. In a predetermined order, the BLE tags were moved from one working location to the next, and 20 datasets were collected from the designated area displayed in Figure 4.18.

Table 4.2 provides a comparison between the average state duration and interval values generated by our DI-HMM model and the actual data collected. The results demonstrate that our proposed method can achieve reasonable accuracy even with realistic and limited datasets. This comparison serves as evidence supporting the effectiveness of our approach in accurately estimating state durations and intervals.

Table 4.2. Duration and Interval of DI-HMM States

| **Working Area** | **Real** | **DI-HMM** |
|---|---|---|
| #1 (Duration-1) | 5.50 | 5.15 |
| #2 (Duration-2) | 5.35 | 5.80 |
| #3 (Duration-3) | 5.70 | 5.70 |
| #4 (Duration-4) | 3 | 1 |
| Storage (Interval) | 18.80 | 18.80 |
| Throughput-Time | 38.35 | 36.45 |

# CHAPTER 5

# CONCLUSIONS AND DISCUSSION

## 5.1 Conclusion

This paper studies the usage of an IPS-based BLE network to estimate manufacturing KPIs such as cycle time. Our approach is to model the manufacturing flow as a Hidden Semi Markov Model (HSMM) problem and then train a model using RSSI data from the BLE network. HSMM enables the system to learn and incorporate prior knowledge of state duration. Furthermore, this work examines when no RSSI data is available, i.e., when products are kept in temporary storage facilities away from BLE scanners. The problem is represented as state intervals by the Duration and Interval Hidden Markov Model (DI-HMM). Then, our first algorithm is extended to the DI-HMM problem using a combined architecture of a classification tree and HSMM.

In algorithm 1, the HSMM initial probability matrix and transition matrix are pre-specified because manufacturing production teams pre-define the unchangeable production flow. Emission probabilities are also pre-determined based on previously collected data. As a result, algorithm 1 re-estimates the duration probability distribution as the model's key parameter. The RSSI observation sequences are generated using HSMM parameters in the software. RSSI values are collected from ESP32 boards programmed as BLE scanners and BLE tags in real-world experiments.

Table 5.1 Generated data vs estimated HSMM duration

| **WORKING AREA** | **CYCLE TIME** | | | | | |
|---|---|---|---|---|---|---|
| | **Worst Result** | | **Best Result** | | **Average Result** | |
| | ***Generated*** | ***HSMM*** | ***Generated*** | ***HSMM*** | ***Generated*** | ***HSMM*** |
| **1** | 2 | 2 | 2 | 2 | 2.41 | 2.41 |
| **2** | 8 | 1 | 5 | 5 | 6.61 | 5.74 |
| **3** | 8 | 8 | 6 | 6 | 6.47 | 6.27 |
| **4** | 4 | 5 | 4 | 4 | 4.00 | 4.13 |
| **5** | 1 | 1 | 1 | 1 | 1.00 | 1.00 |
| **Total** | 23 | 17 | 18 | 18 | 20.49 | 19.55 |

Table 5.2 Real-world data vs estimated HSMM duration

| WORKING AREA | CYCLE TIME | | | | | |
|---|---|---|---|---|---|---|
| | Worst Result | | Best Result | | Average Result | |
| | *Real-world* | *HSMM* | *Real-world* | *HSMM* | *Real-world* | *HSMM* |
| **1** | 4 | 1 | 3 | 3 | 3.28 | 2.88 |
| **2** | 8 | 7 | 8 | 8 | 6.68 | 5.88 |
| **3** | 4 | 5 | 7 | 6 | 5.84 | 6.16 |
| **4** | 10 | 10 | 8 | 7 | 8.76 | 8.56 |
| **5** | 3 | 1 | 4 | 4 | 3.36 | 1.52 |
| **6** | 4 | 4 | 5 | 4 | 4.60 | 4.44 |
| **7** | 2 | 1 | 1 | 1 | 1.48 | 1.04 |
| **Total** | 35 | 29 | 36 | 34 | 33.99 | 30.48 |

As shown in tables 5.1 and 5.2, the HSMM method can re-estimate the duration of each working area using observed RSSI data.

When products are temporarily stored away from BLE scanners, Algorithm 1 cannot estimate the period when no RSSI data is available. Algorithm 2 uses a classification tree and HSMM architecture to solve the DI-HMM problem. The HSMM re-estimation using the classification tree is supervised by Algorithm 2, which allows the duration and interval probability distributions to be trained from RSSI data. The simulation setting is used in DI-HMM, and the MATLAB Bluetooth® Toolbox is used to generate RSSI observations, with the scanner position and tag movement defined in a 2D map. The estimated Cycle-time (CT) of the products in the manufacturing area is summation of state's duration and interval time, represented as $\sum_{q=1}^{N}(d_q + I_{q,q+1})$. Throughput-time (Th) is a ratio of time taken to complete production to a unit of the product so, the estimated CT of the product is equal to the Th. Work-in-Process (WIP) defines the current state of the product in the manufacturing area.

## 5.2 Future Work

The limitation of the IPS-based BLE network is that it does not define the exact location of the product in the storage area due to the short range of BLE. In the future,

combining two or more different wireless technologies, such as BLE and LoRaWAN, will improve the performance of Indoor Positioning Systems. LoRaWAN is a low-power, wide area networking protocol built on top of the LoRa radio modulation technique. Due to the long range of LoRaWAN, it can track areas and define the location of products where BLE is not available, such as the space between two working areas.

# REFERENCES

[1] Montaser, A., & Moselhi, O. (2014). RFID indoor location identification for construction projects. Automation in Construction, 39, 167-179.

[2] Xu, H., Ding, Y., Li, P., Wang, R., & Li, Y. (2017). An RFID indoor positioning algorithm based on Bayesian probability and K-nearest neighbor. Sensors, 17(8), 1806.

[3] Zhang, D., Yang, L. T., Chen, M., Zhao, S., Guo, M., & Zhang, Y. (2014). Real-time locating systems using active RFID for Internet of Things. IEEE Systems Journal, 10(3), 1226-1235.

[4] Huh, J. H., & Seo, K. (2017). An indoor location-based control system using bluetooth beacons for IoT systems. Sensors, 17(12), 2917.

[5] Sou, S. I., Lin, W. H., Lan, K. C., & Lin, C. S. (2019). Indoor location learning over wireless fingerprinting system with particle Markov chain model. IEEE Access, 7, 8713-8725.

[6] Lee, Y. H., & Marsic, I. (2018). Object motion detection based on passive UHF RFID tags using a hidden Markov model-based classifier. Sensing and biosensing research, 21, 65-74.

[7] Sun, S., Li, Y., Rowe, W. S., Wang, X., Kealy, A., & Moran, B. (2019). Practical evaluation of a crowdsourcing indoor localization system using hidden Markov models. IEEE Sensors Journal, 19(20), 9332-9340.

[8] Sun, S., Wang, X., Moran, B., & Rowe, W. S. (2020). A hidden semi-Markov model for indoor radio source localization using received signal strength. Signal Processing, 166, 107230.

[9] Narimatsu, H., & Kasai, H. (2015, July). Duration and interval hidden Markov model for sequential data analysis. In 2015 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[10] Ghimire, S., Thapa, G. B., Ghinire, R. P., & Silvestrov, S. (2017). A Survey on Queueing Systems with Mathematical Models and Applications. American Journal of Operation Research, 7(1), 1-14.

[11] Curry, G. L., & Feldman, R. M. (2010). Manufacturing systems modeling and analysis. Springer Science & Business Media. [pp. 45-59]

[12] Surian, D., Kim, V., Menon, R., Dunn, A. G., Sintchenko, V., & Coiera, E. (2019). Tracking a moving user in indoor environments using Bluetooth low energy beacons. Journal of biomedical informatics, 98, 103288.

[13] Yasmine, Rezgui, and Ling Pei. "Indoor fingerprinting algorithm for room level accuracy with dynamic database." 2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS). IEEE, 2016.

[14] V. Pierlot and M. Van Droogenbroeck, "A New Three Object Triangulation Algorithm for Mobile Robot Positioning," in IEEE Transactions on Robotics, vol. 30, no. 3, pp. 566-577, June 2014, doi: 10.1109/TRO.2013.2294061.

[15] Jung, Joonyoung, Dongoh Kang, and Changseok Bae. "Distance estimation of smart device using bluetooth." The Eighth International Conference on Systems and Networks Communications. 2013.

[16] Nguyen, Loc. "Continuous Observation Hidden Markov Model." (pp 1-83).

[17] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286.

# APPENDICES

# Modeling of Manufacturing Processes using Hidden Semi-Markov Model and RSSI data

Supachai Vorapojpisut
*Fac. of Engineering, Thammasat School of Engineering*
*Thammasat University*
Pathumthani, Thailand
vsupacha@engr.tu.ac.th

Karishma Agrawal
*Fac. of Engineering, Thammasat School of Engineering*
*Thammasat University*
Pathumthani, Thailand
karishma.agra@dome.tu.ac.th

2022 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP) | 978-1-6654-5727-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISAI-NLP56921.2022.9960270

***Abstract*— Temporal behaviors, e.g., cycle time and throughput, are among essential key performance indicators for the management of manufacturing processes. This paper presents a statistical model that captures the processing time spent throughout a production line using RSSI data acquired from Bluetooth Low Energy (BLE) network. First, a Hidden Semi-Markov Model (HSMM) is formulated based on the characteristics of production processes. Then, a learning problem is discussed for the re-estimation of state duration probability distribution using the forward-backward algorithm. The Kullback-Leibler Divergence is used to verify the accuracy by comparing between the original and estimated state duration probability distribution with a score of 0.0573. Finally, physical experiment was performed to evaluate the proposed method.**

***Keywords—Hidden Semi-Markov Model, Bluetooth Low Energy, Received Signal Strength Indicator, Learning problem***

## I. INTRODUCTION

Manufacturing is a complex process that schedules and traces different working areas and operations, including picking, assembly, and forwarding as shown in Fig.1. To understand what happens inside its production, each factory often implements one or more technologies to monitor the status and location of assets and products. Examples of such technologies include barcode, RFID, and indoor positioning systems that have their own advantages and disadvantages. This paper considers the Bluetooth Low Energy (BLE) technology [1] that is widely used as a location tracking and positioning system in industrial, commercial, and personal uses. Compared to other solutions, BLE technology exhibits several characteristics that meet the requirements of production tracking, e.g., suitable range (< 10 m), reasonable tag price (< $30), and battery lifetime (> 1 year).

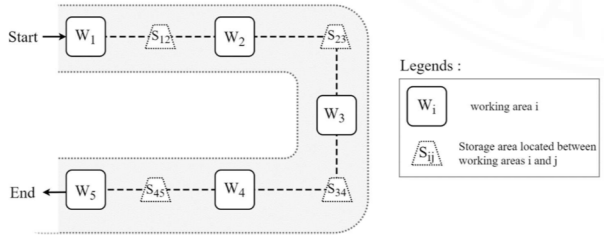



Fig. 1. An example of manufacturing process.

BLE is a wireless personal area network technology that operates in the 2.4GHz frequency band of the Industrial, Scientific, and Medical radio band (or the ISM band). For location tracking applications, the BLE network consists of two groups of devices, BLE tags that broadcast their identities and BLE scanners that discover the presence of nearby BLE tags. After each scanning period, each BLE scanner will report its finding as the list of found BLE tags' data, i.e., tag identity, packet-embedded data, and Received Signal Strength Indicator (RSSI) of such transmission. The timestamp is usually assigned on the server side when the report is received.

There are two major issues to be addressed for applying the BLE technology with product tracking in a factory. The first issue is RSSI varies [2] greatly even though the distance between scanner and tag does not change due to factors such as interference, random noise, and the multipath effects. Another issue is the receiver sensitivity which limits communication range may be different for devices. These results in the RSSI data to be acquired from scanners may be randomized and incomplete. Fig. 2 shows the fluctuating relationship between distance and RSSI values from a BLE experiment using ESP32 devices.

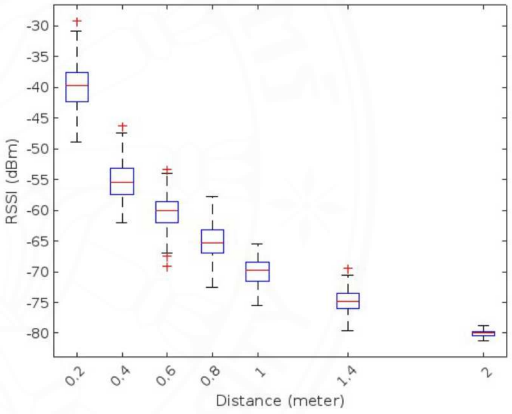



Fig. 2. Relationship between distance and RSSI values.

There are research works attempting to solve RSSI variations for object tracking scenarios. Triangulation and fingerprinting are two popular techniques [3,4] for indoor localization problems that estimate object coordinates from RSSI data obtained from multiple BLE scanners located at anchor points. However, those methods are not suitable for a factory environment that is significantly larger than the coverage area of BLE devices. Another approach is based on the proximity detection of tags [5], then analyzing the time of presence in each area. Even simpler to deploy, this approach constrains the understanding of how products are handled through the whole process. Therefore, this work aims to apply a statistical model that captures both RSSI variations and the flow of products along the production line.

This work aims to model temporal behaviors of manufacturing processes with respect to their benefits in process monitoring and improvement. The Hidden Markov Model (HMM) framework is selected as the mathematical tool to capture how products were handled throughout the process. The structure of the paper is outlined as follows. Section 2

978-1-6654-5727-9/22/$31.00 ©2022 IEEE

explains the notion of the Hidden Semi-Markov Model (HSMM) as an HMM extension for problems with timing dependency. Our problem of interest is formulated and solved as a learning problem in Section 3. Section 4 discusses numerical results for simulated sequences using MATLAB. Our work is concluded in Section 5.

## II. Hidden Semi-Markov Model

HMM [6] is a stochastic model in which the system being modeled as hidden states is assumed to be a Markov process with an observable layer of outcomes. The notion of HMM consists of 5-elements: state ($S$), initial state probability ($\pi$), transition probability matrix ($A$), observation state ($O$), and emission probability matrix ($B$). Even the HMM framework can be applied to real-world problems, the stationarity assumption, i.e., state transition is time-independent, prevents the capture of temporal behaviors that are often used as Key Performance Indicators (KPIs) in manufacturing processes.

This paper considers the notion of Hidden Semi-Markov Model (HSMM) [7] that redefines the transition of hidden states to depend on the sojourn time. Consequently, the processing time at each working area can be characterized as a corresponding element in the transition probability matrix. Contrary to the HMM notion, the HSMM notion prohibits the self-transition of hidden states. Then, how long the system will remain in the same state, or state duration, depends on the current state and its transition probabilities to other states. One merit of the aforementioned concept is to solve the issue of probability computation that decreases exponentially with the number of time steps.

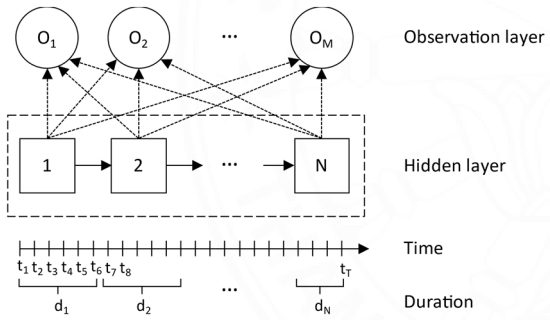



Fig. 3. Structure of Hidden Semi-Markov Model.

The HSMM is defined by a five-tuple: $\lambda = (S, \pi, A, O, B)$ where $S \in \{1,2,\ldots,N\}$ is a set of hidden states, $\pi$ is the initial probability matrix, A is state duration probability distribution, $O \in \{1,2,\ldots,M\}$ is a set of observations, and B is observation probability distribution. Let elements $a_{ij} \in A$; $i,j \in S$ are probabilities to transition from a hidden state i to j that depend on how long the system remains in the state i. Then, all elements $a_{ii}$ of A are supposed to be zero to prevent the self-transition of hidden states.

The choice of a probability distribution for both A, the state duration probability distribution, and B, the observation probability distribution, depends on the problem to be exploited. Since the BLE scanning pattern tends to be periodic, the notion of discrete time is considered throughout the paper, i.e. $t \in \{0,1,2,\ldots,\infty\}$. To simplify the computation, the state duration is supposed to be a random variable with discrete probability distribution bounded by an integer $D > 1$. Then, the state duration probability distribution A can be represented as an N×N×D matrix. That is, the matrix element A(i,j,d) denotes the probability to transition from a state i to j when the system remains in the state i for d step.

$$A(i,j,d) = P(s_{t+1} = j | s_{t-d+1} = \cdots = s_t = i) \quad (1)$$

The HMM/HSMM topology can be classified into three groups: left-to-right, parallel left-to-right, and ergodic models. For sequential processes, the left-to-right HSMM model is suitable due to its one direction of transition to the next state. Consequently, its sequences of states/observations will be bounded with respect to bounded state durations. That is, the state sequence $S^k$ is expressed by $s_1 s_2 \ldots s_{Tk}$ with its corresponding observation sequence $O^k = o_1 o_2 \ldots o_{Tk}$ where $Tk \leq ND$ denotes the upper bound of time spent in the process. For manufacturing processes, such coupled sequence aligns with data collected from production tracking systems.

There are three fundamental problems based on the HSMM framework widely considered in academic works.

- Likelihood problem: Given an HSMM parameter $\lambda$ and an observation sequence $O^k$, determine the likelihood $P(O^k|\lambda)$.
- Decoding problem: Given an observation sequence $O_k$ and an HSMM parameter $\lambda$, discover the best match of a hidden state sequence $S^k$.
- Learning problem: Given a collection of observation sequences $O_k$ and their known state sequences $S^k$, learn the HSMM parameters A and B.

This paper mainly considers the learning problem in which a collection of coupled sequences (known locations and their corresponding RSSI data) is used as the training dataset to find the state duration probability matrix.

## III. Problem Statement

The physical setting of our study is depicted in Fig. 4 where one BLE scanner is installed in each working area and one BLE tag is attached to each product. When a BLE tag comes within range of any BLE scanner, the broadcasting data is captured and collected. The data reported by BLE scanners includes scanner ID, timestamp, and a list of product IDs and their corresponding RSSI values. BLE tags may not be visible for every BLE scanner due to the limited range of BLE communication, but each BLE tag is assumed to be visible to at least one BLE scanner. We assume that BLE tags will always start at the initial working area and end at the final working area. Therefore, the period of data collected from each BLE tag will be bounded.

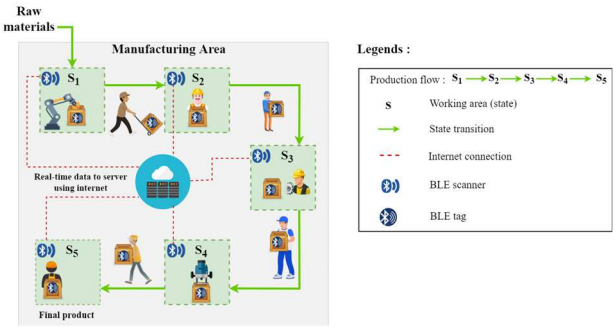



Fig. 4. BLE devices in a manufacturing process.

### A. Assumptions

To formulate manufacturing processes in the HSMM framework, the following assumptions have been made regarding how raw materials and parts are assembled into final products through organized jobs of a production line.

- One BLE scanner is installed for each working area. Therefore, the set of hidden states S and observations O are similar, i.e., S = O = {1,…,N}.
- The process is sequential, i.e., each BLE tag will always start at area 1 and end at area N. Then, A(i,i+1,d) ≥ 0 and A(i,j,d) = 0 where j ≠ i+1 denotes the state transition will occur only from state i to i+1.
- The initial state probability $\pi_1 = 1$ denotes every sequence will always start at state 1.

Note that the assumptions can be generalized for multiple production lines or product families by grouping each variant based on product IDs.

### B. HSMM Model

By equipping one BLE scanner with each working area, the structure of HSMM becomes N hidden states and N observations. Assume that BLE scanners start independently with the same scanning/reporting period. Then, RSSI data collected from BLE tags is related to others based on their timing slot. As shown in Fig. 5, the data timing is grouped as discrete steps regarding the scanning/reporting period.

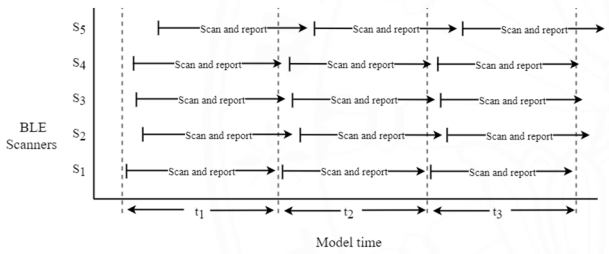



Fig. 5. Timing pattern of BLE scanners.

By storing data from BLE scanners in a database, RSSI values can be queried and composed into a sequence of observations for each BLE tag. Let $o_t^k$ be a collection of RSSI values of k[th] BLE tag at a timing step t:

$$o_t^k = \{ RSSI_{1t}^k, RSSI_{2t}^k, \dots, RSSI_{Nt}^k \}, \tag{2}$$

where $RSSI_{nt}^k$ is the RSSI value of the k[th] BLE tag observed at the n[th] BLE scanner. Then, the sequence of observations $O^k$ is expressed by $\{ o_1^k, o_2^k, \dots, o_{Tk}^k \}$ where Tk denotes the bounded time step for the k[th] BLE tag. To avoid missing element in the sequence $O^k$, let $o_t^k$ with unobservable $RSSI_{nt}^k$ be filled by the value of receiver sensitivity.

To model the relationship between BLE tag locations and observed RSSI values, the observation probability distribution B is supposed to be a multivariate Gaussian [8] and time-invariance. Then, $b_n(o_t^k)$ is defined as the probability that the RSSI pattern $o_t^k$ is observed in the n[th] working area:

$$b_n(o_t^k) = P(o_t^k \mid s_t = n, \mu_n, \Sigma_n), \tag{3}$$

where $\mu_n = \{\mu_{1n},\dots,\mu_{Nn}\}$ and $\Sigma_n = \{\Sigma_{1n},\dots,\Sigma_{Nn}\}$ are the mean and covariance of multivariate Gaussian distribution of RSSI values observed at the n[th] working area.

### C. Algorithm

To capture temporal behaviors of the process, this paper considers the learning problem of the HSMM framework. The main parameter to be solved is the state duration probability distribution A in which its element A(i,i+1,d) where $1 \le d < D$ represents the probability of processing time of the i[th] area. The observation probability distribution B is obtained by performing data collection experiments and assumed to be stationary during the time of interest. Due to the sequential process assumption, the state duration probability distribution A can be simplified by decomposing into a probability distribution of state transition $p_i(d); i \in S, 1 < d \le D$ with respect to the dwell time d in state i and a simplified strictly upper triangular matrix $\bar{A}$.

$$\bar{A}(i,j) = \begin{cases} 1\,; & j = i+1 \\ 0; & elsewhere \end{cases} \tag{4}$$

The re-estimation of the probability distribution $p_i(d)$ from coupled sequences $(S^k, O^k)$ is performed based on the computation of forward-backward variables [7]. The forward variable $\alpha_t(i)$ is the likelihood of being in state i considering $O_{1:t}$ as partial observations up to time t with respect to a given HSMM parameter λ.

$$\begin{aligned} \alpha_t(i) &= P(O_{1:t}, s_t = i|\lambda) \\ &= \sum_{d=1}^{\min(t,D)} \alpha_{t-d}^*(i) p_i(d) \prod_{j=t-d+1}^{t} b_i(o_j), \end{aligned} \tag{5}$$

where $\alpha_t^*(j) = \alpha_t(i)$ if j = i+1 and 0 otherwise denotes the probability for the transition into state i at time t+1. While the backward variable $\beta_t(i)$ is the likelihood of being in state i considering $O_{t:T}$ as partial observations from time t to the end with respect to a given HSMM parameter λ.

$$\begin{aligned} \beta_t(i) &= P(O_{t:T}, s_t = i|\lambda) \\ &= \sum_{d=1}^{\min(T-t,D)} \beta_{t+d}^*(i) p_i(d) \prod_{j=t}^{t+d-1} b_i(o_j), \end{aligned} \tag{6}$$

where $\beta_t^*(j) = \beta_t(i)$ if j = i+1 and 0 otherwise denotes the probability for the system to leave state i at time t-1.

The probability distribution $p_i(d)$ can be updated by computing (5) and (6) for each coupled sequence $(S^k, O^k)$, then solving for its re-estimated value.

$$\hat{p}_i(d) = \frac{\sum_{t=1}^{T-d+1} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{j=t+1}^{t+d} b_i(o_j)}{\sum_{d=1}^{D} \sum_{t=1}^{T-d+1} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{j=t+1}^{t+d} b_i(o_j)} \tag{7}$$

The learning problem is outlined as follow.

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

**Input:** A, B, coupled sequences $(S^k, O^k)$; k = 1,2,…,K
**Training:**
1) Populate $p_i(d)$ with uniform distribution values.
2) Initialize $\hat{P}_i(d)$ matrices
3) Loop for k = 1,…,K
  3.1) Loop for t = 1,…,T
    3.1.1) Compute $\alpha_t(i), \alpha_t^*(t), \beta_t(i), \beta_t^*(t), \hat{p}_i(d)$
    3.1.2) $\hat{P}_i(d) = \hat{P}_i(d) + \hat{p}_i(d)$
  3.2) $p_i(d) = \hat{P}_i(d)/T$
**Output:** $p_i(d)$

## IV. Numerical Results

Two numerical studies were performed to evaluate our proposed algorithm using data from simulated and real-world scenarios.

### A. Simulated data

The first study was to statistically generate state and RSSI sequences using given parameters A, B and $p_i(d)$. HSMM parameters were defined as follows:

- Number of working areas N = 5.
- State duration is bounded by D = 8.
- Probability distribution matrix $P = \{p_i(d)\}$ for the system to leave state i after time d is given by

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

- RSSI values are generated using a Gaussian mixture model with mean and variance:

$$\mu = \begin{bmatrix} -50 & -75 & -100 & -100 & -100 \\ -78 & -55 & -100 & -100 & -100 \\ -85 & -67 & -56 & -70 & -100 \\ -100 & -100 & -70 & -50 & -70 \\ -100 & -100 & -100 & -100 & -50 \end{bmatrix}$$

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

Based on the given parameters, 1000 coupled sequences are generated with an example shown in Fig. 6.

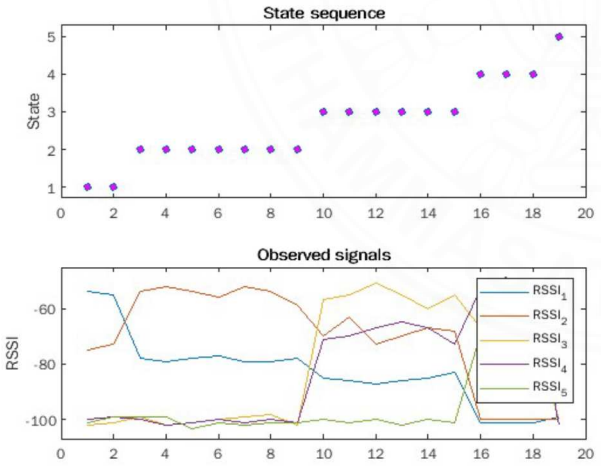



Fig. 6. Example of simulated data.

Algorithm 1 is used with the simulated data to estimate the probability distribution matrix $\hat{P}$. To compare the similarity of the original P and the re-estimated $\hat{P}$, this paper considers the Kullback-Leibler divergence [7] as the distance metric.

$$D_{KL}(p_i(d)||\hat{p}_i(d)) = \sum_{d=1}^{D} p_i(d) \ln \frac{p_i(d)}{\hat{p}_i(d)} \qquad (8)$$

The average value of KLD is 0.057 which is close to zero, hence proves that $p_i(d)$ is nearly equal to $p_i(d)$. To visualize the closeness, Fig. 7 compares the range of state durations between original sequences and the sequences generated from the re-estimated parameters $\hat{P}$.

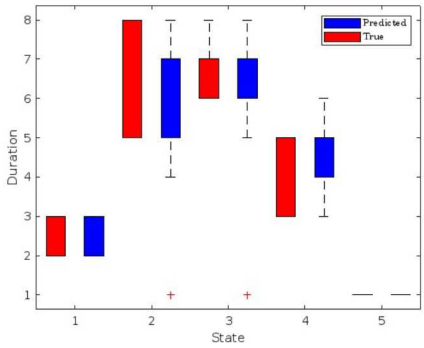



Fig. 7. Comparison of state durations generated from original and re-estimated parameters.

Fig. 8 depicts the trend of average KLD values regarding the number of coupled sequences used in the training phase. This shows that our proposed algorithm converges quite fast with a reasonable amount of data.

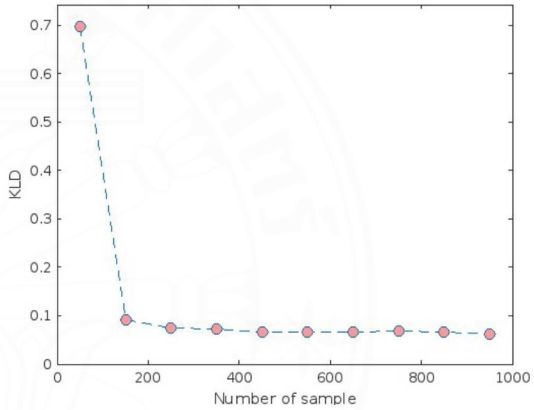



Fig. 8. Trend of average KLD values.

### B. Real-world data

The second study was to investigate real-world scenarios by collecting RSSI data from BLE devices brought by subjects through stations with different activities. Physical experiments consist of three components: ESP32 boards as BLE tags and scanners, Wi-Fi access points, and Firebase as the database. Seven BLE scanners were placed in a U-shape over a $7 \times 4\ m$ an area so that only a few BLE scanners can scan BLE tags, as shown in figure 9.

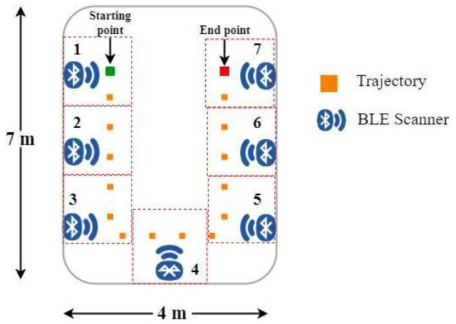



Fig. 9. Allocation of stations in the experiment.

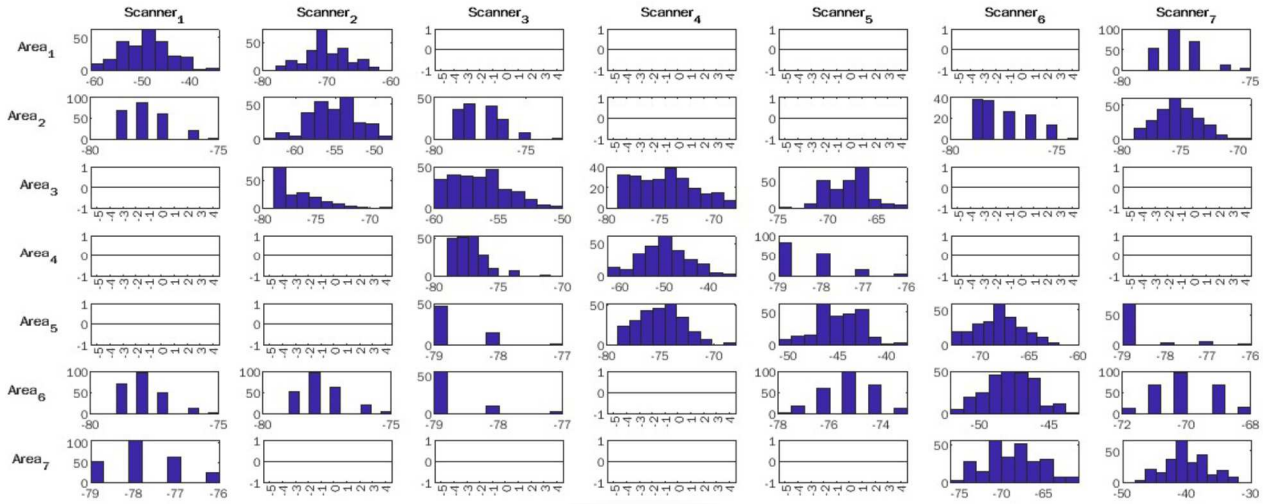



Fig. 10. Pattern of RSSI values as observations.

The ESP32 boards are programmed to function as BLE scanners and BLE tags, respectively. Each BLE scanner is programmed to scan, store, and send data to the Firebase's real-time database. The scanning period for all BLE scanners is 5 seconds. A total of 25 datasets are collected while moving BLE tags from one working area to the next in a predefined sequence. At the final area, the BLE tag will be reset to mark the end of the process. Timing mentioned in the following was expressed as the unit of scanning period, i.e., 5-second.

The dataset is used for the fitting of Gaussian mixture model B and the estimation of $\hat{p}_i(d)$. Fig. 11 illustrates the alignment between one experimental data and a state sequence obtained from the trained HSMM model.

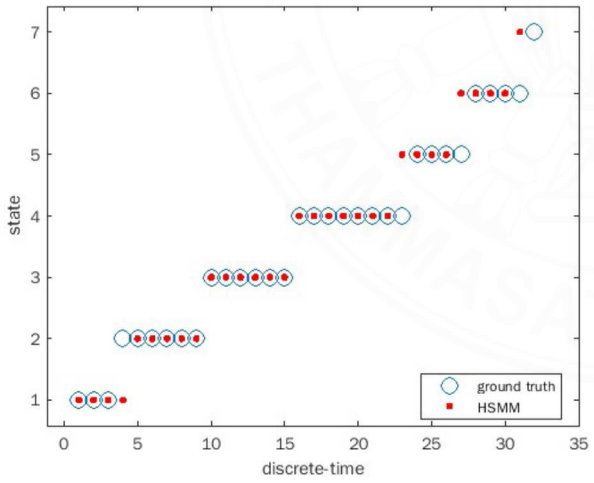



Fig. 11. Alignment of hidden states.

Table 1 compares between average values of state durations between the collected data and those generated by our HSMM model. The average cycle time can be computed from the summation of state durations, of which are 33.99 and 30.48, respectively.

TABLE I. STATE DURATIONS (UNIT OF 5 SECS)

| Duration | #1 | #2 | #3 | #4 | #5 | #6 | #7 |
|---|---|---|---|---|---|---|---|
| Real | 3.28 | 6.68 | 5.84 | 8.76 | 3.36 | 4.60 | 1.48 |
| HSMM | 2.88 | 5.88 | 6.16 | 8.56 | 1.52 | 4.44 | 1.04 |

## V. CONCLUSION

This paper presents an approach to capture the temporal behaviors of manufacturing processes as an HSMM model. Based on the learning problem, the probability distribution $\hat{p}_i(d)$ can be estimated from RSSI data collected from experiments. Then, the HSMM model can be used in many aspects. One possible application is to generate the sequence of state durations that can be used to compute the manufacturing cycle time. This will gain advantages for the estimation of productivity in production lines.

## REFERENCES

[1] K. E. Jeon, J. She, P. Soonsawad and P. C. Ng, "BLE Beacons for Internet of Things Applications: Survey, Challenges, and Opportunities," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 811-828, April 2018.

[2] T. Wattananavin, K. Sengchuai, N. Jindapetch, and A. Booranawong, "A Comparative Study of RSSI-Based Localization Methods: RSSI Variation Caused by Human Presence and Movement," Sensing and Imaging, vol. 21, no. 31, Dec. 2020.

[3] P. Kriz, F. Maly, and T. Kozel, "Improving Indoor Localization Using Bluetooth Low Energy Beacons," Mobile Information Systems, vol. 2016, Apr. 2016.

[4] Z. Jianyong, L. Haiyong, C. Zili and L. Zhaohui, "RSSI based Bluetooth low energy indoor positioning," 2014 Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN), 2014, pp. 526-533.

[5] A. Rácz-Szabó, T. Ruppert, L. Bántay, A. Löcklin, L. Jakab, and J. Abonyi, "Real-Time Locating System in Production Management," Sensors, vol. 20, no. 23, pp. 6766, Nov. 2020.

[6] K. Ding, X. Zhang, F. T. S. Chan, C. -Y. Chan and C. Wang, "Training a Hidden Markov Model-Based Knowledge Model for Autonomous Manufacturing Resources Allocation in Smart Shop Floors," IEEE Access, vol. 7, pp. 47366-47378, 2019.

[7] S-Z. Yu, "Hidden semi-Markov models," in Artificial Intelligence, vol. 174, no. 2, pp.215-243, Feb. 2010.

[8] 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, Jan. 2020.

# 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 xx xxx 20xx; received in revised form xx xxx 20xx; accepted xx xxx 20xx

## Abstract

This study proposes a statistical model that uses received signal strength indicator (RSSI) data acquired from a Bluetooth Low Energy (BLE) network to characterize the temporal characteristics of an entire production process. A Hidden Semi-Markov Model (HSMM) is formulated based on the production process characteristics, and the forward-backward algorithm is used to re-estimate the probability distribution of state durations. The proposed method is validated using numerical, simulation, and real-world experiments with promising results. The Kullback-Leibler Divergence (KLD) score is 0.1843 for the numerical result, and the model achieved an average vector distance score of 0.9740 in the simulation result. 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 provides a useful tool for predicting a production line's cycle time and throughput time.

**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, each of which has a specific set of tasks/operations that transform raw materials into finished products, as shown in Fig. 1. To ensure that the production process is completed on time, it is critical to have accurate and up-to-date product status and 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 a noisy environment. Consequently, the application of BLE tracking is much more limited compared to barcode/RFID.

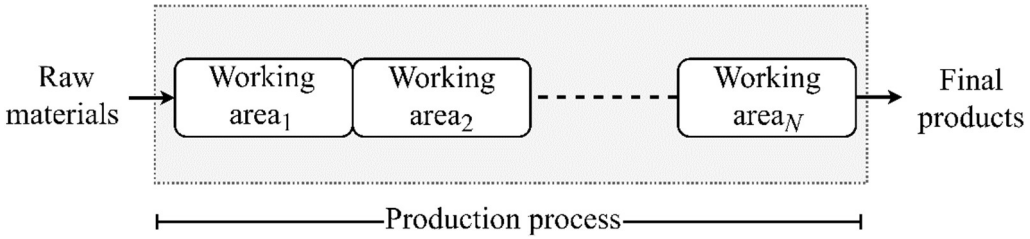



Fig. 1 General example of a production process.

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

Key Performance Indicators (KPIs) of a production process can be defined using temporal behaviors [3-4]. Examples of KPIs for production processes are cycle time and throughput time. Cycle time [3, 5] refers to the amount of time it takes to complete one cycle of a particular operation, while throughput time [4] is the total amount of time it takes to process a certain material from raw material to a final product. Based on these KPIs, manufacturers can identify areas for improvement, reduce costs, and optimize productivity. Therefore, this information is especially important for manufacturers who attempt to produce high-quality products in the shortest possible timeframe to remain competitive.

The traditional method of manually monitoring products to calculate cycle time in manufacturing had various limitations, such as the potential for human errors and limited data collection. At present, manufacturers have adopted a more precise and efficient approach by utilizing barcode/RFID [1-2] and statistical methods to analyze historical data for cycle time calculation. 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 [6] 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 by using RSSI values to estimate cycle time and throughput time, which is a new approach compared to previous studies [7-8] that mainly focused on tracking and monitoring activities. This paper provides a more detailed and comprehensive version of the work [9] presented at the iSAI-NLP conferences, 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. Section 3 defines and explains 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 production product can be computed using the mathematical solutions presented in this paper.

## 2. Backgrounds

### *2.1. Bluetooth Low Energy (BLE) technology*

Bluetooth Low Energy (BLE) [8-9] technology offers wireless connectivity with low cost, low power consumption, and simple installation, making it 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 and packet-embedded data, and RSSI data [10-11]. 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, where higher values mean a stronger and more dependable connection. Decibels (dBm) is the standard unit of measurement for RSSI.

There are two major problems in using 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 [10]. Secondly, the inconsistent receiver sensitivity of each device [10-11] 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 using WeMos R32 (ESP32) boards.

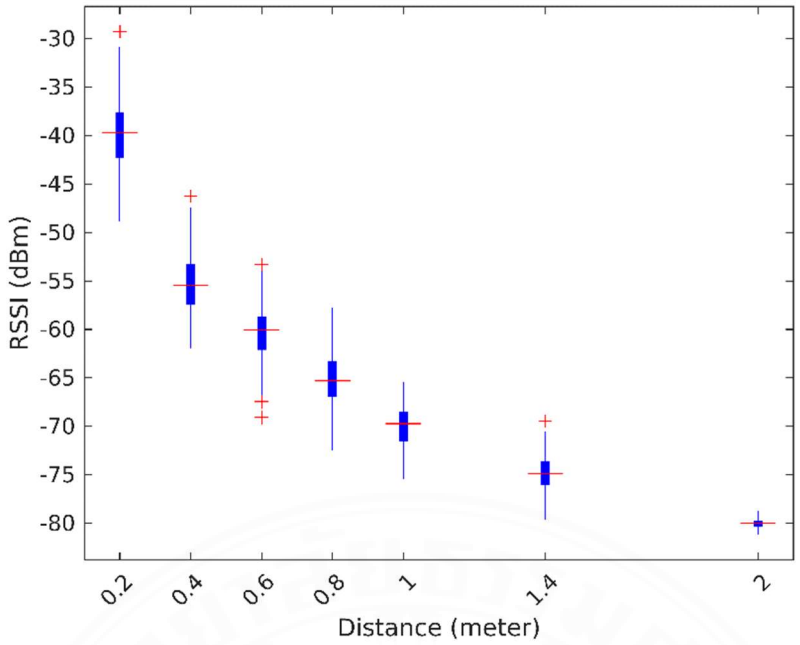



Fig. 2 RSSI vs Distance of BLE devices.

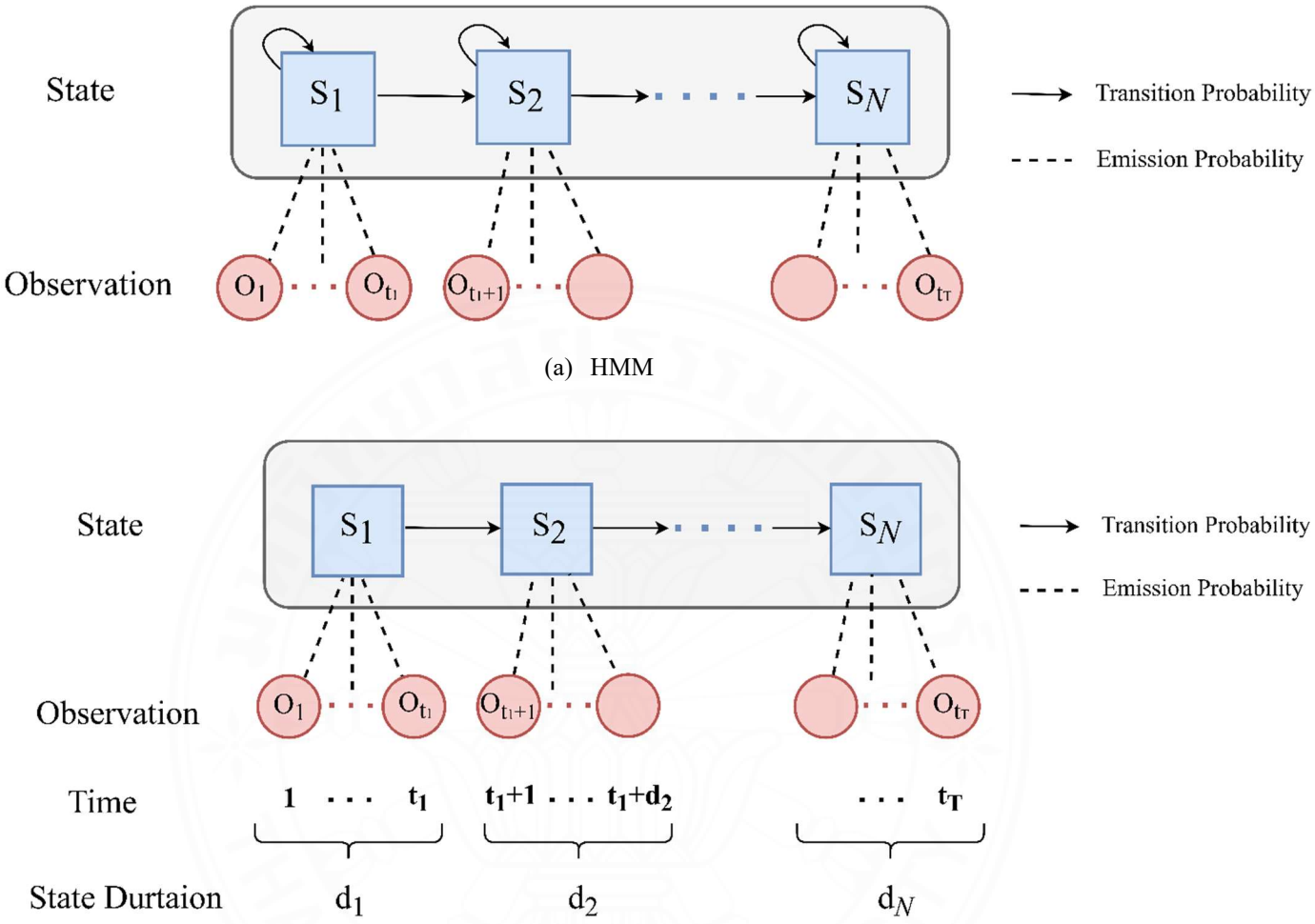
Various studies have been conducted to resolve the issue of RSSI fluctuation in object-tracking applications. Two popular methods for indoor localization are trilateration and fingerprinting, which utilize RSSI values obtained from multiple BLE scanners to estimate the 2D coordinates of a BLE tag [10]. 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 [10]. However, it may have limitations when multiple BLE scanners give conflicting results.

Recent studies have shown that the Hidden Markov Model (HMM) has the potential to explain indoor trajectory using RSSI data. For example, Han, Rho, & Lim [12] used the BLE signal from the smartwatch and implement an HMM algorithm to determine the user's indoor location and most likely sequence based on previous RSSI data. Similarly, Arslan, Cruz, and Ginhac [13] utilized real-time BLE RSSI data from the construction site and HMM to identify semantic trajectories for extracting worker movement insights to improve safety. However, conventional HMM approaches do not specify the dwell time of the object in each state, which is highly unlikely in practical scenarios [14-15]. The Hidden Semi-Markov Model (HSMM) has been proposed as an extension of the conventional HMM, which includes modeling the duration of each state. HSMM has been successfully applied in indoor tracking [14-16] and various other domains, including Human-Robot Collaboration (HRC) [17] in assembly scenarios and Machine Condition Recognition [18]. Despite these advancements, there is limited literature specifically focusing on understanding and predicting temporal characteristics in production settings.

### *2.2. Hidden Semi-Markov Model (HSMM)*

The Hidden Markov Model (HMM) [12-14] 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, hidden states of the process, which are not directly observable. The second layer represents the emission states, which are related to the hidden states but can be observed or measured. The concept of HMM comprises five components, including state ($S$), initial state probability ($\pi$), 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 $S \in \{1, 2, ..., N\}$. The transition probability matrix $A$ represents the probability of moving from one hidden state to another. These transition probabilities are

utilized to calculate the probability of a sequence of states given a sequence of observations. The HMM uses the Markov property, which defines that the probability of transitioning from one state to another depends only on the current state, to predict the next state based on the current state. This paper considers various stages of production, such as material preparation, processing, and finishing as the set of hidden states $S$, since RSSI information being used as observation $O$ may not imply the exact location of a device.




(a) HMM

(b) HSMM

Fig. 3 Graphical representation of HMM and HSMM.

The transition probability matrix $A$ in an HMM is time-independent, which is not suitable for modeling temporal behaviors. The Hidden Semi-Markov Model (HSMM) [17-20] addresses this issue by introducing the variable duration $d$, which defines how long to be in each state. Fig. 3 illustrates the two primary characteristics of an HSMM that distinguish it from an HMM [16]:

(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 amount of 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 do.

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

$$p_i(d) = Pr\left(d \middle| S_t = i\right) \tag{1}$$

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 $i$ [9]. There are three basic problems with the HSMM framework used in academic works [9, 16]:

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

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

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

This paper focuses on the learning problem of determining the probability matrix for state durations.

## 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 being manufactured. 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 $\pi$ 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 equation for calculating the cycle time is as follows:

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

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. Then, the throughput time can be represented as:

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

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 [9].

(1) The working areas (stations) are represented as the hidden states $S \in \{1, 2, ..., N\}$. 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 $O_t = [RSSI_{1,t}, RSSI_{2,t}, ..., RSSI_{N,t}]$.

(2) The emission probabilities $B$ consisting of $b_i(O_t) = P(O_t \mid s_t = i, \mu_i, \sum_i)$ are assumed to be fixed but may be updated periodically based on the collected RSSI data [14-15]:

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

where, $\mu_i^m$ 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 [15-16]. Therefore, $b_i(O_{t1:t2})$ can be expressed as the product of the probabilities of observing each RSSI data point from epoch $t_1$ to $t_2$, i.e.,

$$b_i(O_{t_1:t_2}) = \prod_{t=t_1}^{t_2} b_i(O_t) \tag{5}$$

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

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

(4) The initial state probability $\pi_i$ 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. The forward variable $\alpha_t(i)$ in HSMM is the likelihood of being in the state $i$ given $O_{1:t}$ as partial observations up to time $t$, for a given HSMM parameter $\lambda$ [16].

$$\alpha_t(i) = P(O_{1:t}, S_i \textit{ ends at } t \mid \lambda) = \sum_{d=1}^{min(t,D)} \alpha_{t-d}^*(i)\, p_i(d)\, b_i(O_{t-d+1:t}) \tag{7}$$

$$\alpha_t^*(i) = P(O_{1:t}, S_i \textit{ begins at } t+1 \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)\, a_{j,i} \tag{8}$$

where, $t$ =1, …, $T$, $i$ =1, …, $N$, $\alpha_t^*(i)$ 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 $\lambda$. The proximity likelihood of a location is calculated using the formula $\widehat{S_t} = arg \max_{1<i<N} \alpha_t(i)$. The HSMM backward variable $\beta_t^*(i)$ is defined as follows [16]:

$$\beta_t(i) = P(O_{t:T} \mid S_i \textit{ begins at } t, \lambda) = \sum_{d=1}^{(T-t,D)} \beta_{t+d}^*(i)\, p_i(d)\, b_i(O_{t:t+d-1}) \tag{9}$$

$$\beta_t^*(i) = P(O_{t:T} \mid S_i \textit{ ends at } t-1, \lambda) = \sum_{j=1}^{N} a_{i,j}\, \beta_t(j) \tag{10}$$

where, $t = T, \ldots, 1$, $i = 1, \ldots, N$, $\beta_t^*(i)$ 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 $\lambda$. Then, the duration probability $\hat{p}_i(d)$ matrix is re-estimated based on the forward and backward variables as [16]:

$$\hat{p}_i(d) = \frac{\sum_{t=1}^{T-d+1} \alpha_{t-1}^*(i)\, p_i(d)\, b_i(O_{t:t+d-1})\, \beta_{t+d}^*(i)}{\sum_{d=1}^{D} \sum_{t=1}^{T-d+1} \alpha_{t-1}^*(i)\, p_i(d)\, b_i(O_{t:t+d-1})\, \beta_{t+d}^*(i)} \tag{11}$$

The outline of the learning problem used in the paper is as follows [9].

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

**Input:** $A, B$, coupled sequences $(S^k, O^k)$; $k = 1,2,...,K$

**Training:**

Set the values of $p_i(d)$ with uniform distribution.

Initialize the $\widehat{P}_i(d)$ matrix.

**Loop** for $k = 1,...,K$

  **Loop** for $t = 1,...,T$

    **Calculate** $\alpha_t(i), \alpha_t^*(t), \beta_t(i), \beta_t^*(t)$

    **Compute** the $\widehat{p}_i(d)$

    $\widehat{P}_i(d) = \widehat{P}_i(d) + \widehat{p}_i(d)$

  $p_i(d) = (\widehat{P}_i(d)) / T$

**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 $i$ govern the cycle time for that specific stage of the production process. It is important to note that the $\hat{p}_i(d)$ Eq (11) 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.

### *4.1. Numerical experiment*

In this experiment, 1000 state/observation sequences were based on given HSMM parameters ($\pi$, $A$, $B$, $P$). 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.

$$P = \begin{bmatrix} 0 & 0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0.1 & 0.3 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.7 & 0.1 & 0.2 \\ 0 & 0 & 0.3 & 0.4 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.2 & 0.2 & 0.2 & 0 \\ 0 & 0.4 & 0.4 & 0.2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12)$$

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

$$\mu = \begin{bmatrix} -50 & -75 & -100 & -100 & -100 & -100 & -100 \\ -78 & -55 & -80 & -100 & -100 & -100 & -100 \\ -95 & -78 & -56 & -87 & -100 & -100 & -100 \\ -100 & -100 & -70 & -50 & -79 & -100 & -100 \\ -100 & -100 & -100 & -78 & -54 & -80 & -100 \\ -100 & -100 & -100 & -100 & -79 & -50 & -80 \\ -100 & -100 & -100 & -100 & -100 & -100 & -53 \end{bmatrix}, \quad \sigma = \begin{bmatrix} 5 & 2 & 1 & 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 2 & 1 & 1 & 1 \\ 1 & 1 & 3 & 5 & 2 & 1 & 1 \\ 1 & 1 & 1 & 2 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 4 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 3 \end{bmatrix} \tag{13}$$

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 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 state using 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 allows 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.

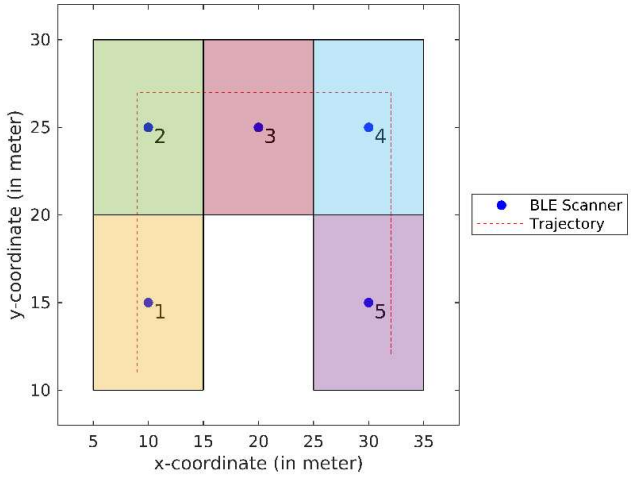



Fig. 4 Layout with BLE 2-D scanner coordinates

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

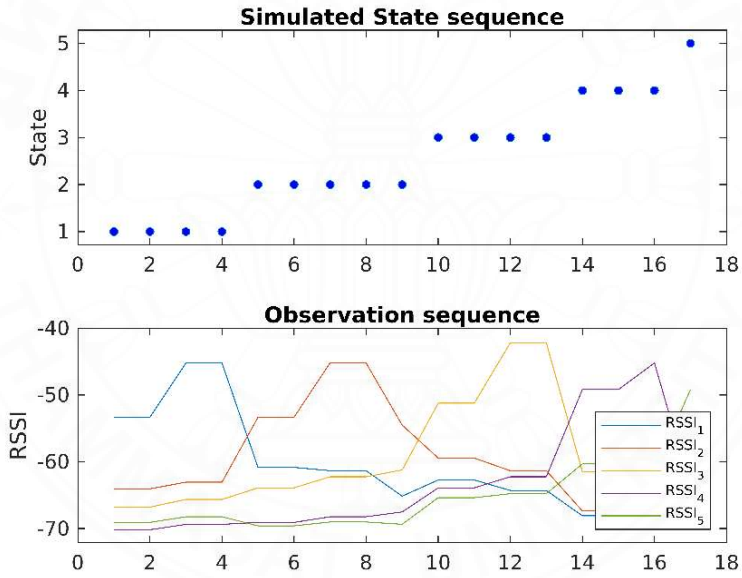



Fig. 5 Example of simulated data.

*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 programmed as BLE scanners that connect with the WiFi access point and report ID/RSSI/timestamp to the Firebase database.

(2) WeMos R32 boards 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.

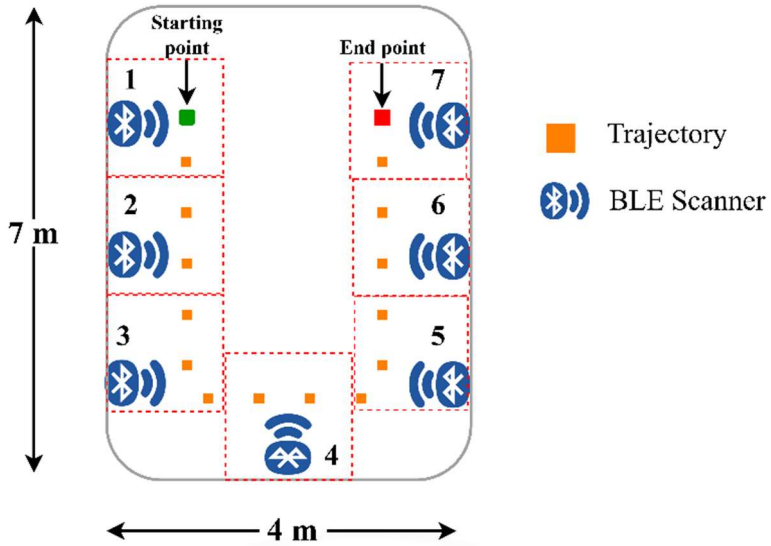



Fig. 6 BLE scanner allocation in real-time experiment.

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.

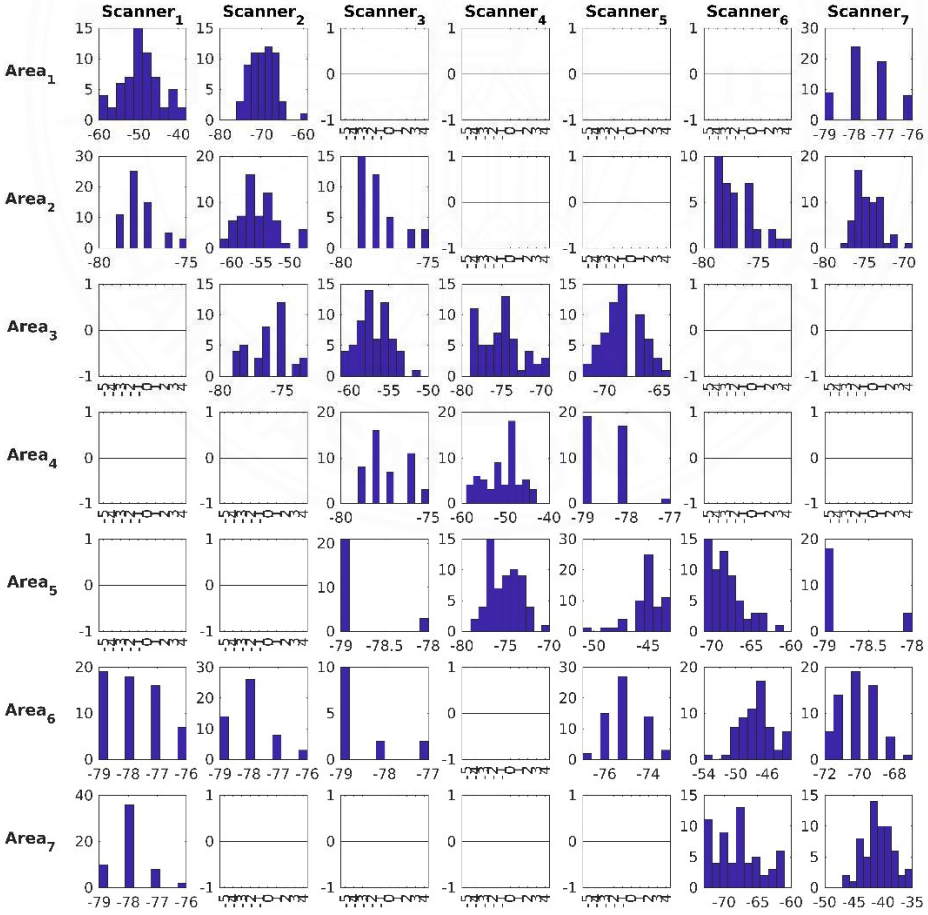



Fig. 7 RSSI measurements in real-world environments.

# 5. Discussion

## *5.1. Analysis of numerical experiment*

In this study, the Kullback-Leibler Divergence (KLD) [16] is utilized as the distance metric to determine the similarity between the original $p$ and the re-estimated $\hat{p}$.

$$D_{KL}\left(p_i(d) \left\| \widehat{p_i}(d)\right.\right) = \sum_{d=1}^{D} p_i(d) \ ln \frac{p_i(d)}{\widehat{p_i}(d)} \tag{14}$$

The average KLD value for the entire system is approximately 0.1843. This value is close to zero, indicating that the estimated $\hat{p}_i(d)$ 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.

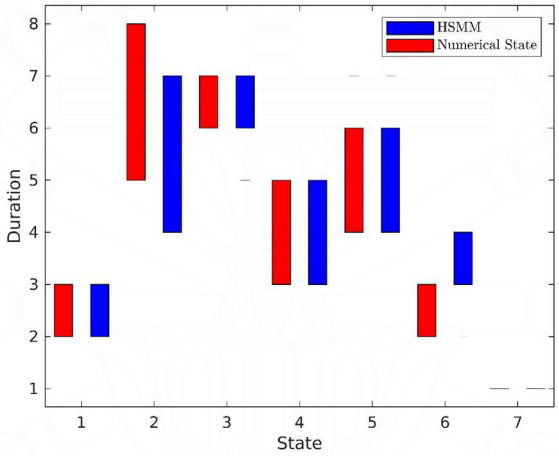



Fig. 8 Predicted vs Actual Duration Comparison.

Fig. 9 shows the trend of the average KLD values with respect to the number of coupled sequences used in the training phase. This demonstrates that the proposed algorithm converges quickly with a reasonable amount of data. The decrement of the KLD value from 1.7281 to 0.1843 as the number of samples increased from 100 to 1000 provides strong evidence that Algorithm 1 is a reliable and accurate method for estimating the duration probability.

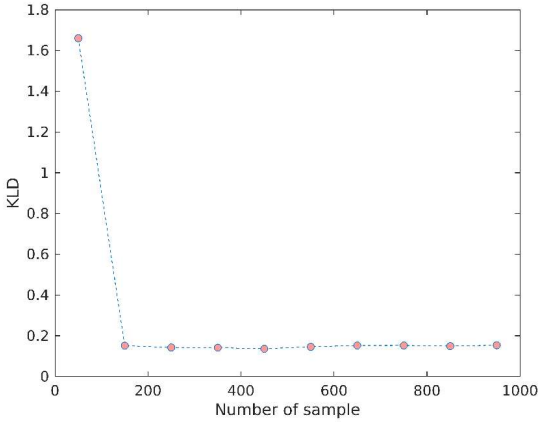



Fig. 9 Trend of average KLD value in numerical experiment.

### 5.2. Analysis of 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 favorable because it shows that the estimated throughput time is close to the simulated sequences. The study was then continued by running simulations and estimations for another 500 state sequences as part of the forecast setting.

The similarity between the forecasted and simulated sequences is evaluated using 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 is “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 using the equation $\sqrt{(X-Y)^2}$, 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 would be $\sqrt{((3-3)^2+(2-2)^2+(4-3)^2)} = \sqrt{0+0+1^2} = 1$. The lower the vector distance, the greater the similarity between the two sequences. A value of zero indicates that the 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, indicating 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.

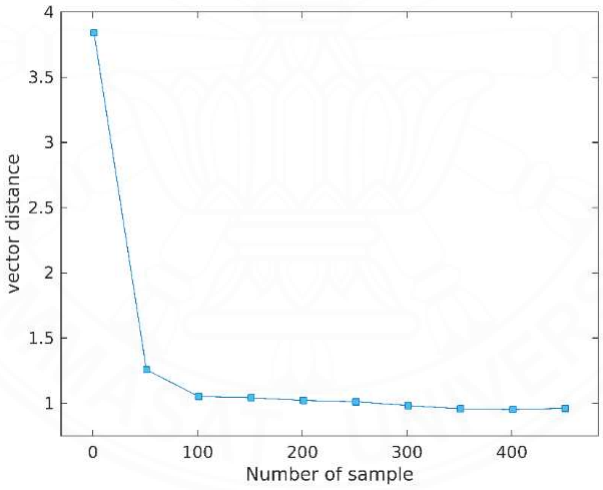



Fig. 10 Vector distance for simulated and forecasted sequences.

### 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 then filtered based on tag ID to retain only the RSSI values associated with the scanner number. The filtered data was then 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.

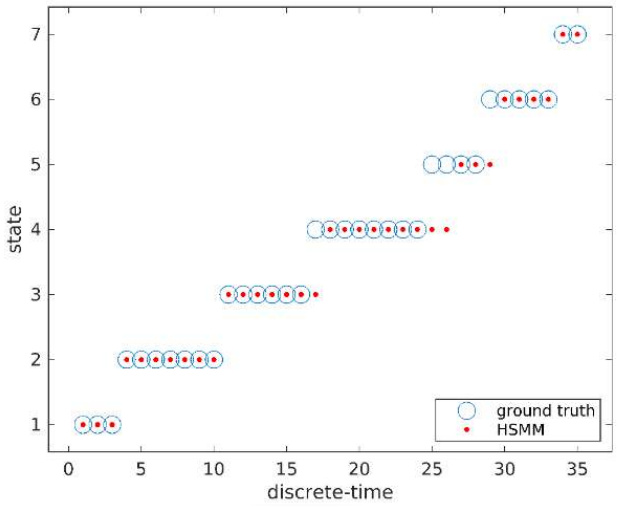



Fig. 11 Hidden state alignment example.

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. This 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

The main objective of this study is to uncover the temporal behavior and identify potential correlations in the HSMM using RSSI data from the production line, which previous research has been unable to achieve. The proposed method updates duration probabilities and utilizes forward and backward procedures to determine the likelihood of observed sequences in an HSMM, and then employs learning procedures to update the duration parameter.

To evaluate the performance of the HSMM method, three experiments were conducted, demonstrating reasonable accuracy. The numerical experiment resulted in a comparison of 28.51 epochs versus 27.69 epochs, the simulation experiment showed 20.53 epochs versus 19.89 epochs, and the real-world experiment yielded 33.99 epochs versus 30.48 epochs. These results show that the actual and predicted values are quite close to each other, indicating the effectiveness of the HSMM model in predicting 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 using RSSI and timestamp data for predicting important 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, pp. 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] F. J. Ramirez, 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, October 2022.

[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, January 2019.

[5] C. M. Liu, “Lean Transformation for Composite-Material Bonding Processes,” International Journal of Engineering and Technology Innovation, vol. 2, no. 1, pp. 48-62, January 2012.

[6] 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.

[7] K. Marso and D. Macko, “A New Parking-Space Detection System Using Prototyping Devices and Bluetooth Low Energy Communication,” International Journal of Engineering and Technology Innovation, vol. 9, no. 2, pp. 108-118, April 2019.

[8] 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, pp. 308, January 2022.

[9] S. Vorapojpisut and K. Agrawal, “Modeling of Manufacturing Processes using Hidden Semi-Markov Model and RSSI data,” 2022 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), November 2022.

[10] T. Kluge, C. Groba, and T. Springer, “Trilateration, Fingerprinting, and Centroid: Taking Indoor Positioning with Bluetooth LE to the Wild,” World of Wireless, Mobile and Multimedia Networks, August 2020.

[11] C. R. Pratiwi, P. Kristalina, and A. Sudarsono, “A Performance Evaluation of Modified Weighted Pathloss Scenario Based on the Cluster Based-PLE for an Indoor Positioning of Wireless Sensor Network,” International Journal of Engineering and Technology Innovation, vol. 9, no. 1, pp. 61-74, January 2019.

[12] D. Han, H. Rho, and S. Lim, "HMM-Based Indoor Localization Using Smart Watches' BLE Signals," Conference on the Future of the Internet, August 2018.
[13] M. Arslan, C. Cruz, and D. Ginhac, "Semantic Trajectory Insights for Worker Safety in Dynamic Environments," Automation in Construction, vol. 106, pp. 102854, October 2019.
[14] 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.
[15] S. Sun, X. Wang, B. Moran, A. Ai-Hourani, and W. S. T. Rowe, "Radio Source Localization Using Received Signal Strength in a Multipath Environment," International Conference on Information Fusion, July 2019.
[16] 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, p. 107230, January 2020.
[17] C. J. 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.
[18] W. Yang and L.-H. Chen, "Machine condition recognition via hidden semi-Markov model," *Computers & Industrial Engineering*, vol. 158, p. 107430, August 2021.
[19] 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.
[20] K. Li 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, pp. 117758, June 2022.

**Original Article**

# Production flow modeling based on BLE-based RSSI data with non-detectable areas

Karishma Agrawal[1*] and Supachai Vorapojpisut [1]

Department of Electrical and Computer, Faculty of Engineering, Thammasat University, Rangsit, 12120, Thailand

*** Corresponding author, Email address: karishma.agra@dome.tu.ac.th**

## Abstract

This study presents a method for modeling manufacturing processes to predict key performance indicators (KPIs) such as cycle time using Bluetooth Low Energy (BLE) data. We consider BLE applications to be similar to Radio-Frequency IDentification (RFID) scenarios, with a single BLE scanner indicating a single working area. This work considers when Received Signal Strength Indicator (RSSI) data is unavailable in some areas, such as, when products are in temporary storage areas away from the production areas. We solve this problem with a Duration and Interval Hidden Markov Model (DI-HMM), in which time spent in production areas is represented as duration and those with absence data as intervals. To parameterize the DI-HMM model, we propose a two-stage machine-learning problem based on a classification tree and a Hidden Semi Markov Model (HSMM). To investigate the proposed model, the RSSI observation sequences are generated using MATLAB Bluetooth Toolbox and real-world experimentation. The runtime scenario compares estimated and original states, and the average accuracy of 100 test sequences is around 95%. In the offline forecast scenario, an estimated DI-HMM parameter is used to forecast 200 sequences, then compared with sequences with a vector distance with a similarity score of 0.4717.

Ref. code: 25656210030018FRH

**Keywords:** Manufacturing process, Bluetooth Low Energy, Duration and Interval Hidden Markov Model, Classification Tree, Hidden Semi Markov Model

## 1. Introduction

A factory is a combination of various buildings that host several manufacturing processes to produce final products to be stored and sent to customers. Generally, the factory area has two sections based on their operations: production areas and warehouse areas. The production areas are where a sequence of works is performed to manufacture final products, and the warehouse areas store these products until customers demand them. There are usually small storage areas within the production area to temporarily keep intermediate parts among working areas. Product tracking data plays a crucial role to provide visibility of productivity in manufacturing areas such as the amount of work-in-process, storage capacity, and machine status.

Production flow refers to how raw materials move in sequential order from one working area to another to get final products. Each working area will complete its specific task and move the finished parts to the next working area. If the next working area is occupied with some unfinished work, then the parts will be shifted into a temporary storage area. Several manufacturing KPIs are being defined based on production flow information including cycle time, machine utilization, and production throughput. The behavior of production flow can be understood by the tracking of changes from raw material to parts and then products in production areas using barcode and RFID systems.

A Radio-Frequency IDentification (RFID) device is the most widely used device for tracking objects in malls, hospitals, factories, and other buildings (Zhu, Mukhopadhyay, &

Kurata, 2012). An RFID system consists of two parts: an RFID reader and tags. RFID tags are attached to moving objects, while RFID readers are placed in a specific location to detect tag entries. The tags communicate via radio signal with readers as they pass by. The main drawback of RFID is that it can only determine entry time, whereas knowing when products enter and leave a production flow is essential for predicting their temporal behavior. Bluetooth Low-Energy (BLE) devices are widely used for indoor detection applications (Subedi & Pyun, 2020). In comparison to other solutions, BLE technology has several characteristics that satisfy the requirements of production tracking, such as a suitable range (10 m), a reasonable price tag (<$30), and a long battery life (>1 year). BLE object detection consists of two types of devices: BLE tags that broadcast their identities and BLE scanners installed at particular locations that look for nearby BLE tags. Proximity detection uses Received Signal Strength Indicator (RSSI) signals to identify nearby BLE devices and collect data such as RSSI, product ID, and timestamps for analyzing entry and exit times (Narzt et al., 2016).

Numerous studies have been proposed for the modeling of the temporal behaviors of production flow. Queuing theory is the most popular and traditional approach for estimating the temporal behaviors of production flow, such as cycle time and throughput time. The basic definition of queueing is that the production task arrives, waits for service, and then exits after finishing the process. However, these works (Gao et al., 2019) usually consider the timing for each station separately. An RFID-enabled graphical deduction model (rfid-GDM) (Ding, Jiang, Sun, & Wang, 2017) was proposed to capture the time-sensitive and other aspects of RFID-tagged products in the production flow. Their approach allows the decomposition of events from fixed and moveable RFID readers into a sequence of states. But the rfid-GDM concept does not aggregate RFID data from multiple production sequences into statistical parameters. This limits its application in the manufacturing process with respective operations. The unsupervised measurement (Nakai, Maekawa, & Namioka, 2016) studied, wearable sensor data

arranged as segments or "motifs" and figures out cycle time based on the motif's repetition intervals. However, the unsupervised measurement considers only the cycle time of one station, while neglecting the temporal characteristics of the whole production line. The use of BLE and HMM (Subedi & Pyun, 2020) was used to define trajectories, and another work that combined BLE and HMM (Arslan, Cruz, & Ginhac, 2019) explained semantic trajectories to better understand worker mobility and improve safety. But these works ignore scenarios with the absence of BLE data due to the limited range of BLE communication. To overcome limitations related to the BLE detection range, we consider the HMM mathematical framework since the model can represent both the statistical behavior of RSSI values as observations and the sequence of production flow as state transitions.

The main objective of this paper is to develop a mathematical model for predicting temporal metrics of the production flow using the HMM concept. We consider BLE applications similar to RFID scenarios where a single BLE scanner is installed in a single working area. Three core ideas make our works different from others:

1) Our scenario is motivated by the installation of barcode/RFID in production lines, so the BLE signal may be detected by one scanner of the occupied area, many (closed by areas), or none (no closing area).

2) The conversion of RSSI values into the proximity of areas is to be aggregated as a stochastic model of sequence along the production line.

3) We use a two-stage learning problem to parameterize the model.

The paper is structured as follows. Section 2 reviews the concepts and technologies used. Section 3 defines the tracking problem as the DI-HMM model and the algorithm used to solve it. Section 4 contains the numerical results. The conclusion of the paper is discussed in Section 5.

## 2. Concept and Technologies

### 2.1. BLE based Radio Frequency IDentification (RFID) technology

RFID has two components: reader and tag. There are two types of RFID tags: passive RFID and active RFID (Zhu, Mukhopadhyay, & Kurata, 2012). Mostly, passive RFID is used in industries. RFID tags obtain power from the reader therefore, the reader can receive a signal from that tag only when that tag reaches the reader. The RFID reader tends to be bulky in size and covers less distance. BLE is a related technology that uses a similar concept to RFID but with a battery-powered tag. We use an RFID-like scenario where a single BLE scanner identifies a particular working area.

BLE is a low-power and short-range wireless technology that can be utilized for indoor object detection. RSSI reflects the distance between tags and scanning devices in dBm (Subedi & Pyun, 2020). Theoretically, distance and RSSI have an inverse relationship, but the RSSI fluctuates even when the location is the same. In our paper, we used the Wemos D1 R32 which is an ESP32 board with a PCB antenna. The initial study employed two boards, one as a BLE tag (broadcasting mode) and the other as a BLE scanner. By positioning the BLE tag at various distances and gathering RSSI data. Figure 1 shows the relationship between distance and RSSI.

Fig. 1 RSSI vs Distance of BLE devices

Figure 1 depicts the RSSI variation for each distance. Three fundamental issues with RSSI-based systems are signal strength attenuation, signal interference, and multipath propagation (Subedi & Pyun, 2020). Trilateration and fingerprinting (Jondhale et al., 2022) are popular approaches that use RSSI values to detect the position of objects. Both approaches assume there are three or more BLE scanners that read RSSI values from the same BLE tag to find the object in 2D coordinates. However, BLE localization methods are ineffective in our

study because they require at least two or more BLE scanners to find the coordinates and match the closest suitable area. Additionally, it will not work when no BLE data is available. That is why we propose the HMM-based method to compensate for such issues with the information of states in the production sequence.

### 2.2. Hidden Markov Model (HMM)

HMM is a double-layer stochastic process with some specific hidden states based on the Markov process that relies upon observation states (Rabiner, 1989). Based on the HMM concept, the production sequence and transition in the manufacturing area can be determined from an observation sequence. HMM consists of 5-elements: State ($S$), Initial State probability ($\pi$), Transition Matrix ($A$), Observation State ($O$), and Emission Matrix ($B$). The following three conditions are required for the realization of HMM:

1) **The Markov assumption:** The HMM's next state depends just on the current state.
2) **Independent assumption:** The emission probability of the current observation depends only on the current state, consequently independent from other states and another observation.
3) **The Stationarity assumption:** State transition probabilities do not consider an actual time at which the transformation of the state performs

Manufacturing is a complex process where products go through different working areas and operations, including picking, assembly, and forwarding. Cycle time is a critical KPI in the manufacturing industry because it describes temporal characteristics of working areas that can be used in the evaluation and forecasting of production efficiency. In the production process, production time is a fundamental aspect of the transition. To resolve this problem, the Hidden Semi Markov Model (HSMM) introduces the variable duration ($D$) that relies on each state

and controls the transition matrix ($A$) (Sun, Wang, Moran & Rowe, 2020). Model accuracy increases if the system depends on time in comparison with HMM. HSMM has two main properties that make it different from HMM. First, the self-transition probability is assumed zero, i.e., no return to the same state, and a single state depends upon the sojourn duration (Sun, Wang, Moran & Rowe, 2020).

## 3. Problem statement

This study used a BLE-based system to track and collect product movement information to estimate the total cycle time of products. A BLE-based system uses BLE tags being attached to products, and BLE scanners being strategically placed in working areas to detect BLE tags. However, the coverage area of the BLE scanner in the working area is limited. As a result, when the product is between two working areas, the BLE scanner may not be able to receive precise RSSI data, and the product location becomes unknown. Collecting product ID and RSSI data is the first step in understanding the production flow and its temporal metrics. Then, using the RSSI data sequence, the product's proximity location and production flow are estimated using the DI-HMM forward-backward algorithm. The other parameters, such as the overall cycle time and the cycle time for each area, are predicted using the learning algorithm.

Our previous study (Vorapojpisut & Agrawal, 2022) formulated the product tracking problem as an HSMM model $\Delta_{HSMM} = \{A, B, \pi, p\}$ that defines its state transition probability to depend on the sojourn time, which is the time for workers or machines to complete their tasks. HSMM $\Delta_{HSMM}$ consist of four mains of parameters (Sun, Wang, Moran & Rowe, 2020).

$$\Delta_{HSMM} = \{A, B, \pi, p\} \quad (1)$$

Where $A$ represents the transition matrix with duration probability $p$ for each state, $B$ signifies the emission matrix, and $\pi$ is the initial state probability.

The HSMM model can represent duration time in its context, but cannot capture the transition period between working areas. The Duration and Interval Hidden Markov Model (DI-HMM) $\Delta_{DI-HMM} = \{A, B, \pi, p, L\}$ defines the periods without observations as state intervals (Narimatsu & Kasai, 2015). In HSMM, the next state $S_j$ starts immediately after ending the present state $S_i$, but, the next state $S_j$ in DI-HMM starts after a state interval that occurs after ending its previous state $S_i$. A state interval $I_{i,j}$ between two consecutive states where $i, j \in \{1,2, \dots, N\}$ is expressed by interval length probability $P(L_{i,j})$ assumed to be the Gaussian distribution. A time length of observation of the DI-HMM is $\sum(d_i + I_{i,j})$, where $d_i \in D$ and $I_{i,i+1}$stands for the time difference between ending-time of $S_i$ and stating-time of $S_j$.

Fig. 2. Sequence of DI-HMM

The following assumptions are made to reflect manufacturing process characteristics.

1) Each working area with a BLE scanner represents the state $S_i$, where $i = 1,2,3,..N$ and $N$ denotes the number of working areas. In this case, $M$ observed (RSSI) signals are generated based on the states, therefore $M = N$.
2) The transition matrix $A$ is fixed to reflect that the manufacturing process of similar products is sequential and identical. Since all product manufacturing begins in the $S_1$ working area, the initial state probability for the $S_1$ working area is 1 and 0 for the others.
3) The emission probabilities $B$ are assumed to be fixed, but may be updated periodically based on the collected RSSI data (Sun, Wang, Moran & Rowe, 2020), as follows:

$$b_i(O_{t:t+d-1}) = \prod_{k=t}^{t+d-1} b_i(O_k) \quad (2)$$

The sequential likelihood of the HSMM denoted as $P(O|\Delta_{HSMM})$ can be determined through a forward-backward procedure. The forward-backward has two sub-divided variables: forward variable and backward variable. The HSMM forward variable (Sun, Wang, Moran & Rowe, 2020) for estimating likelihood is defined as follows:

$$\alpha_t(i) = P(\, O_{1:t},\, S_i \; ends \; at \; t \mid \Delta_{HSMM}) \; = \; \sum_{d=1}^{min\,(t,D)} \alpha^*_{t-d}(i) \; p_i(d) b_i(O_{t-d+1:t}) \qquad (3)$$

$$\alpha^*_t(i) = P(O_{1:t}\,, S_i \; begins \; at \; t+1 \mid \Delta_{HSMM}) = \sum_{i=1}^{N} \alpha_t(i) \; a_{j,i} \qquad (4)$$

$$t \; = \; 1, \ldots, T \qquad i \; = \; 1, \ldots, N$$

where, $\alpha_t(i)$ is the joint probability of obtaining the partial observation sequence up to time *t* and ending in the state *i* at the time $t$, given the model $\Delta_{HSMM}$, and $\alpha^*_t(i)$ 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 $\Delta_{HSMM}$. The proximity likelihood of a location is calculated using the formula $\widehat{S_t} = arg \max\limits_{1<i<N} \alpha_t(i)$. The HSMM backward variable (Sun, Wang, Moran & Rowe, 2020) is defined as follows:

$$\beta_t(i) = P(O_{t:T} |\, S_i \; begins \; at \; t \,, \Delta_{HSMM}) = \; \sum_{d=1}^{(T-t,D)} \beta^*_{t+d}(i) \; p_i(d) b_i(O_{t:t+d-1}) \qquad (5)$$

$$\beta^*_t(i) = P(O_{t:T} \mid S_i \; ends \; at \; t-1, \Delta_{HSMM}) = \; \sum_{j=1}^{N} a_{i,j} \; \beta_t(j) \qquad (6)$$

$$t \; = \; T, \ldots, 1 \qquad i \; = \; 1, \ldots, N$$

where $\beta_t(i)$ is the probability of the partial observation sequence from epoch *t* to the end, given that the systems start in state *i* at the time *t* and with a model $\Delta_{HSMM}$, and $\beta^*_t(i)$ 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 $\Delta_{HSMM}$. Then, the duration probability $\hat{p}_i(d)$ matrix is re-estimated based on the forward and backward variables as (Sun, Wang, Moran & Rowe, 2020):

$$\hat{p}_i(d) \; = \; \frac{\sum_{t=1}^{T-d+1} \alpha^*_{t-1}(i) \; \mathrm{p}_i(d) \; b_i(O_{t:t+d-1}) \beta^*_{t+d}(i)}{\sum_{d=1}^{D} \sum_{t=1}^{T-d+1} \alpha^*_{t-1}(i) \; \mathrm{p}_i(d) \; b_i(O_{t:t+d-1}) \beta^*_{t+d}(i)} \qquad (7)$$

The working area contains BLE scanners that collect RSSI data and determine the location of BLE tags based on strong RSSI values. While the movement and storage of products outside working areas will lead to weak and noisy RSSI values such that product location becomes unknown. To solve the problem of missing RSSI data for some tags, we propose a classification tree (Kang & Zadorozhny, 2016) to identify inside/outside scenarios based on BLE data, as shown in Figure 3.

Fig 3. Classification tree to detect inside/outside scenarios

**Input:** $O_{1:T}^{z} = \{o_1^z, o_2^z, \dots o_T^z\}$ where $Z$ is number of training sequences

$ctree$ Classification tree model

**Training phase:**

1. $p_{sum} = 0$
2. $I_{sum} = 0$
3. **for** $z$ = 1 to $Z$
4. Assign the HSMM parameters $\{\pi, A, B\}$
5. $z$ == 1 assign $p$ as uniformly distributed probability
6. **for** $t$ =1 to $T$
7. $label = predict(ctree,\ o_t)$
8. **if** $label == \{working\ area\}$
9. Calculate $\alpha_t$ , $\alpha_t^*$, $\beta_t$, and $\beta_t^*$ using (3), (4), (5), and (6)
10. **else**
11. Calculate interval $I$
12. **end**
13. **end**
14. Update parameter $\hat{p}$ using (7)
15. Calculate $L$ using (8)
16. $p_{sum} = p_{sum} + \hat{p}$
17. $p_{avg} = {p_{sum}}/{Z}$
18. $p = p_{avg}$
19. **end**

**Output:** $\hat{p}$ and $L$

Algorithm 1. Re-estimate the duration and interval probabilities using DI-HMM

Algorithm 1 is to re-estimate the duration and interval probabilities using the classification tree and HSMM. When RSSI data is classified as a positive case (working area), the training algorithm of the HSMM model $\Delta_{HSMM} = \{A, B, \pi, p\}$ will proceed normally. When RSSI data is classified as a negative case (outside the working area), the training algorithm of HSMM model $\Delta_{HSMM} = \{A, B, \pi, p\}$ is paused, and the counting of the timing steps will be used for the modeling of interval probability $L_{i,j}$.

$$L_{i,j} = p(I_{i,j}) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(I_{i,j}-\mu)^2}{2\sigma^2}} \quad (8)$$

where $i = 1,2,.N-1$ and $j = 1,2,.N$.

## 4. Numerical Results

To evaluate our proposed algorithm, we conducted three numerical studies using data from simulated and real-world scenarios. A simulated experiment generates the RSSI sequence using the left-to-right HMM concept and the MATLAB Bluetooth toolbox. Real-world data is collected in a physical setup using four BLE scanners to represent the working areas and one storage area without a BLE scanner.

### 4.1. Simulated Experiment

This study simulated end-to-end BLE transmission scenarios in the presence of a path loss model, RF impairments, and additive white Gaussian noise (AWGN). The MATLAB Bluetooth toolbox version R2022b was used to generate simulated data by providing the 2-D coordinates of the BLE scanners and moving positions for BLE tags. Algorithm 2 illustrates how to estimate the RSSI based on the scanner position with moving tags.

**Input:** Environment = Industrial, $P_t = 0\ dB$ Tx power
$Scanner_{pos} = [x_1\ y_1; x_2\ y_3; \ldots; x_N\ y_N]$ , where $N$ is number of scanners

$Tag_{Motion} = [x_1\ y_1; x_2\ y_3; \ldots; x_L\ y_L]$, where $L$ is total path steps

$Tag_{at\ scanner} = [pos_1; pos_2; \ldots; pos_N]$ ,

$Interval = [I_{12min}\ I_{12max};\ I_{23min}\ I_{23max}; \ldots; I_{(N-1)\ N_{min}}\ I_{(N-1)\ N_{max}}]$

$Duration = [d_{1_{min}}\ d_{1_{max}}; d_{2_{min}}\ d_{2_{max}}; \ldots; d_{N_{min}}\ d_{N_{max}}]$

**Sequence generation:**

1. **for** $trajectory = 1$ to $L$
2. Tag position = $Tag_{Motion}(trajectory, :)$
3. Nearest scanner = $find\ (Tag_{at\ scanner} == trajectory)$
4. **if** isempty (Nearest scanner) == 1
5. t = randomly select $Interval$ based on the two Nearest scanners
6. $S_{selected}$ = 'Storage area'
7. **else**
8. t = Select a random $Duration$ based on the Nearest scanner
9. $S_{selected}$ = Nearest scanner
10. **end**
11. $S_{time}$ (1: t) = $S_{selected}$
12. $S = [S\ ,\ S_{time}]$
13. **for** $scan = 1$ to $N$
14. Scanner position = $Scanner_{pos}\ (scan, :)$
15. Distance real = distance between the Tag position and Scanner position
16. **for** $k$ = 1 to t
17. Distance = Distance real ± randomly generated minor error
18. $PL_{dB}$ = Calculate path loss with End-to-End Bluetooth BR/EDR Simulation Procedure.
19. $RSSI = P_t - PL_{dB}$
20. $o(scan, k) = RSSI$
21. **end**
22. $O = [O\ ,\ o]$
23. **end**
24. **end**

**Output:** Hidden state $S$ and RSSI observations $O$

Algorithm 2 Generate RSSI using MATLAB Bluetooth toolbox

Fig. 4 BLE scanner position and BLE tag trajectory.

500 different sequences were generated based on the setup of the BLE scanners and BLE tag trajectory shown in Figure 4 $\boldsymbol{Duration} = [\mathbf{3,8};\ \mathbf{5,6};\ \mathbf{4,7};\ \mathbf{2,4};\ \mathbf{1,1}]$ and $\boldsymbol{Interval} = [\mathbf{3,15};\ \mathbf{5,10};\ \mathbf{5,8};\mathbf{10,15}]$ were used for algorithm 2. Each simulated sequence contains two types of information: observed (RSSI) signals and state sequences, as illustrated in Figure 5.

Fig. 5. Example of simulated sequences (states and RSSI)

We conducted the runtime estimation and the offline forecast studies. The runtime estimation compares the state sequences and estimated DI-HMM sequence. In the offline forecast scenario, the re-estimated DI-HMM is used to forecast sequences, then average the forecast and state sequences using the sample count.

#### 4.1.1. Runtime estimation scenario

In runtime estimation, 100 observed sequences are randomly selected to re-estimate the duration and interval probabilities using Algorithm 1. Then, the DI-HMM state sequences are estimated from RSSI values using the re-estimated duration and interval probabilities. Figure 6 shows the simulated state sequence as ground truth and the estimated DI-HMM state sequence.

Fig 6. Comparison of the original and estimated states

DI-HMM model accuracy is evaluated using the confusion matrix, as shown in figure 7. The confusion matrix is then used to evaluate performance, and classification $\boldsymbol{accuracy} = \boldsymbol{correct\ predictions} / \boldsymbol{total\ predictions} * \mathbf{100}$. The accuracy of this model is $(\mathbf{521} + \mathbf{561} + \mathbf{519} + \mathbf{273} + \mathbf{41} + \mathbf{4569})/\mathbf{6759}$, $\mathbf{0.9593}$ or approximately $\mathbf{95}\%$.

Fig. 7 Confusion matrix

### 4.1.2. Offline forecast scenario

In the offline forecast, different state sequences are sampled to re-estimate the duration and interval probabilities with Algorithm 1. Then, the model is used to forecast another 200 state sequences. To compare the similarity of sequences, we encode the simulated/forecasted state sequence using run-length encoding (RLE). RLE is a simple algorithm for lossless data compression that works when a simulated/forecasted state sequence contains the same value repeatedly. For example, RLE's input is "AAABBCCCC," and its output is "3A2B4C". After that, the vector distance is used to compare the multidimensional similarity of the forecasted and simulated RLE vectors. The relationship between the training sample size and its vector distance is displayed in Figure 8. The average vector distance value for the entire system is approximately 0.4717, which is close to zero and demonstrates that the estimated duration and interval are close to the original sequences used for the training of duration and interval parameters.

Fig 8. Vector distance for simulated and forecasted sequences

### 4.2. Real-World Experiment

In a real-world scenario, different stations carry out various tasks to collect RSSI data from BLE devices. The experiment consists of three components: Wemos R32 boards as BLE tags and scanners, Wi-Fi access points, and the Firebase database. Figure 2 demonstrates the BLE scanner's coverage area and the relationship between distance and RSSI. Four BLE scanners are arranged in a U shape over a 12×5m area, as shown in Figure 9.

Every 5 seconds as one timing unit, each BLE scanner scans, stores, and sends data to Firebase's real-time database. Each BLE tag was moved from one working location to the next in a predetermined order with 20 datasets collected from the area shown in figure 9.

Fig 9. Scanner location in the experiment.

Table 1 compares the average state duration and interval values between the data generated by our DI-HMM model and the data collected. This proves that our proposed method can achieve reasonable accuracy with realistic and limited datasets.

Table 1. State durations and interval (unit of 5 secs)

## 5. Conclusion

The main benefit of our study is that it can uncover temporal behavior and extract potential correlations in the DI-HMM that earlier research was unable to do using RSSI data from the production flow. With the help of a two-stage learning problem, it is possible to estimate the probability distributions $\hat{p}_i(d)$ and $L_{i,j}$ from RSSI data collected from experiments. After that,

the DI-HMM model can be used to compute the crucial KPI metric, i.e., cycle time and throughput time. The estimated cycle time of the products in the manufacturing area is the summation of the state's duration and interval time represented as $\sum_{q=1}^{N}(d_q + I_{q-1,q})$. Throughput time is the ratio of time taken to complete production to a unit of the product so, the estimated cycle time of the product is equal to the throughput time. Work-in-Process defines the current state of the product in the manufacturing area. This will benefit manufacturers by assisting them in estimating productivity on production flow. Moreover, we can forecast future target product behavior based on the most recent forecasts.

400 **References**


401 Arslan, M., Cruz, C., & Ginhac, D. (2019). Semantic trajectory insights for worker safety in
402 dynamic environments. *Automation in Construction*, *106*, 102854.
403 https://doi.org/10.1016/j.autcon.2019.102854

404 Ding, K., Jiang, P., Sun, P., & Wang, C. (2017). RFID-Enabled Physical Object Tracking in
405 Process Flow Based on an Enhanced Graphical Deduction Modeling Method. *IEEE*
406 *Transactions on Systems, Man, and Cybernetics: Systems*, *47*(11), 3006–3018.
407 https://doi.org/10.1109/tsmc.2016.2558104

408 Gao, S., Rubrico, J. I. U., Higashi, T., Kobayashi, T., Taneda, K., & Ota, J. (2019). Efficient
409 Throughput Analysis of Production Lines Based on Modular Queues. *IEEE Access*, *7*,
410 95314–95326. https://doi.org/10.1109/access.2019.2928309

411 Jondhale, S. R., Mohan, V., Sharma, B. B., Lloret, J., & Athawale, S. V. (2022). Support
412 Vector Regression for Mobile Target Localization in Indoor Environments. *Sensors*,
413 *22*(1), 358. https://doi.org/10.3390/s22010358

414 Kang, Y., & Zadorozhny, V. (2016). Process Discovery Using Classification Tree Hidden
415 Semi-Markov Model. *2016 IEEE 17th International Conference on Information*
416 *Reuse and Integration (IRI)*. https://doi.org/10.1109/iri.2016.55

417 Nakai, D., Maekawa, T., & Namioka, Y. (2016). Towards unsupervised measurement of
418 assembly work cycle time by using wearable sensor. *2016 IEEE International*
419 *Conference on Pervasive Computing and Communication Workshops (PerCom*
420 *Workshops)*. https://doi.org/10.1109/percomw.2016.7457056

421 Narimatsu, H., & Kasai, H. (2015). Duration and Interval Hidden Markov Model for
422 sequential data analysis. *2015 International Joint Conference on Neural Networks*
423 *(IJCNN)*. https://doi.org/10.1109/ijcnn.2015.7280808

Narzt, W., Mayerhofer, S., Weichselbaum, O., Haselbock, S., & Hofler, N. (2016). Bluetooth Low Energy as enabling technology for Be-In/Be-Out systems. *2016 13th IEEE Annual Consumer Communications &Amp; Networking Conference (CCNC)*. https://doi.org/10.1109/ccnc.2016.7444817

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286. https://doi.org/10.1109/5.18626

Subedi, S., & Pyun, J. Y. (2020). A Survey of Smartphone-Based Indoor Positioning System Using RF-Based Wireless Technologies. *Sensors*, *20*(24), 7230. https://doi.org/10.3390/s20247230

Sun, S., Wang, X., Moran, B., & Rowe, W. S. (2020). A hidden semi-Markov model for indoor radio source localization using received signal strength. *Signal Processing*, *166*, 107230. https://doi.org/10.1016/j.sigpro.2019.07.023

Vorapojpisut, S., & Agrawal, K. (2022). Modeling of Manufacturing Processes using Hidden Semi-Markov Model and RSSI data. *2022 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing (ISAI-NLP)*. https://doi.org/10.1109/isai-nlp56921.2022.9960270

Zhu, X., Mukhopadhyay, S. K., & Kurata, H. (2012). A review of RFID technology and its managerial applications in different industries. *Journal of Engineering and Technology Management*, *29*(1), 152–167. https://doi.org/10.1016/j.jengtecman.2011.09.011

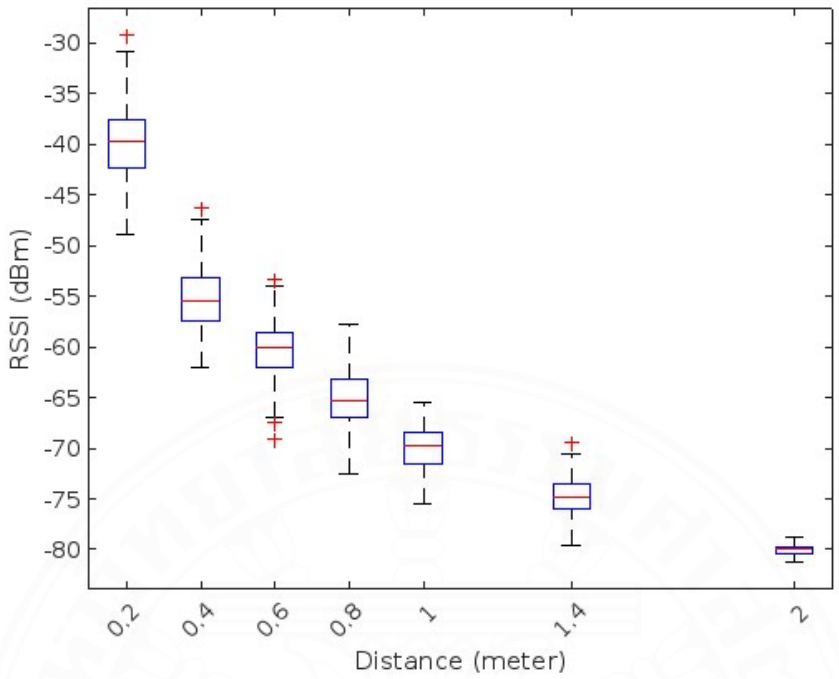



Fig. 1 RSSI vs Distance of BLE devices

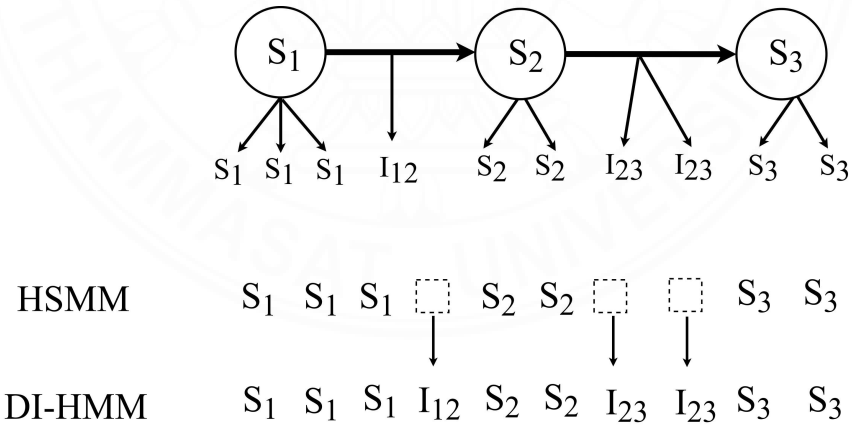



Fig. 2. Sequence of DI-HMM

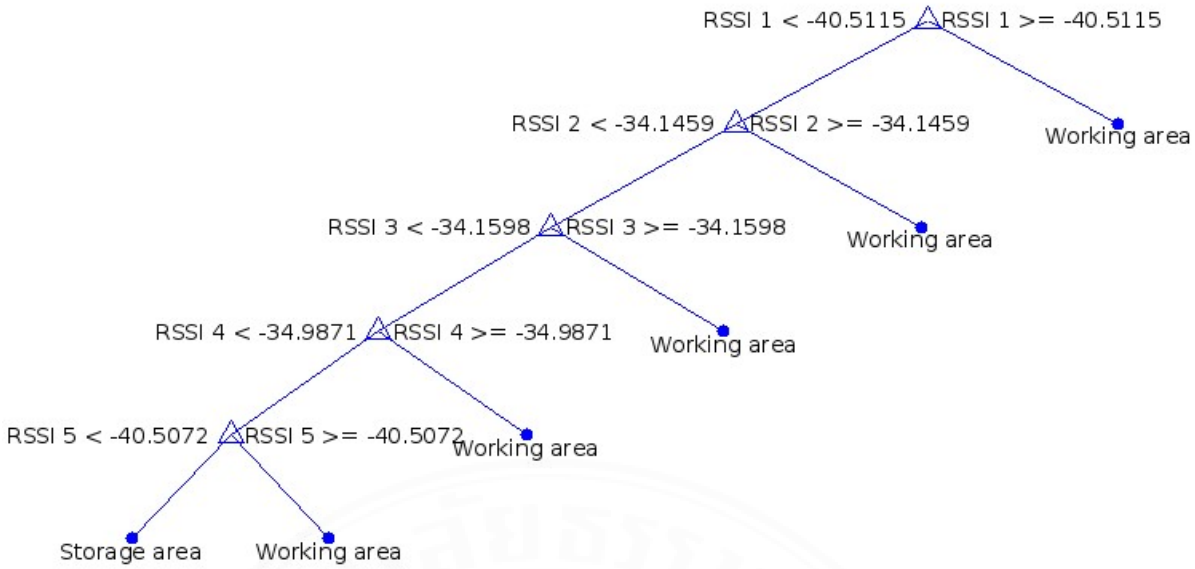



Fig 3. Classification tree to detect inside/outside scenarios

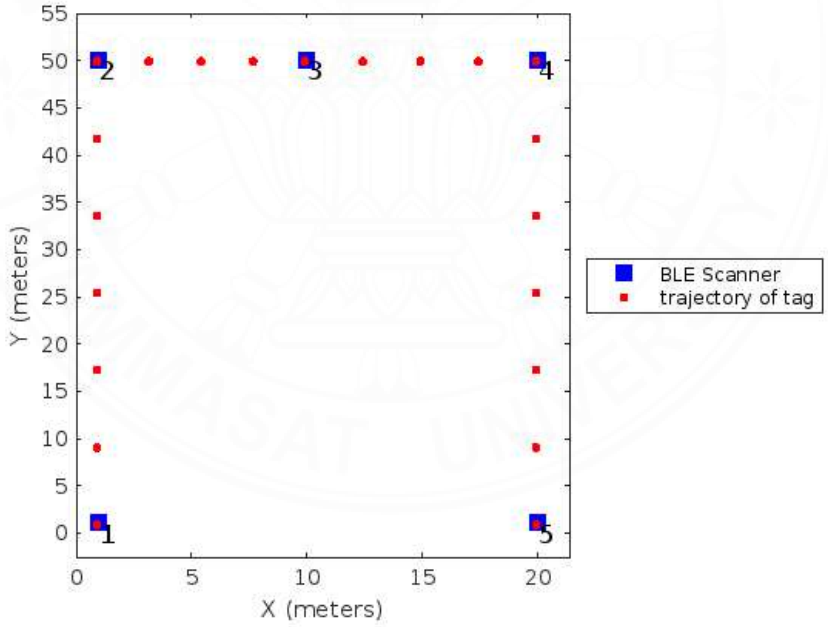



Fig. 4 BLE scanner position and BLE tag trajectory.

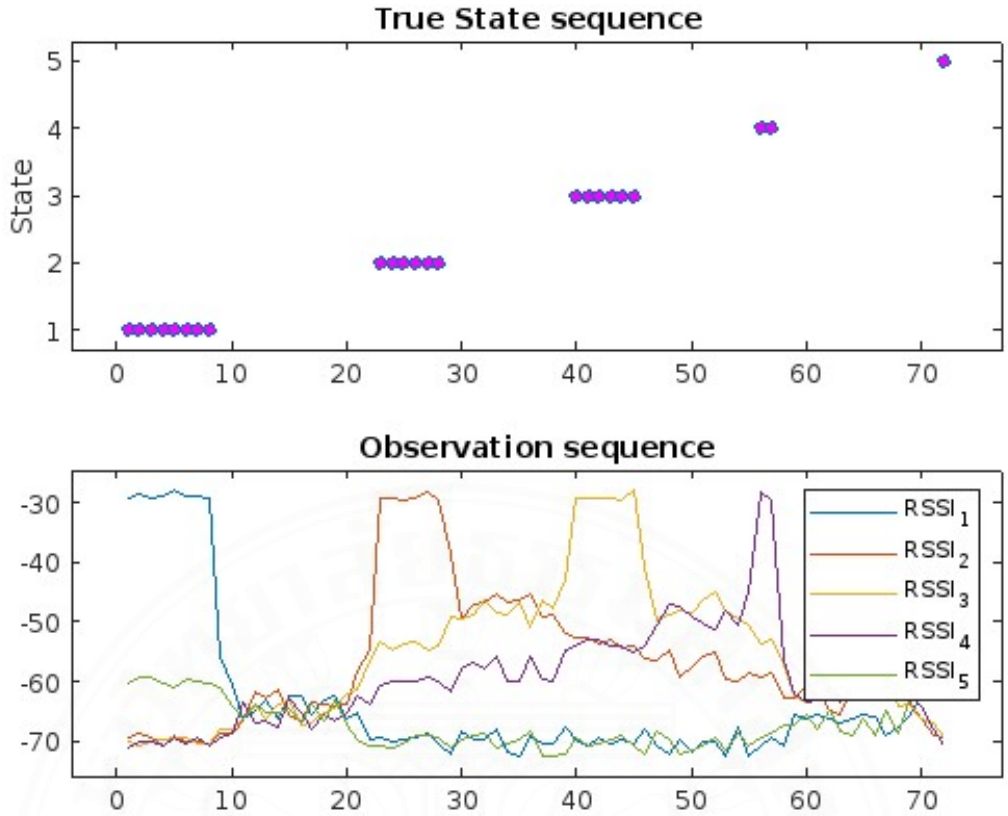



Fig. 5. Example of simulated sequences (states and RSSI)

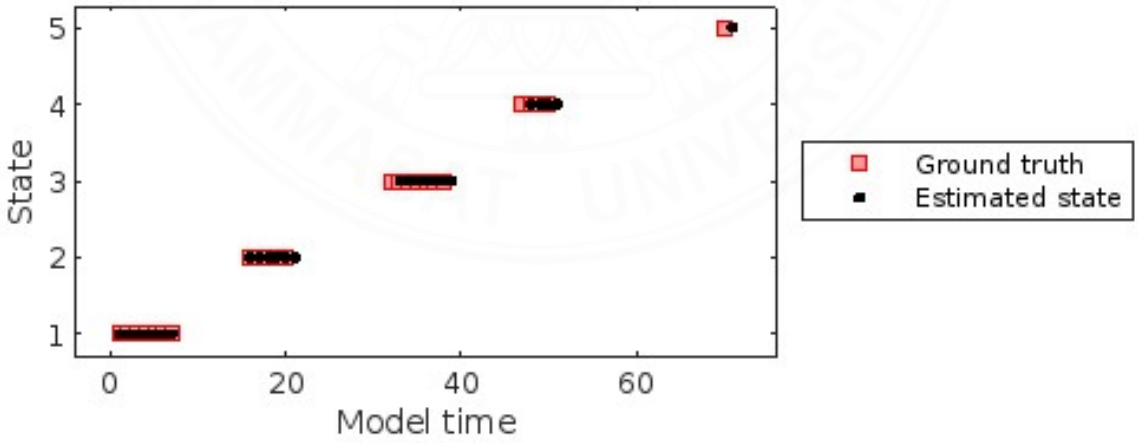



Fig 6. Comparison of the original and estimated states

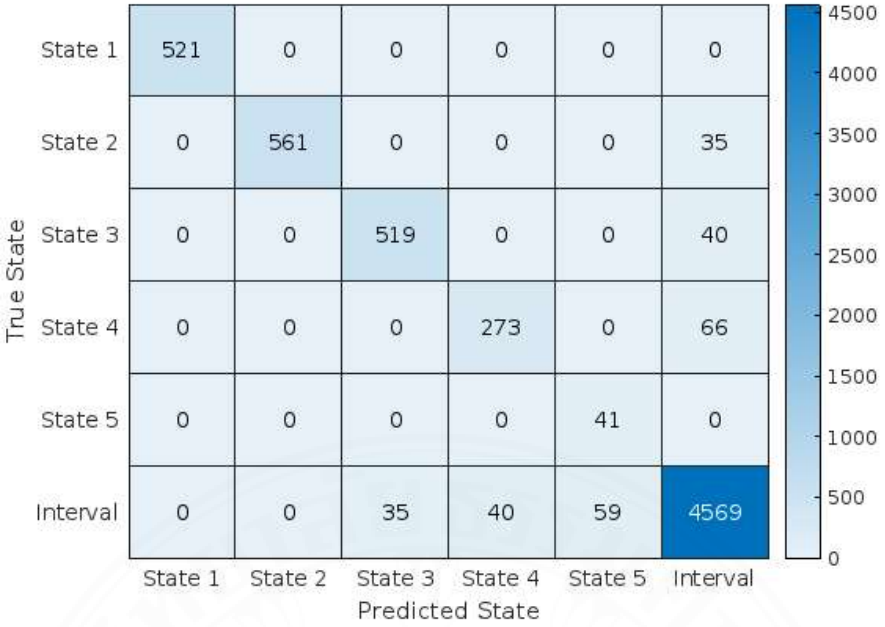



Fig. 7 Confusion matrix

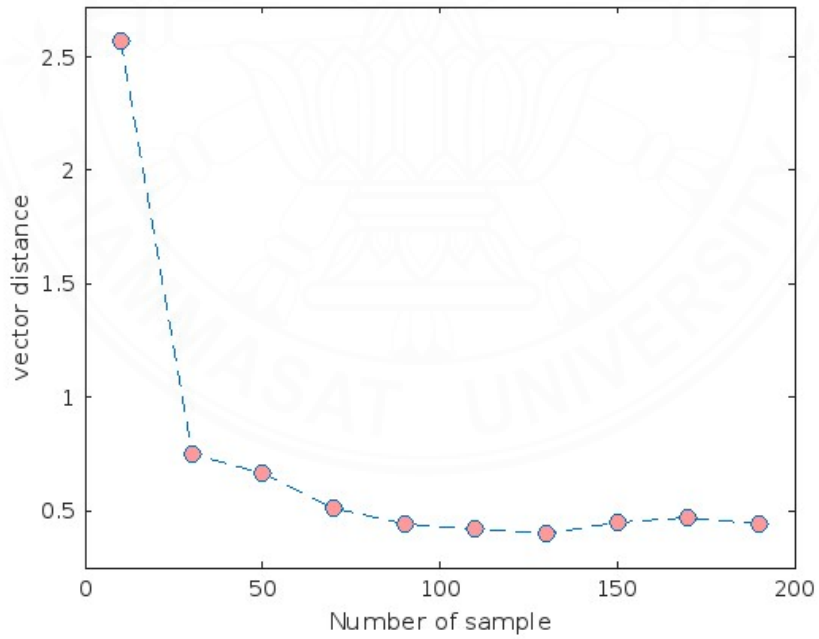



Fig 8. Vector distance for simulated and forecasted sequences

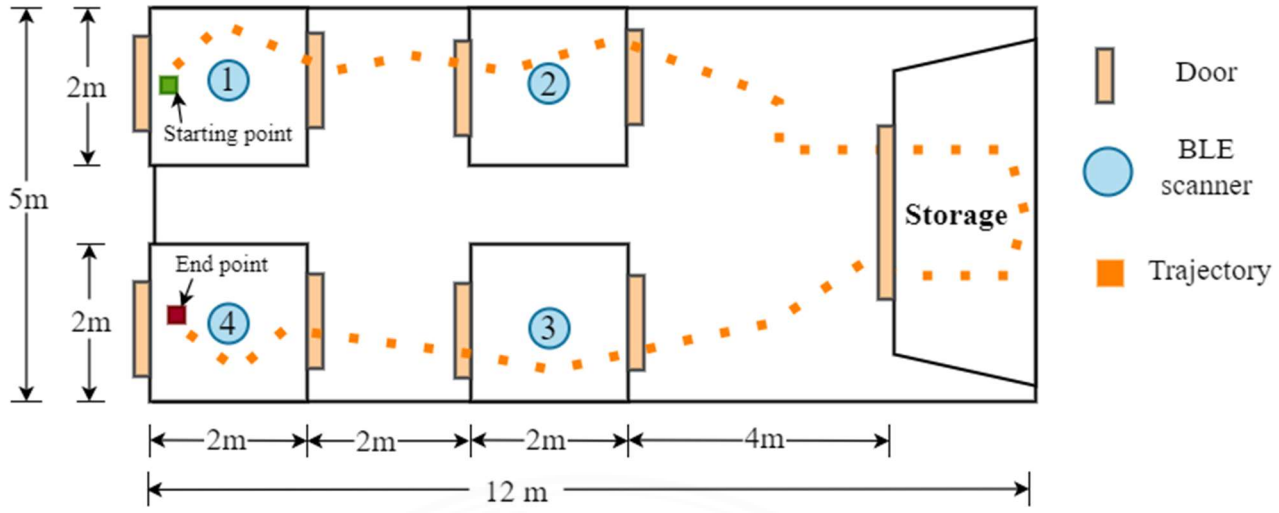



Fig 9. Scanner location in the experiment

Table 1. State durations and interval (unit of 5 secs)

| Area | #1 (Duration-1) | #2 (Duration-2) | #3 (Duration-3) | #4 (Duration-4) | Storage (Interval) | Cycle Time |
|---|---|---|---|---|---|---|
| Real | 5.50 | 5.35 | 5.70 | 3 | 18.80 | 38.35 |
| DI-HMM | 5.15 | 5.80 | 5.70 | 1 | 18.80 | 36.45 |