PROVIDING PROXIMITY SAFETY AND SPEEDING ALERTS TO WORKERS ON CONSTRUCTION SITES USING BLUETOOTH LOW ENERGY RTLS

Yusheng Huang

A Thesis in

The Department of

Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Master of Applied Sciences (in Building Engineering) at

Concordia University

Montreal, Quebec, Canada

December 2020

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Yusheng Huang

Entitled: Providing Proximity Safety and Speeding Alerts to Workers on Construction Sites Using Bluetooth Low Energy RTLS

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Building Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. SangHyeok Han

______________________________ Examiner
Dr. SangHyeok Han

______________________________ External Examiner
Dr. Farnoosh Naderkhani

______________________________ Supervisor
Dr. Amin Hammad

______________________________ Supervisor
Dr. Zhenhua Zhu

Approved by ______________________________
Dr. Ashutosh Bagchi
Chair of Building, Civil and Environmental Engineering

December,2020 ______________________________
Dr. Mourad Debbabi, Interim Dean,
Gina Cody School of Engineering and Computer Science

ABSTRACT

Providing Proximity Safety and Speeding Alerts to Workers on Construction Sites Using Bluetooth Low Energy RTLS

Yusheng Huang

The construction sector is one of the most dangerous industrial sectors. Struck-by object or equipment is one of the main causes of fatal accidents on construction sites. Although many regulations have been designed for struck-by accidents, these accidents are still causing many injuries and fatalities. According to the U.S. Bureau of Labor Statistic, the struck-by accidents has led to 112 deaths on construction site in 2018. The application of real-time location systems (RTLS) on construction sites provides new possibilities in construction safety management. Previous researchers have proposed using RTLS to track the location of workers and equipment on construction sites to improve construction safety. However, the previous methods have some limitations (e.g. cabling problems, positioning quality). Furthermore, providing effective safety alerts to workers within dangerous proximity to equipment has not been addressed in previous research. This research aims to develop a method for providing near real-time proximity alerts to workers on construction sites using Bluetooth Low Energy (BLE) RTLS based on angle of arrival (AOA). This RTLS can provide acceptable accuracy coupled with large coverage without the need of timing cables. Also, with the support of two-way communications between the tags and sensors, it is possible to provide vibro-tactile alerts to the workers using wristbands. In addition, alerts representing different cases of proximities and speeding were defined. The prototype system has the following features: (1) less cabling by using wireless technologies for data transmission, (2) less false alerts by generating the alerts to specific entities based on the micro-schedule of activities, (3) easily perceived alerts. Tests were conducted on a construction site of an electric substation to test the accuracy of the RTLS and the performance of the prototype system. The test results indicated that the prototype system is capable of detecting proximities and generating timely alerts to the involved entities.

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my supervisors, Dr. Amin Hammad and Dr. Zhenhua Zhu for their intellectual support, encouragement and patience. Their advice and criticism were my most valuable asset during my studies.

Besides, I would like to thank my teammates Miss. Chen Chen, Mr. Mohammad Akbarzadeh, my best friends Mr. Xiaoning Ren and Mr. Amr Amer, and all my colleagues for their support and help.

In addition, I want appreciate the support from Hydro-Quebec and Mitacs in this research. I also want to thank Mr. Donald Konan from Hydro-Quebec for his support in the outdoor tests.

At last, no words can express my endless appreciation to my mother and my grandparent for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to my girlfriend for her love, understanding, prayers and continuing support to my research work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AOA	Angle of Arrival
BLE	Bluetooth Low Energy
CV	Computer Vision
DEW	Dynamic Equipment Workplace
DRMS	Distance Root Mean Squared
FOV	Field of View
GC	Geometric Constraints
GND	Ground
GPS	Global Positioning System
MAS	Multi-Agent System
MDR	Missing Data Rate
MRSE	Mean Radial Spherical Error
MS	Mobile Station
OC	Operational Constraints
RFID	Radio-Frequency Identification
RSS	Received Signal Strength
RTLS	Real-Time Location System
RTS	Robotic Total Station

SD	Standard Deviation
TDOA	Time Difference of Arrival
TOA	Time of Arrival
ToF	Time of Flight
UWB	Ultra-Wideband
VBAT	Battery Voltage
WLAN	Wireless Local Area Network

Chapter 1: Introduction

1.1 General Introduction

Construction sites are one of the most dangerous workplaces in the world. In Europe and the United States, most workplace fatal accidents happen on construction sites. In 2017, more than 20% of all fatal accidents at workplace in the 28 European countries took place on construction sites (Eurostat, 2019). Similarly, in the United States, among all 4,779 workplace fatalities in 2018, 1,008 fatalities (about 21%) happened during construction activities (U.S. Bureau of Labor Statistic, 2018). In Japan, the construction industry also had the most fatalities resulting from accidents and the second most non-fatal accidents in 2019 (JISHA, 2020). Unlike most other workplaces, construction sites are dynamic. The environment and resources on construction sites are constantly changing, which makes the construction sites difficult to control and sometimes chaotic. This is one of the main reasons for the high fatality number of the construction industry.

The fatal accidents on construction sites are mainly caused by falls (33.5%), struck-by object or equipment (11.1%), electrocutions (8.5%), and caught-by equipment/material (55%), which are regarded as the 'fatal four' (Occupational Safety and Health Administration, 2020). Many regulations have been imposed to deal with these causes. However, these accidents are still causing many injuries and fatalities. In addition, the regulations are sometimes violated during construction activities. For these violated regulations, the majority of the ten most frequently cited ones in 2018 are related to the 'fatal four' (Occupational Safety and Health Administration, 2020). In addition, although regulations can mitigate risks in advance by regulating workers' activities, they cannot prevent accidents in real-time. Therefore, it is necessary to develop a method which can effectively generate alerts to the workers at risk in real-time. The applications of real-time location system (RTLS) on construction sites can help the safety management in a different way. By using the real-time location data of construction resources, the potential risks of activities, which do not comply with the regulations, can be detected (e.g., proximity to equipment, speeding vehicle/equipment on construction site, worker entering unsafe areas). Then, alerts can be generated in near real-time to the related workers. In addition, the location data of different resources on construction sites can be used for post-analysis to improve safety management. Furthermore, the same data can be very useful to analyze productivity issues (e.g., path planning and scheduling).

Previous research proposed using RTLS to track the locations of construction resources for construction management. However, it is important to find the balance between positioning quality and the cost of the RTLS. The accuracy is low for safety management on construction sites for methods using low-cost radio-frequency identification (RFID) (Chae & Yoshida, 2010; Teizer, 2015). Technologies such as ultra-wideband (UWB) and the surveying-level Global Positioning System (GPS), on the other hand, are accurate but expensive, which reduces their applicability on construction sites. Moreover, deploying sensor-based RTLS in construction projects is difficult. For example, UWB RTLS requires timing cables to synchronize the data of the sensors surrounding the site (Ruiz & Granja, 2017; Siddiqui et al., 2019), leading to higher cost and more chaos. On the other hand, previous research related to generating proximity alerts suffers from generating many false alerts, which affects the reliability of the systems(Ruff, 2007; Teizer et al., 2010). In addition, the method of generating alerts (e.g., audio, visual, or tactile alerts) has not been discussed in detail. To prevent accidents more effectively, a method that can warn workers in near real-time is needed.

1.2 Research Objectives

This thesis aims to develop a method with less cabling using Bluetooth Low Energy (BLE) RTLS based on angle of arrival (AOA) technique, which can provide sub-meter positioning accuracy with a relatively lower cost compared to UWB (Quuppa, 2019a), to provide near real-time proximity safety alerts to workers on a construction site. The research has four objectives:

(1) Testing the feasibility of using a wireless scheme of BLE RTLS on construction sites.

(2) Developing a method which effectively detects proximities in near real-time during real construction activities.

(3) Designing and testing vibro-tactile alerts sent to workers, which can be easily perceived to notify the workers about different cases of proximities effectively.

(4) Developing a prototype system, which detects proximities and generates alerts to related entities in near real-time.

1.3 Thesis Organization

This research is presented as follows:

Chapter 2 - Literature Review: This chapter reviews the literature about RTLS technologies and the applications for safety management. Furthermore, research about vibration signals is reviewed to understand which vibration signals are easier to perceive.

Chapter 3 - Providing Proximity Safety Alerts in Outdoor Construction Sites: In this chapter, the proposed methodology of the proximity safety alerts system using BLE RTLS is presented. Then, the prototype development is presented. At last, a lab test is conducted to evaluate the performance of the BLE RTLS being used in this research.

Chapter 4 - Applying BLE RTLS on Construction Site: This chapter presents the case study of applying BLE RTLS and the proposed method on a construction site of an electric substation. The setup of the RTLS is illustrated. Two tests are conducted to test the accuracy of the system in positioning. Then, a test is conducted to test the performance of the proposed system in detecting proximities on the construction site. At last, the alerts representing different seriousness of proximities are defined. Two tests and a field trial are conducted to evaluate the performance of the alert generation function

Chapter 5 - Conclusions and Future Work: This chapter summarizes the presented work considering the contributions and the limitations. This chapter also discusses future research directions.

Chapter 2: Literature Review

2.1 Introduction

In this chapter, a brief review of the RTLS technologies will be made and the previous applications of RTLS on construction sites will be reviewed. Also, the researches which discussed stimulus from wearable accessories to the human body will be reviewed. This literature review aims to evaluate the feasibility of using RTLS and wearable accessories to build a near real-time near-miss alert generating system.

2.2 Real-Time Location Systems

An RTLS is defined as a combination of hardware and software systems that can automatically detect the position of entities in real-time within its coverage (Li et al., 2016). Any technology that can position an entity in real-time is regarded as RTLS. In the systematic literature review of RTLS application on construction sites (Li et al., 2016; Soltanmohammadlou et al., 2019), the authors listed all the technologies of RTLS that have been applied in research in the past decade: Radio-frequency Identification (RFID), Goble Position System (GPS), Ultra-Wide Band (UWB), Ultrasound, Infrared, Bluetooth, Magnetic Signals, Wireless Local Area Network (WLAN) and Computer Vision (CV). Table 2-1 shows the comparison of the accuracies of different RTLS technologies applied in construction research, and the accuracy of the BLE RTLS (*Quuppa*, 2020) used in this research. RFID, UWB, CV and GPS are the most popular technologies used in construction research. RFID RTLS was proven to be helpful in resource tracking, securing construction sites and improving safety management (Chae & Yoshida, 2010; Kelm et al., 2013; H.-S. Lee et al., 2012; Skibniewski & Jang, 2009). However, most of the RFID RTLS cannot provide sub-meter accuracy and the coverage of the sensors is small. UWB RTLS, on the other hand, is emerging because of its high accuracy, long-range and immunity to weather factors (e.g., fog, rain, or clutter) (Cheng et al., 2011). However, the UWB RTLS often requires cabling between sensors, which promises the positioning performance but decreases the applicability on construction sites (Siddiqui et al., 2019). Unlike most sensor-based RTLSs, GPS is easy to be deployed because it does not require the installation of sensors on construction sites. The accuracy of GPS can vary hugely from sub-meter to above 10 m (Li et al., 2016). For surveying level GPS, the cost is high, which makes the application infeasible in construction projects. Moreover, GPS does not work in indoor environments and its performance is degraded during cloudy and rainy weather. The CV based positioning can also provide high positioning accuracy (Yang et al., 2010; Zhu et al., 2016). As another advantage, the target objects are not required to carry any device (Li et al., 2016). However, its performance can be easily affected by many factors such as occlusions. CV is not able to identify the targets' ID. Therefore, with only the CV based RTLS, it is not possible to assign alerts to the specific entities when proximity is detected. The BLE RTLS based on AOA has the potential to address the limitations of the above-mentioned positioning technologies. It can provide sub-meter accuracy with 100 m coverage in both outdoor and indoor environments without the need of timing cables (Quuppa, 2020). In addition, the BLE signal is less affected by occlusions compared with some positioning technologies such as ultrasound.

Table 2-1 Accuracy of different RTLS technologies

Technology	Accuracy
RFID	2.8 m (Skibniewski & Jang, 2009)
GPS	1.1 m (Pradhananga & Teizer, 2013)
UWB	0.30 m (C. Zhang et al., 2012)
Computer vision	0.658 m (Park & Brilakis, 2012)
Ultrasound	0.97 m (Jang & Skibniewski, 2009)
BLE	0.5 m – 1.0 m (*Quuppa*, 2020)

As shown in Figure 2.1, four main algorithms are being used in sensor-based positioning systems, which are time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and received signal strength (RSS) (Wang et al., 2013). Bensky (2016) has illustrated the four methods in detail as follows:

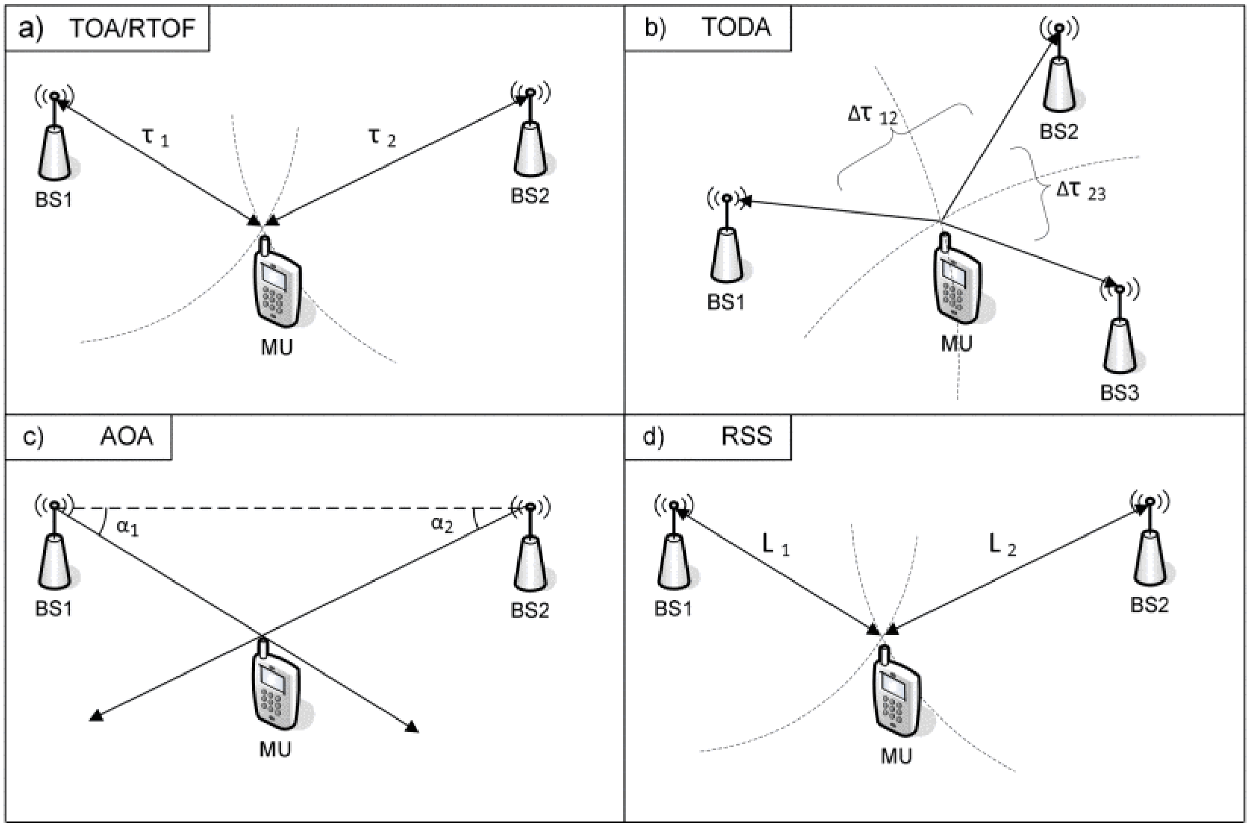


Figure 2.1 Positioning technologies (Wang et al. 2013)

(1) Time of Arrival (TOA)

TOA is based on the time of flight (ToF) of signals. It calculates the distances between the mobile station (tag) and the fixed terminals (sensors) by ToF. Then, by using the geometric relationships based on the distances, the position of the mobile station can be determined. Figure 2.2 shows an example of determining the two-dimension coordinates of the target mobile station (MS). BS1, BS2, and BS3 are the three fix terminals and d_1, d_2 and d_3 represent the distances from terminals to the mobile stations.

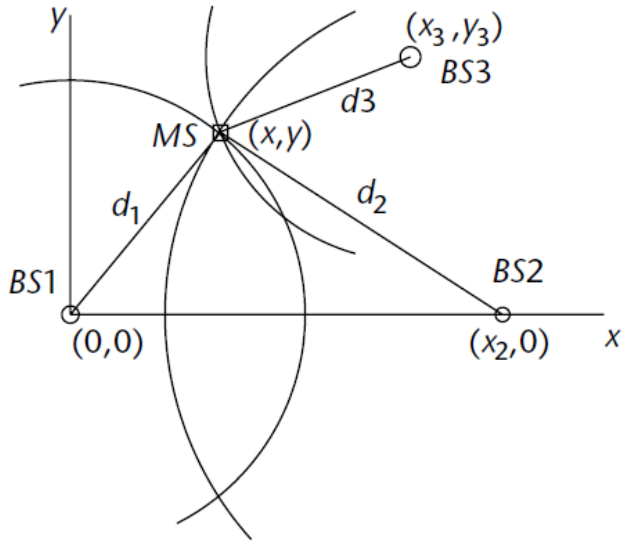


Figure 2.2 Two-dimensional three base terminals deployment (Bensky, 2016)

(2) Time Difference of Arrival (TDOA)

Estimation using TDOA is also based on ToF of signals. Instead of measuring the ToF of a transmission, it measures only the time difference when two terminals receive the signal of a transmission. Therefore, TDOA transmissions do not need to include time information. As a result, the modification of hardware or software can be reduced, which makes the method more applicable. By using the known positions of the fixed terminals and the time difference, a set of equations of a hyperbola can be generated. Then the position of the mobile station can be calculated by finding the intersection.

For TOA and TDOA, the actual distance measurements may not intersect at one point. Because of that, extra processes are required to deal with the error from measurement (e.g., least-squares error criterion).

(3) Angle of Arrival (AOA)

The AOA method estimates the position of the target by triangulation. In the two-dimensional case, with the known pose of at least two base stations and the direction of the arrival of the signal, the position of the target can be estimated as shown in Figure 2.3, where BS1 and BS2 are the positions of the known terminals and T represents the target. a_1, a_2 rep resent the AOA of the signals. Since the AOA measured by sensors has errors, a range is given to the measurement as uncertainty, as shown in Figure 2.4. The shaded area indicates the possible error of the estimation. When the target comes closer to the line between the two sensors, the area becomes larger. The two sensors are not able to estimate the position of the tag if it is on the line connecting the sensors. Therefore, for 2D estimation, the method should have three sensors not on the same line. More sensors can reduce the error of the result.

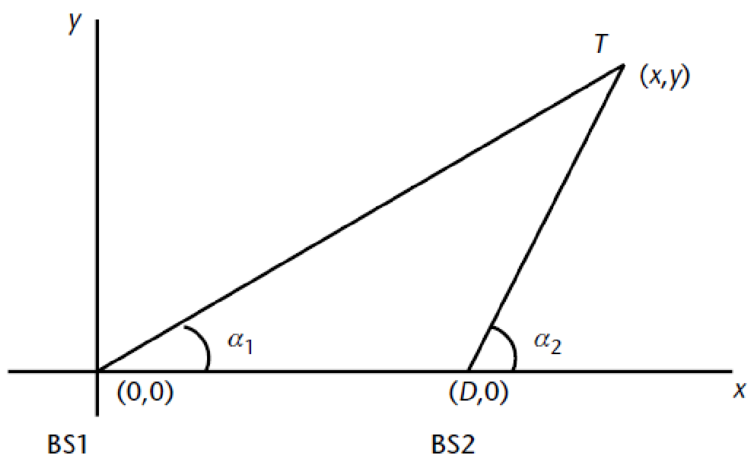


Figure 2.3 Triangulation in two dimensions (Bensky, 2016)

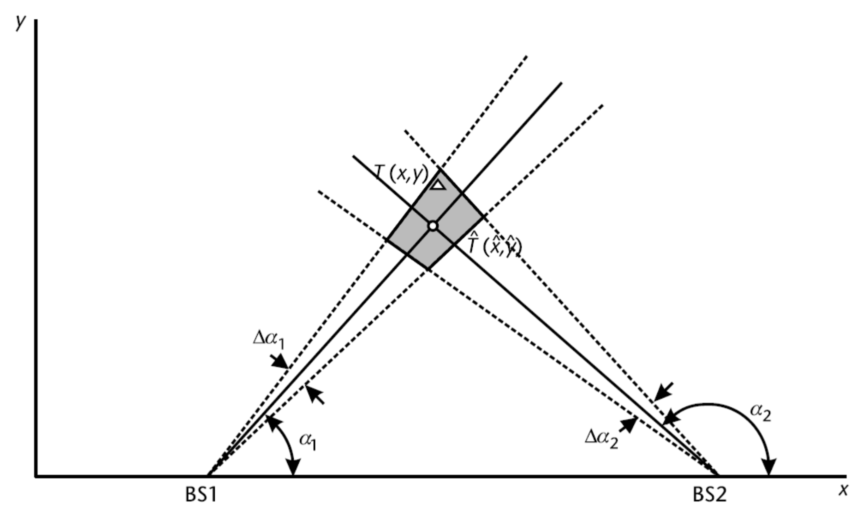


Figure 2.4 Position uncertainty due to antenna beamwidth (Bensky, 2016)

(4) Received signal strength (RSS)

Signal strength decreases as the distance of the transportation of the signal increases. By measuring the received signal strength indicator of a transmission, the distance from the transmitter to the receiver can be calculated. Similar to TOA, with the known positions of multiple receivers and the distances from the transmitter to the receivers, the position of the transmitter can be estimated. However, the RSS method cannot measure as accurate as the TOA method does.

2.2.1 Applying RTLS for Proximity Detection

Previous studies have tried to apply RTLS for improving construction site safety. According to Soltanmohammadlou et al. (2019), 75 papers discussed using RTLS for construction management from 2008 to 2018.

The active RFID was applied for preventing potential collisions of on-foot workers and heavy equipment (Chae & Yoshida, 2010; Teizer et al., 2010). The RFID was applied to monitor the proximities between workers and equipment. The RFID receiver can only intercept signals with enough RSS, which decreases with distance. By configuring the RSS from the transmitter, a safety area in which the receiver can intercept the signals is generated. When the entity equipped with an RFID receiver enters the area, alerts will be generated. However, their methods are not stable since the RSS of the RFID signals can be easily affected. The safety area has an irregular shape, as shown in Figure 2.5, which shows the boundary of a dangerous zone of a dump truck. The big differences in safety distances in different directions reduce the reliability in using this dangerous zone for generating alerts.

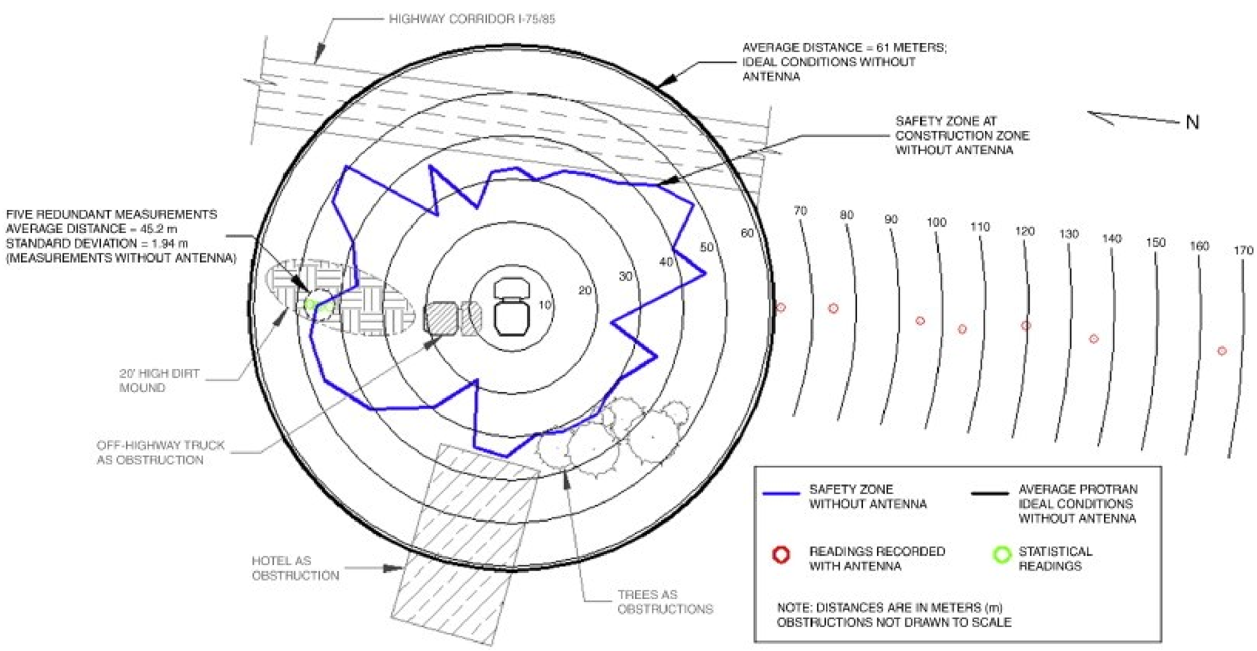


Figure 2.5 Dangerous zone of truck (Teizer et al., 2010)

Some researchers have tried other technologies to meet the accuracy requirement of safety monitoring. Wu et al. (2010) developed a real-time warning system using ultrasound, which has the highest accuracy among all the technologies (Li et al., 2016). By analyzing the historical records of accidents on construction sites, they listed three data requirements for the detection of potential accidents: the position of entities, identity information, and environmental information. To meet these requirements, they used ultrasound to get accurate position data, RFID to identify entities and Wireless Sensors Network to collect environmental information. Although ultrasound RTLS is one of the most accurate RTLS, its performance can be easily affected by obstacles.

In addition to the uncertain accuracy, the high false alert rate is also an important consideration when evaluating the feasibility. The frequent false alerts that do not specify a dangerous situation in reality (e.g., the distance between two entities is smaller than the threshold distance, but the two entities are moving in opposite directions) may eventually lead to ignoring alerts (Ruff, 2007). To decrease the false alert rate, an unsafe proximity detection method for outdoor construction sites was developed by Wang and Razavi (2016) considering the position, speed, and orientation of equipment and entities' reaction time. The authors applied a Kalman filter to smooth the raw data. When deciding whether alert needs to be generated, there are five situations to be considered as shown in Figure 2.6: (1) moving worker (MW) approaching static equipment (SE); (2) moving equipment (ME) approaching moving worker; (3) moving equipment approaching static worker (SW); (4) moving equipment 1 approaching moving equipment 2; (5) moving equipment approaching static equipment. In Figure 2.6, $\vec{P}$ and $\vec{V}$ represent the relative position and relative speed of one entity with respect to the other entity, respectively.

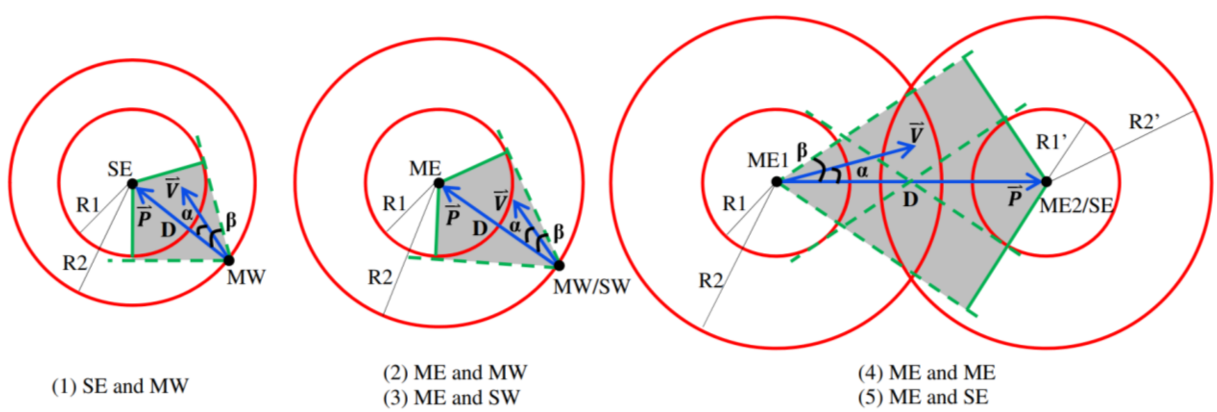


Figure 2.6 Five situations of proximity (Wang & Razavi, 2016)

In their study, they defined two safety distances for the equipment, which are the alert distance ($R1$) and warning distance ($R2$). The alert distance is determined based on the maximum length of the equipment. The warning distance is used for predicting the potential risk, which is entities inside the alert area of a piece of equipment. The warning distance considers the speed of entities and their reaction time. The definitions of the warning distance for proximities between equipment and worker and proximities between equipment and equipment are described in Eq. (2-1) and Eq. (2-2), respectively.

$$R2 = R1 + BD \tag{2-1}$$

$$R2 = R1 + R1' + BD \tag{2-2}$$

where $R1'$ is the alert distance of the other equipment, BD is the buffer distance, which is the difference between $R1$ and $R2$.

The definitions of BD for these two proximities are described in Eq. (2-3) and Eq. (2-4). They consider the reaction distance (RD) and the braking distance (BD')

For proximities between worker and equipment:

$$BD = (RD\ of\ worker) + (RD\ of\ equipment) + BD' \tag{2-3}$$

For proximities between equipment and equipment:

$$BD = (RD\ of\ equipment\ 1) + (RD\ of\ equipment\ 2) + BD' \tag{2-4}$$

where

$$RD\ of\ worker = reaction\ time\ of\ worker\ \times speed\ of\ worker \tag{2-5}$$

$$\begin{aligned} RD\ of\ equipment &= (reaction\ time\ of\ operator + execution\ time) \\ &\times speed\ of\ equipment \end{aligned} \tag{2-6}$$

$$BD' = \frac{(speed\ of\ equipment)^2}{2 \times (deceleration\ of\ equipment)} \tag{2-7}$$

The safety rules for alert generation are shown in Figure 2.7. For proximity between worker and equipment, when a worker enters the alert zone, the most serious alert is generated. If a worker is in the warning zone, and the worker is approaching the alert zone of the equipment, the alert for the warning is triggered. Similarly, if the alert zone of a piece of equipment overlaps with another piece of equipment's alert zone, the most serious alert is generated. Otherwise, if the warning zones have overlap and one alert zone of equipment is going to overlap with another alert zone of equipment, the alert for the warning is triggered.

The consideration of velocity and reaction time can estimate the potential risks that cannot be reflected by the position. Hence, this method can reduce the dangerous zone of stationary or slowly-moving equipment to reduce unnecessary alerts. However, when generating the alert zone, this method only considers the size of the equipment. It does not consider the area which could be affected by the task. In addition, their method is developed on the assumption that workers and equipment will not suddenly change their state of movement. This assumption may also lead to missing alerts of potential risks. For example, in the case that a worker is moving near the alert zone of stationary equipment but not approaching it, no alert is generated. However, if the worker suddenly changes his movement direction, he will be very close to the edge of the alert zone. Not enough time is given to the worker to react and then he will be in the workspace of equipment which is dangerous.

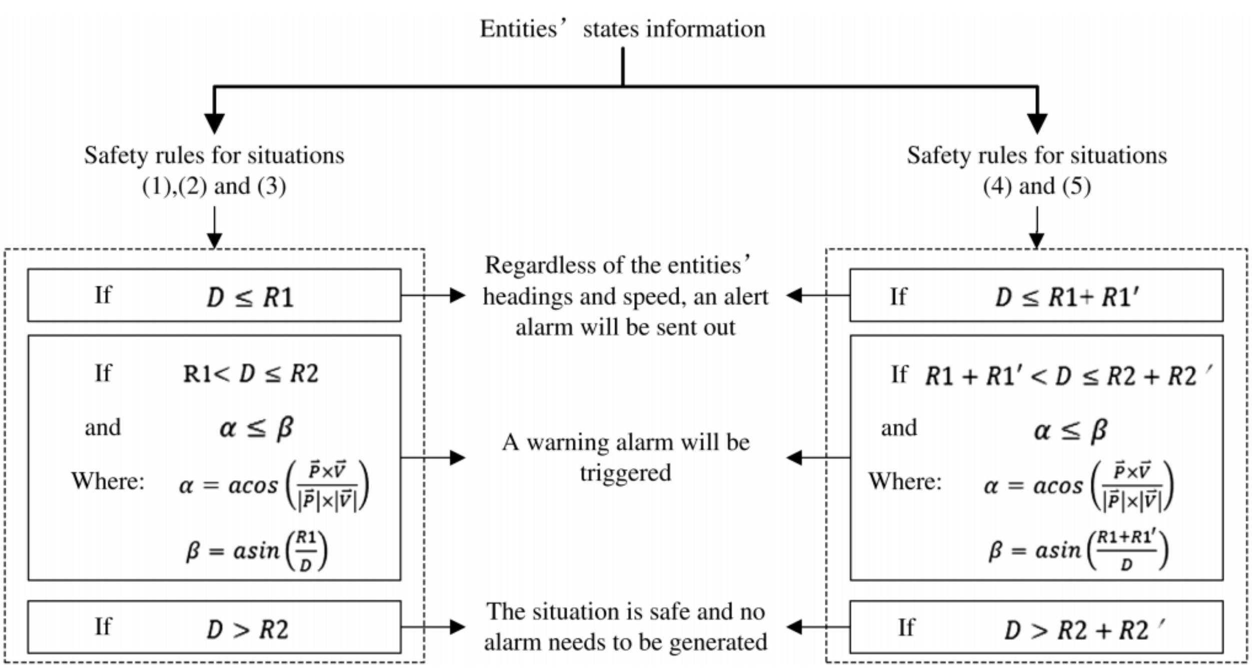


Figure 2.7 Flowchart for unsafe-proximity identification (Wang & Razavi, 2016)

In addition to the consideration of the movement of entities, with the sub-meter accuracy provided by UWB, previous researchers also tried to estimate the pose of equipment to detect the potential collisions. Zhang et al. (2012; 2012) used UWB system to capture the pose of a crane and built a multi-agent system (MAS) prototype which can detect potential collisions (Figure 2.8). They listed six requirements of using UWB RTLS which are accuracy, visibility, scalability and real-time requirement (data package should fit the maximum reading capacity of sensors), tag form factors, power supply and networking environment. It should be noted that most of the listed requirements are also required for other kinds of sensor-based RTLS. With the configurations of the system to meet the mentioned requirements, eight steps were designed for the data processing and pose estimation of the crane: (1) Identifying the ID of tags to identify which tags represent which part of the crane; (2) Filtering raw data based on heuristics. For example, if tags are outside the expected area or have too high velocity, the data will be deleted; (3) Calculate missing data using extrapolation; (4) Averaging data over a certain period T_k (which is called synchronized); (5) Smoothing data through another filter considering Geometric Constraints (GC); (6) Filling missing data using GC; (7) Averaging over tags; (8) Estimating the pose. With the estimated pose of cranes, the MAS prototype detects the potential collisions between cranes and applies re-planning in near real-time. During the tasks, the poses of the cranes are sent to the coordinate agent and the crane agent. Their poses and movements are reflected virtually in Autodesk Softimage. Once a potential collision is detected by the crane agents, the coordinate agent will decide the priority of the crane based on the pre-design priority rules. Then, the lower-priority crane agent should re-plan a new path which is sent to the operator for implementation. The potential collision detection and the path re-planning are conducted by using the Motion Strategy Library and the Proximity Query Package integrated into Softimage.

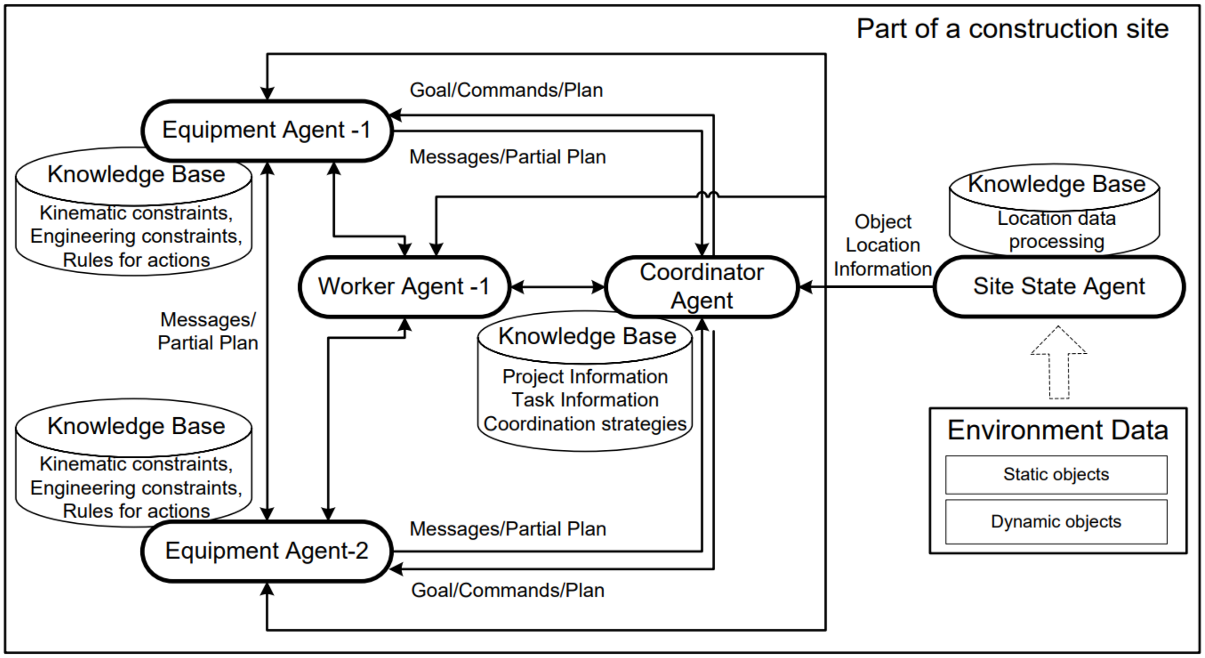


Figure 2.8 Multi-agent system for crane pose estimation (Zhang & Hammad, 2012)

By using UWB data, Vahdatikhaki and Hammad (Vahdatikhaki & Hammad, 2014) built a method to get the state of an excavator in near real-time. Then, an optimization-based pose estimation method (Vahdatikhaki et al., 2015) was developed to find the pose of the excavator, as shown in Figure 2.9. The estimation starts with reading data for a period of time ΔT which is the analysis period. Then, the data are averaged over a shorter period of time dt. To fill the missing data, interpolation is used. Extra processes are developed in order to eliminate errors. The optimization-based corrections are applied, and then the pose of the excavator during ΔT is estimated based on the corrected positions of data collectors which are tags.

With the near real-time pose of the excavator, dynamic equipment workplace (DEW) (Vahdatikhaki & Hammad, 2015) is generated to improve construction safety. The cylindrical workspace (Figure 2.10 (a)) and buffer workspace (Figure 2.10 (b)) have limitations, such as not fully considering the movement of the equipment. Their method generates DEWs considering the geometry, pose and speed of the excavator.

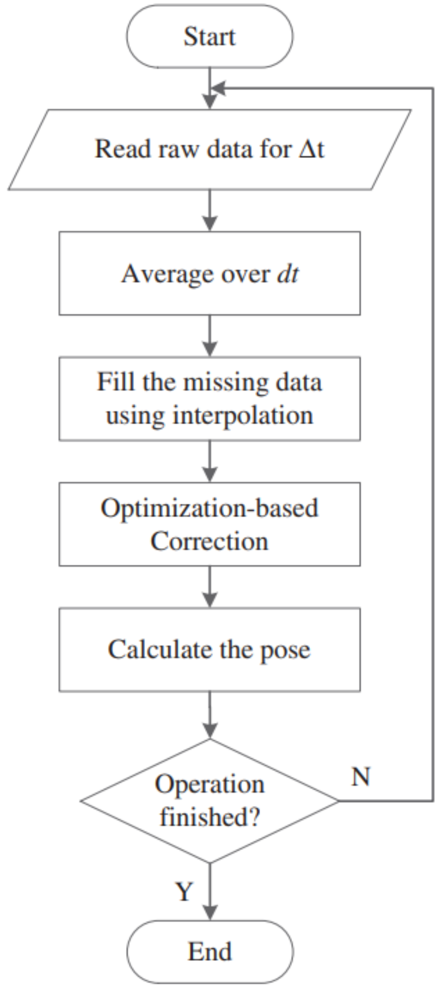


Figure 2.9 Flowchart for excavator pose estimation (Vahdatikhaki et al., 2015)

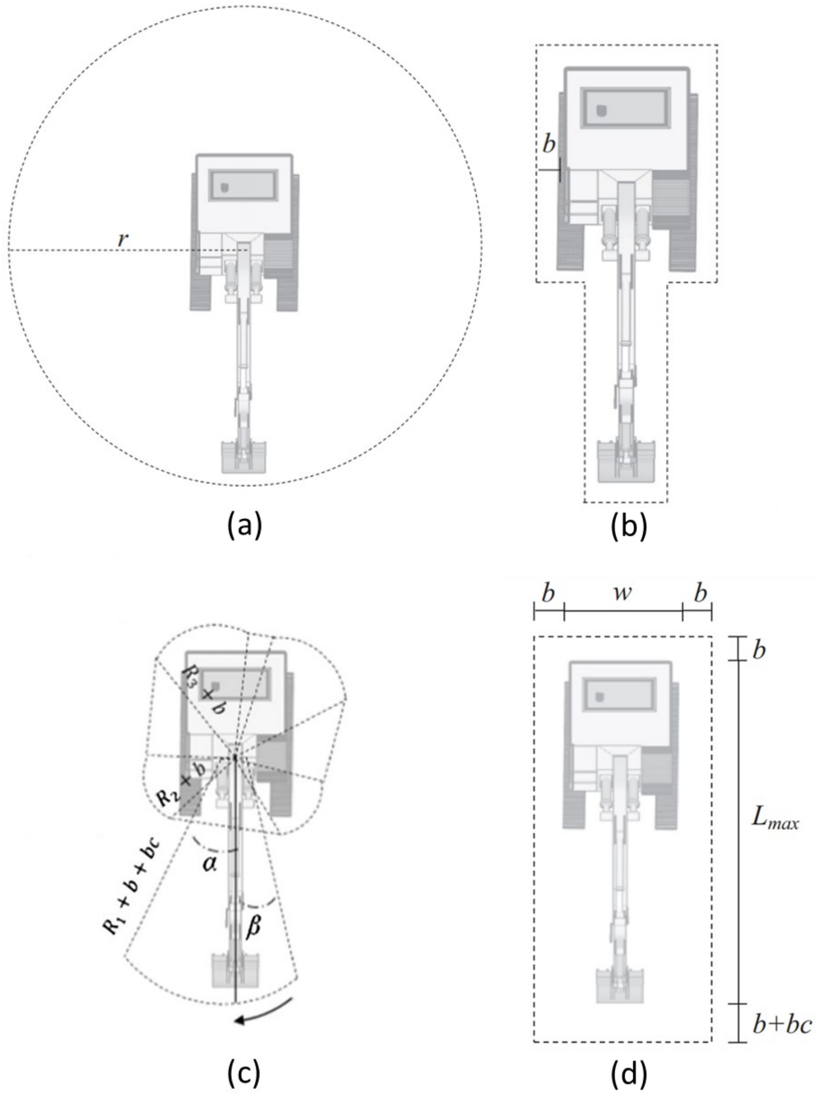


Figure 2.10 (a) Cylindrical workspace, (b) Buffer workspace, (c) Proposed workspace in the stationary state, (d) Proposed workspace in traversal states (Vahdatikhaki & Hammad, 2015)

Figure 2.10 (c) shows the DEW of the excavator in the stationary state, which means that the excavator is in a swinging state or loading/dumping state. R_1, R_2 and R_3 are the radius of the bounding box representing each part of the equipment and b is a buffer. The angle $\propto$ represents the amount of swing of the excavator with current angular speed (ω_1) and acceleration/deceleration (τ_a) over the stoppage time of the operation (t_s) which is calculated according to Eq. (2-8). The angle β represents the amount of swing of the excavator if it stops swinging to its current direction and starts to swing to the opposite over t_s. It is calculated according to Eq. (2-9), where ω_1 is the current angular speed and τ_s is the predefined value of swinging acceleration/ deceleration in the case of swing direction shift. bc represents the bucket motion clearance buffer which is determined by the projection of the speed of the tip of the bucket on the horizontal plane (v_{b_x}), the projection of the acceleration/deceleration (τ_{b_x}) and t_s as given in Eq. (2-10).

$$\propto = \frac{1}{2}\tau_a {t_s}^2 - \omega_1 t_s \tag{2-8}$$

$$\beta = \frac{1}{2}\tau_s (t_s - \frac{\omega_1}{\tau_s})^2 - \frac{{\omega_1}^2}{2\tau_s} \tag{2-9}$$

$$bc = \frac{1}{2}\tau_{b_x} {t_s}^2 - v_{b_x} t_s \tag{2-10}$$

Figure 2.10 (d) shows the DEW of excavator in traversal state where L_{max} and w are the length and width of the equipment, b is the buffer and bc represents the motion clearance buffer of the excavator. The buffer bc can be calculated according to Eq. (2-11), where τ_t is the acceleration/deceleration, v is the speed and t_s is the stoppage time of the operation.

$$bc = \frac{1}{2}\tau_t {t_s}^2 - v t_s \tag{2-11}$$

With the DWGs generated for pieces of equipment on a construction site, potential collisions can be detected between different pieces of equipment and the congestion index on the construction site can be calculated.

The previous researchers also consider the schedule for generating safety alerts. Zhang et al. (2012) developed dynamic virtual fences based on the schedule for generating alerts to workers who enter the dangerous workspace. With the information about the tasks, schedule and location retrieved from the 4D BIM, they defined the dynamic workspaces and imported them into the BIM for generating dynamic virtual fences.

The previous researchers have applied RTLS to detect proximities and risks on construction sites and more considerations are given in the decision to improve the accuracy of the potential collision estimation. However, the previous research did not consider the situations that workers and equipment are working together in the same team. The workers and equipment in the same team may get very close in order to finish the tasks. In this case, the movement of the equipment should be very slow and the operator of equipment should pay attention to his teammates. Alerts Generated in such circumstances should be considered as false alerts, which may disturb the workers and operators and eventually lead to ignoring the alerts. In this research, a method considering the grouping information retrieved from the micro-schedule is developed to eliminate the false alerts being assigned to entities in the same team.

2.2.2 Applying UWB RTLS on Construction Sites

Previous studies have done experiments to test the performance of applying UWB RTLS in the construction environment. Maalek and Sadeghpour (2013; 2016) conducted experiments for tracking static resources in an indoor construction environment using UWB. They measured the Distance Root Mean Squared (DRMS) and the Mean Radial Spherical Error (MRSE) which represent the accuracy of the UWB RTLS in the 2D plane (X-Y) and 3D space (X-Y-Z), respectively. The DRMS and the MRSE are calculated according to Eq. (2-12) and Eq. (2-13), respectively, where (x_i, y_i, z_i) represent the coordinates read by the RTLS and (x_{Actual}, y_{Actual}, z_{Actual}) represent the ground truth measured by a total station.

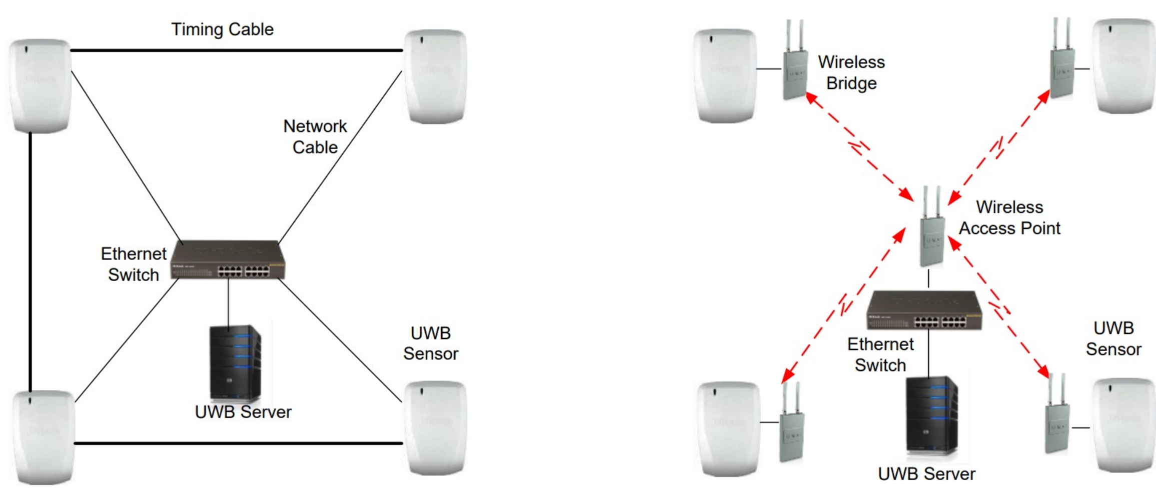
$$DRMS = \sqrt{\frac{\sum_{i-1}^{n}(x_i - x_{Actual})^2}{n} + \frac{\sum_{i-1}^{n}(y_i - y_{Actual})^2}{n}} \tag{2-12}$$

$$MRSE = \sqrt{\frac{\sum_{i-1}^{n}(x_i - x_{Actual})^2}{n} + \frac{\sum_{i-1}^{n}(y_i - y_{Actual})^2}{n} + \frac{\sum_{i-1}^{n}(z_i - z_{Actual})^2}{n}} \tag{2-13}$$

They designed seven experiments in order to test: (1) the overall accuracy, (2) the impact of the presence of metallic objects on the site, (3) the impact of occlusion by moving equipment, (4) the impact of attaching tags to metallic surfaces, (5) the impact of using the wireless scheme of UWB (using only AOA), (6) the impact of monitoring several tags exceeding the data handling capacity (7) the impact of using fewer sensors. The wired scheme (using TDOA and AOA) is used to ensure accuracy and performance, but the need for cables may decrease its applicability. The wireless scheme (using only AOA) can avoid the cabling problem but has negative impacts on the performance of the UWB RTLS. Figure 2.11 shows the comparison between the wired scheme and the wireless scheme. Table 2-2 shows the results of the seven experiments. Overall, the system can provide sub-meter positioning for static objects in indoor construction sites. 2D measurements are more accurate than 3D measurements. The average relative error is used to show the impact on the performance in different scenarios. The higher average relative error means that the experiment has more errors than the base experiment. The results show that the presence of metallic objects, moving obstacles, removing timing cables and lack of sensors have significant negative effects on the accuracy of UWB RTLS. Too many tags and attaching tags to the metallic surface have less impact on the UWB RTLS performance.

Table 2-2 Results of the seven experiments (Maalek & Sadeghpour, 2013)

No.	Experiment	Average accuracy (cm)		Minimal accuracy (cm)		Average relative error		Comparison basis
		2D	3D	2D	3D	2D	3D	
1	Base experiment	16	34	41	79			–
2	Multipath effect	24	60	67	146	97%	111%	Exp. 1
3	Signal blockage	33	65	52	68	231%	219%	Exp. 1
4	Metal surface	15	30	47	69	8%	18%	Exp. 1
5	Remove timing cable	27	37	53	63	114%	58%	Exp. 4
6	Number of tags	34	60	34	77	27%	11%	Exp. 4
7	Number of receivers*	14	26	54	76	78%	42%	Exp. 4



(a) (b)

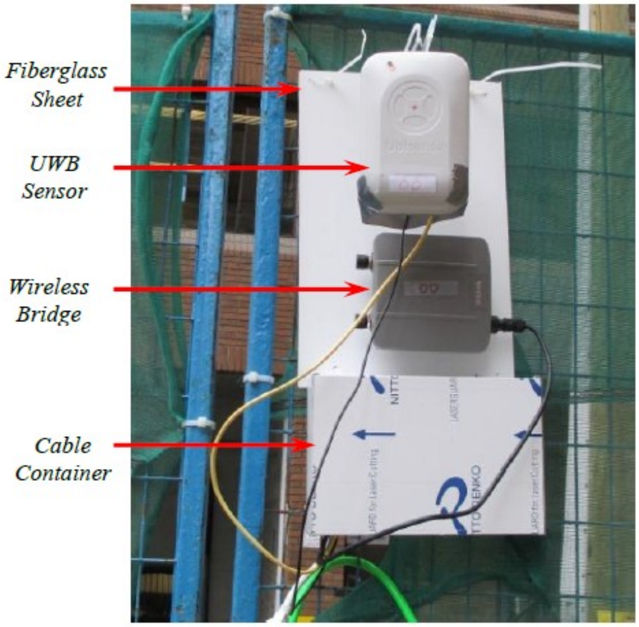
Figure 2.11 Schematic diagram of UWB system (Zhang et al., 2012)

Siddiqui (2014) have conducted an indoor case study to compare the performance in receiving data of the wireless scheme and wired scheme of UWB RTLS. The performance in receiving data can be reflected by the missing data rate (MDR). The MDR is calculated according to Eq. (2-14).

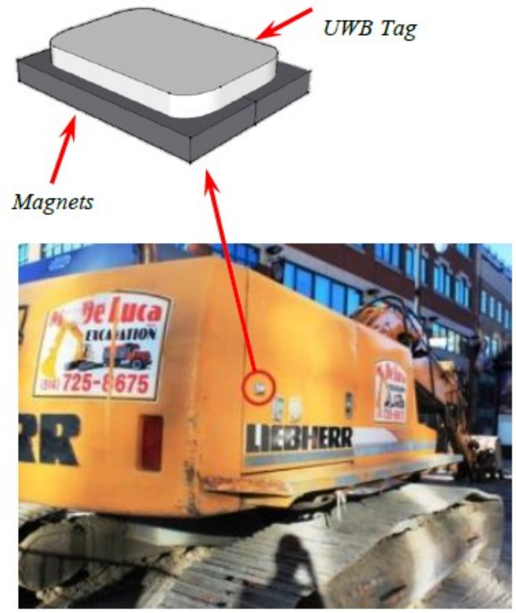
$$MDR = 1 - \left(\frac{Actural\ Total\ Readings}{Expected\ Update\ Rate \times Total\ Duration\ (sec)} \times 100\%\right) \qquad (2\text{-}14)$$

Four sensors were installed at the four corners of a confined room with dimensions of 4.16 m × 7.32 m. Three tags with an expected update rate of 34 Hz were attached to an RC-crane. The RC-crane rotated the fully extended boom clockwise during the tests. When applying the wired scheme of UWB system, the average MDR of the three tags was 18.68 % if the RC-crane was placed at the center of the UWB covered area. When the RC-crane was near the edge of the covered area, the average MDR was 9.94 %. When using the wireless scheme, the average MDR is 77.56 % and 70.69 % for the position at the center and near the edge, respectively. Caused by the impact of dilution of precision and the effects from the surrounding objects, more missing data were found in the second position (near the edge). In both positions, the MDR is higher when applying the wireless scheme.

Another case study (Siddiqui et al., 2019) was conducted in a construction site in downtown Vancouver to evaluate the feasibility of the wireless scheme to track dynamic entities. The installations of sensors and tags are shown in Figure 2.12. They attached tags on excavators to track their poses. The position data of tags and the missing data rate were analyzed. In their results, the system can capture the poses of construction resources in outdoor conditions. Applying the wireless scheme of UWB is feasible, but the performance is lower which can be reflected by the positioning accuracy and the missing data rate (MDR). In addition, the results of their case study are not as expected compared with the results from other studies testing the wireless scheme of UWB (Maalek & Sadeghpour, 2013; Teizer et al., 2008). This can be attributed to the uncontrollable calibration process on construction sites in their case (Siddiqui et al. 2019).



(a)



(b)

Figure 2.12 (a) UWB sensor panel (b)Tags with magnets (Siddiqui et al. 2019)

In order to evaluate the performance of UWB RTLS in outdoor construction sites, Cheng et al. (2011) compared the positioning results from UWB and robotic total station (RTS) as the ground truth. In their experiment, they used a robust Kalman filter to smooth sensory data. Then, a synchronization was performed to resample the two different signals of data (RTS data and RTLS data) to the same frequency. Their time synchronization was performed by maximizing the cross-correlation which is the measurement of the similarity between two signals. The authors used up-sampling to process the RTS data, so no information was lost. All UWB data from $t_{i-\frac{1}{2}}$ to $t_{i+\frac{1}{2}}$ are used to compare with RTS data at t_i, as shown in Eq. (2-15) ~ Eq. (2-18), where W_{ij} is the weight factor, $U[x, y, t_j]$ is the position from UWB at time t_j, while $R[x, y, t_i]$ represents the position from RTS at time t_i.

$$Error[t_i] = \sum_{j \epsilon J(i)} W_{i,j} Error[t_i, t_j] \tag{2-15}$$

$$Error[t_i, t_j] = \| U[x, y, t_j] - R[x, y, t_i] \| \tag{2-16}$$

$$W_{ij} = \frac{4t_j - 4t_i + 2}{\Delta T^2} \tag{2-17}$$

$$\sum_{j \epsilon J(i)} W_{i,j} = 1 \tag{2-18}$$

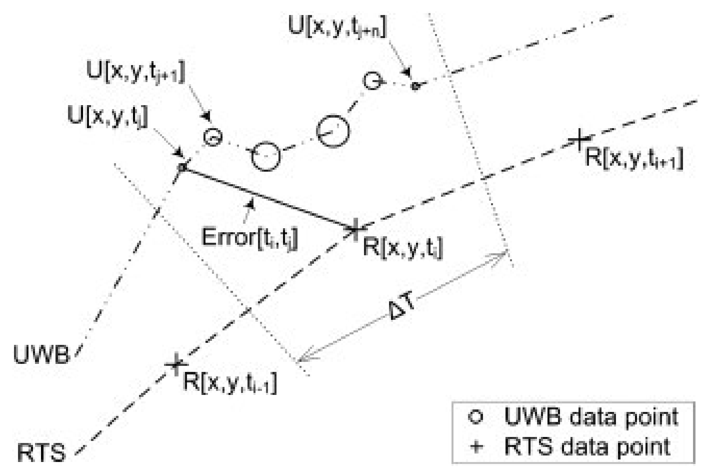


Figure 2.13 Schematic of error computation: UWB location track signal and visualization of comparison with RTS signal. (Cheng et al., 2011)

In their study, trails on a real construction site were made to evaluate the performance of UWB in making positioning. The first field trial was conducted in a confined work area of about 2,400 m^2 with eight sensors. And the second trial is conducted in an open area of about 65,000 m^2 with nine sensors. According to their results, a commercially-available UWB is able to provide real-time location data of construction resources. When the UWB coverage diameter increases to 70 m, the error rates are still within the expected tolerance. When the diameter is increased to 270 m, the error rate grows, but the accuracy is still within 4 m while over 75% of the errors are within 2 m.

In summary, the previous researchers have conducted tests to evaluate the performance of UWB RTLS on construction sites. Their results showed that the performance of UWB RTLS was good when the system was operated with timing cables. Since the UWB RTLS are usually based of TDOA and AOA, when removing timing cables, the accuracy of the system dropped significantly and more missing data appeared. Although the timing cables can improve the accuracy of UWB RTLS, they give more hard tasks to the project management such as higher cost, difficult site planning and house-keeping. Applying an RTLS based on other technologies not requiring timing cables can break the trade-off between the performance and difficulty in project management. One solution is applying the BLE RTLS based on AOA which can provide similar accuracy and coverage. This research proposes a wireless installation scheme for applying BLE RTLS based on

AOA in a large construction site. Tests are conducted to evaluate the feasibility of applying this scheme and the BLE RTLS on construction sites.

2.3 BLE RTLS based on AOA

BLE (Bluetooth Technology Website, n.d.) is integrated into Bluetooth 4.0 and the following versions (current version is 5.2). It operates in the 2.4 GHz radio frequencies as the classic Bluetooth technology. Unlike the classic Bluetooth, which is used mainly for streaming applications, BLE is developed to transfer small data (Afaneh, 2019). For applications that only require small data updates (e.g., states of the beacon, heart rate), BLE can provide the same data transferring performance with much lower power consumption. So, the batteries of BLE tags have a longer life. Also, because the data packet is smaller, the data processing time is less and the development process is simpler. The BLE only has a smaller number of channels (3 channels) than the classic Bluetooth (32 channels). Fewer channels lead to quicker discovery and connections. These advantages make BLE ideal for real-time Internet-of-Things (IoT) applications that must operate for a long time. Besides, the 100 m coverage range makes this technology feasible in positioning. The direction of arrival (DOA) information of the BLE signals can be measured by the Switched Beam System (SBS) or Adaptive Array System (AAS)(Monfared et al., 2018). The DOA information can be processed into AOA information through algorithms such as MUSIC (Tang, 2014).

The proposed research uses a BLE RTLS based on AOA call Quuppa (2019). The application and performance of the system have been discussed in the field of health care and asset management in large indoor environments (Cao et al., 2018; Van der Ham, 2015). Van der Ham (2015) developed an asset tracking system for hospitals based on BLE RTLS. He concluded that Quuppa system can make sub-meter positioning in the hospital and BLE RTLS is suitable for indoor positioning. Cao et al. (2018) compared their self-developed BLE RTLS with Quuppa system. They developed a prototype based on the two BLE RTLSs that can monitor the position of an infant and generate alerts to parents when an infant is inside a danger zone. They concluded that both BLE RTLSs meet the requirements for developing infant monitoring and alert generating system. Moreover, Quuppa provides better accuracy and less false alerts than the self-developed BLE RTLS, although the response time for alerts is longer.

2.4 Tactile Alerts

Previous studies (Ding et al., 2013; Lee et al., 2014; Zhou & Ding, 2017) have tried many methods to generate alerts to the workers and equipment operators on construction sites. Marks and Teizer (2013) developed a proximity warning system which generates visual alerts (flashing LED) and audible alerts to equipment operator once proximity is detected. Teizer (2015) developed the Smarthat, which can generate audio alerts to on foot workers who are facing proximity risks. Yi et al. (2016) developed an early-warning system for heat strains of workers. When detecting the heat strain of a worker, visual alerts will be generated through the wristband to the worker according to the level of heat strain through a wristband. However, those studies did not discuss in detail how to effectively generate alerts to the workers.

Visual, auditory and vibro-tactile signals are the three main ways to generate alerts. It is easier to provide visual alerts to the operators of heavy equipment than to on-foot workers because the operators have a known field of view. The performance of audio alerts on construction sites is nerfed because of noise. Sinclair and Haflidson (1995) reported that the average noise on construction sites is 98.8 dBA. Suter (2002) averaged the noise generated by different equipment. The results are very close or higher than the noise limit (90 dBA) regulated by OSHA (Government of Canada, 2020). Besides, construction workers are exposed to different kinds of risks. They should always pay attention to the surrounding conditions during their tasks. Therefore, the workers' visual and acoustic senses are occupied.

On the other hand, tactile stimuli can catch the attention of the workers without using their visual sense and acoustic senses. Another advantage of using vibro-tactile alerts is that the response time of the human body to vibro-tactile signals is shorter. Peon and Prattichizzo (2013) studied the reaction to three kinds of stimuli. They concluded that vibro-tactile stimuli provide a faster response than audio and visual stimuli. Furthermore, they also found that the reaction time from hybrid stimuli (visual and tactile) showed no difference from the reaction time from only tactile alerts. In addition, while compared with the other two stimuli, vibro-tactile stimuli are easier to perceive under workloads. De Souza et al. (2018) developed an experiment to evaluate the performance of visual alerts, auditory alerts and vibro-tactile alerts when the receivers are under certain workloads. They designed a monitoring task to simulate the workload. Users during the long-duration supervision tasks may easily lose their attention. From the results, they found that vibro-tactile alerts will lead to the shortest reaction time compared to audio and visual alerts. Moreover, their post-test survey showed that most users preferred vibro-tactile alerts.

Frequency, amplitude, vibration length, pattern and excitatory direction are the five interior factors of vibro-tactile signals (Hwang & Hwang, 2011). Many researchers have done case studies to understand how the five factors affect the perception of vibro-tactile signals. Feige (2009) designed an experiment to simulate real-world situations and test the perception of vibro-tactile signals from wristbands. He designed five different patterns to be recognized (Figure 2.14). The result shows that almost every pattern was correctly identified by participants without significant differences.

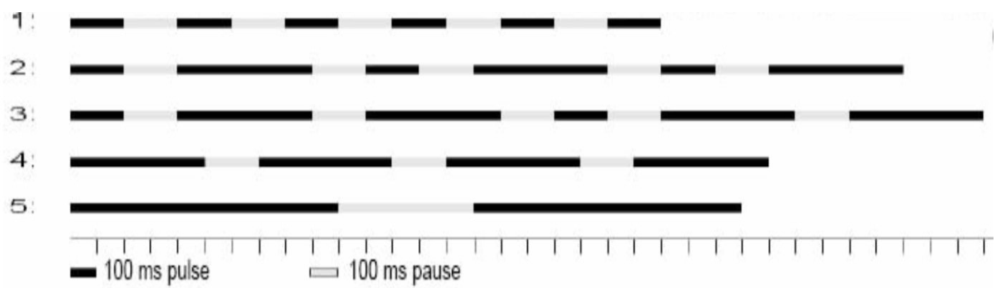


Figure 2.14 Five patterns of vibration (Feige, 2009)

Saket et al. (2013) also made an experiment to see how patterns affect the perception of vibro-tactile signals. They designed eight different patterns (Figure 2.15) for testing and found that simple patterns are easier to distinguish.

Exterior factors also have effects on the perception of vibration. For example, in cold weather, worker's coat and gloves will reduce the perception. Workloads can also be a cause of less recognition of vibration signals. Moreover, different parts of the human body have different perception ability. Another case study (Hwang & Hwang, 2011) showed that the different vibro-tactile signal can affect the user's emotion.

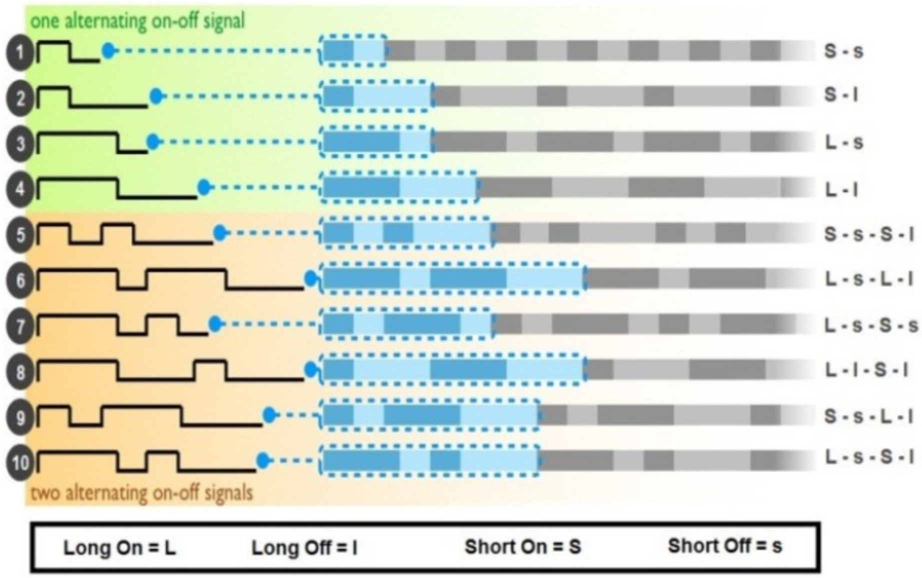


Figure 2.15 Alert patterns (Saket et al., 2013)

2.5 Summary

The literature review presented in this chapter focuses on RTLS technologies, experiments for RTLS performance evaluation and the RTLS applications in the domain of proximity detection and generating alerts. Several researchers have proved that the applications of RTLS can detect proximities and improve construction safety. But there are limitations depending on the technologies being used. The RFID RTLS is easy to deploy, but most RFID RTLS cannot meet the accuracy requirement in safety management. Ultrasound is accurate, but its performance is not stable. The cost of surveying level positioning with GPS is high. In addition, the GPS does not work in an indoor environment, and its performance can be easily affected by the weather. For UWB RTLS, they usually require timing cables which makes the installation and the management harder on construction sites. One RTLS technology that can address the mentioned limitations is BLE RTLS based on AOA. The review of BLE RTLS based on AOA and their applications shows its applicability for construction safety management. Moreover, the previous studies have considered multiple factors (e.g., pose of equipment, motion state, schedule) to improve the accuracy of proximity detection and reduce false alerts. But they did not consider the situation that the worker and equipment are in the same team.

This chapter also includes the literature reviews of vibro-tactile signals which will be sent to workers. The tactile signals can effectively transmit information to the receiver. Compared to other signals such as visual and audio signals, they are easier to perceive when the human is under workloads in a noisy environment. Previous researchers have also proved that the reaction time of the human body for tactile alerts is generally faster than the other signals. In addition, the vibro-tactile signal was proved to have the capacity to transfer information to human effectively. Using vibro-tactile signals to generated alerts representing different seriousness of proximities is promising. The form of the vibro-tactile signal is relating to the performance of the alert. The length of the signal, the strength of the signal and signal pattern are the three aspects to be consider.

Chapter 3: Providing Safety Alerts in Outdoor Construction Site

3.1 Introduction

A method for generating near real-time proximity safety alerts by using BLE RTLS based on AOA is developed to improve construction safety. In this chapter, the requirements of using the BLE RTLS on outdoor construction sites are listed with their solutions. Then, the methodology is proposed. The methodology consists of two parts, which are the data processing and proximity detection, as shown in Figure 3.1. In the data processing, raw data are processed to estimate the position, orientation and velocity of entities with attached tags. With the known information provided by the data processing, the proximities are detected considering the position of entities, movements and grouping information. Once a proximity event is detected, near real-time alerts are generated to the workers and equipment operators who are involved in the event. A prototype system is developed based on the proposed methodology.

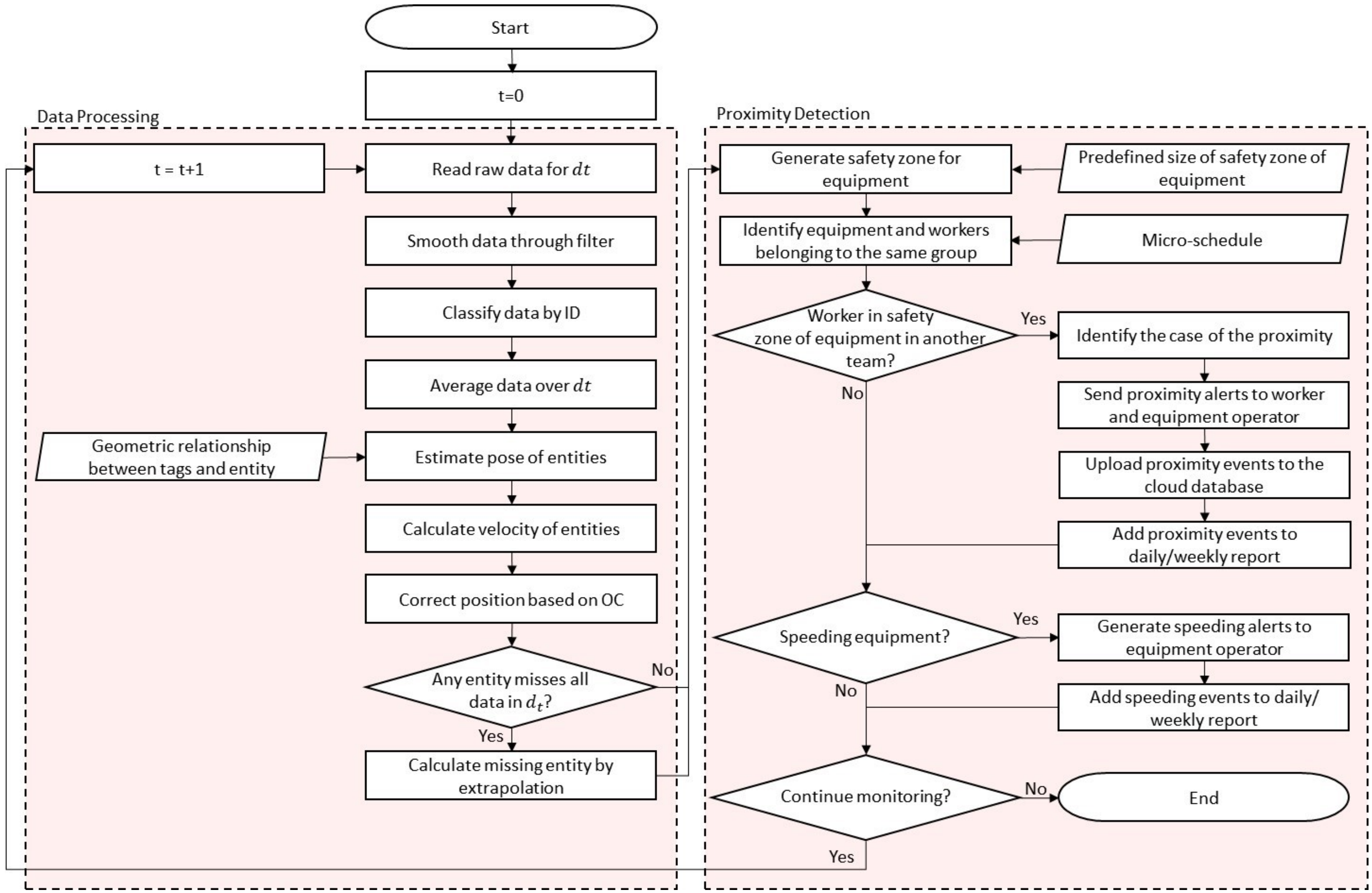


Figure 3.1 Data processing and proximity detection

3.2 Requirements of Using BLE RTLS Based on AOA

Six requirements for using UWB RTLS on construction sites are listed by Zhang et al. (2012). Based on their idea, five requirements are identified for the application of BLE RTLS on construction sites:

(1) Field of view (FOV) of sensors

FOV of sensors is the most critical factor affecting the performance of the system. In order to guarantee good performance, the sensors should be installed where they have a line of sight to as many tags as possible. Overlaps between the FOVs of sensors are also required by the AOA method for accurate positioning. For construction sites where achieving the line of sight is difficult, the most common way is to install sensors in a high position to get more visibility of tags.

(2) Data handling capacity of sensors

The RTLS sensors have limited capacity in handling tags data. The data handling capacity depends on the specific type of sensors. For the BLE RTLS used in this research, the maximum capacity that the sensors can handle is 250 packets/s. One data packet represents one transmission from the tag to the sensors. In order to avoid missing data caused by limited data handling capacity, tags should be set to have an acceptable update rate depending on the total number of tags on the site according to Eq. (3-1), where C is the data handling capacity of sensors, r_i is the update rate of tag i and n is the total number of tags. For instance, in this research, when 15 tags with the same update rate are activated, the maximum update rate can be set to 16 Hz.

$$C \geq \sum_{i=0}^{n} r_i \tag{3-1}$$

(3) Network environment and power supply

The sensors should be connected to the same network as the server computer. One solution to reduce the need for long ethernet cables is to build a wireless network environment using antennas as shown in Figure 3.2. The antennas are installed at the top of each pole and connected to sensors. Another antenna is installed on a pole near the site office and is connected to the office network. The power supply can be replaced by portable generators or solar panels placed near each pole to provide power to antennas and sensors.

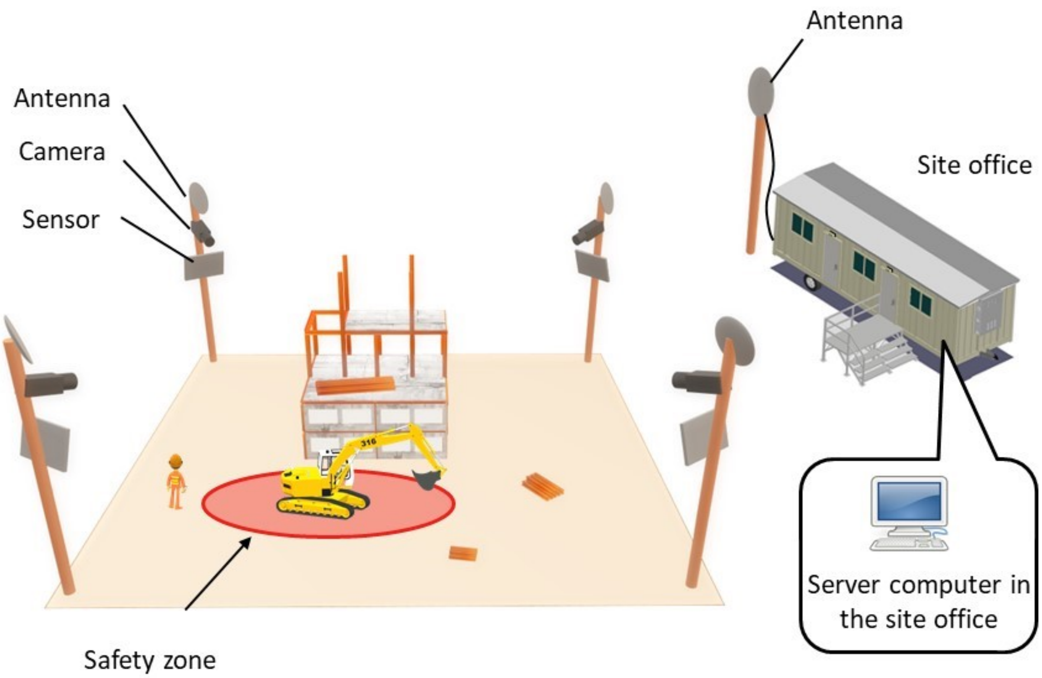
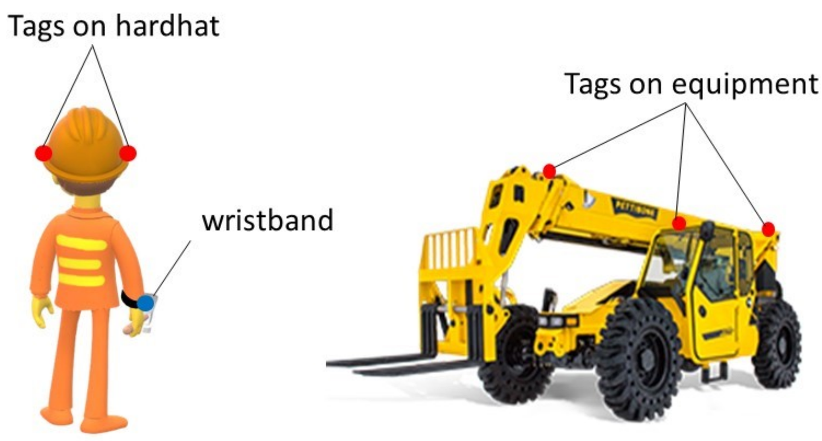


Figure 3.2 Using antennas to connect sensors to the server in the site office

(4) Visibility of tags

Although setting the sensors at a high position can help increase the visibility of tags, some occlusions are inevitable. Since the construction site is dynamic, tags can be occluded for short periods by different construction resources. To address this issue, several tags are attached to the same entity. Even if data of one tag is missing, the entity's position can still be calculated by using data from the other tags. In this research, two tags are attached on the sides of the workers' hardhat, as shown in Figure 3.3. For equipment, at least two tags are placed on the top surfaces where they have a line of sight to most sensors.



(1) (2)

Figure 3.3 Tags attached on hardhat and equipment

(5) Reflective materials

The metallic surfaces can reflect the BLE signals on construction sites. Since the BLE RTLS uses AOA to calculate the position of tags, reflections of signals will generate more errors in the results. One solution to avoid reflection effects from attaching tags directly to the metallic surface of a piece of equipment is adding a sponge pad between the equipment and tag.

3.3 Data Processing

The data processing attempts to process the raw location data from the RTLS to obtain useful information for proximity detection. After collecting the locations of tags from the RTLS, the raw data are smoothed by a filter (i.e., Kalman filter, exponential filter). However, the smoothed data still have many random errors caused by occlusions or other uncertain factors on construction sites. To minimize errors, extra processing is applied.

As mentioned in Section 3.2, to increase the visibility of tags, multiple tags are attached to the same entity. To identify the tags attached to the same entity, the classification step is performed. With this step, tags attached to the same entity are grouped according to the tags' ID. To get a generalizable position of tags in near real-time, data of the same tag are averaged over a short period of time (e.g., 1s). The position data of the same tag over a short period of time dt are averaged to represent the position of the tag at the time T_i, as shown in Eq. (3-2), where (x_{T_i}, y_{T_i})

represent the position of the tag at time T_i, (x_t, y_t) represent the filtered data during dt and n is the number of data of the tag that the sensors can capture during dt.

$$\begin{cases} x_{T_i} = \sum x_t/n \\ y_{T_i} = \sum y_t/n \end{cases} \tag{3-2}$$

After averaging data over dt, based on the known geometric relationship between tags on the entity, the pose and velocity of the entity can be estimated. According to Vahdatikhaki et al. (2015), using geometric relationships for pose estimation has an important assumption that the tags are attached to a rigid part of the entity. Taking the estimation of the worker position as an example, as shown in Figure 3.4(a), two tags are attached on the sides of the hardhat. The center of the hardhat can represent the position of the worker. By averaging over the two tags, the position of the worker can be estimated according to Eq. (3-3), where X_{T_i} and Y_{T_i} represent the position of the entity and m is the number of tags attached on the entity. The orientation of an entity is estimated based on the position of tags attached to the entity. Also, in this example, the orientation of the worker can be estimated as the lane perpendicular to the line connecting the two tags. θ is the angle with the x axis. In the case that only one tag is attached to the entity, the orientation of the entity is assumed to be parallel to its velocity.

$$\begin{cases} X_{T_i} = \sum x_{T_i}/m \\ Y_{T_i} = \sum y_{T_i}/m \end{cases} \tag{3-3}$$

The next step is to calculate the velocity of the entity (Figure 3.4 (b)). With the position of the entity at T_i and T_{i-1}, the velocity of an entity in the period $[T_{i-1}, T_i]$ is calculated according to Eq. (3-4), where V_{X_i} and V_{Y_i} represent the velocity of entity during the period $[T_{i-1}, T_i]$ in the x and y directions, respectively.

$$\begin{cases} V_{X_i} = \dfrac{(X_{T_i} - X_{T_{i-1}})}{T_i - T_{i-1}} \\ V_{Y_i} = \dfrac{(Y_{T_i} - Y_{T_{i-1}})}{T_i - T_{i-1}} \end{cases} \tag{3-4}$$

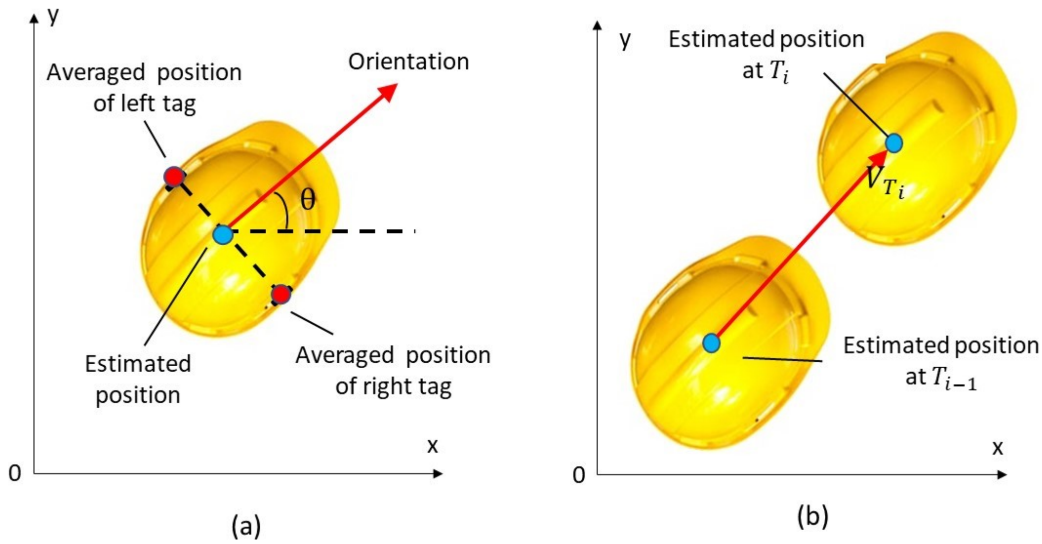


Figure 3.4 (a) Averaging over two tags, (b) Calculating velocity

Although the previous process can improve the accuracy, the estimated position of entities still has many errors. One of the errors that can be observed easily from the data is that the movement of an entity violates the operational constraints (OC) (Vahdatikhaki & Hammad, 2014). An operational constraint means the operational limitation of an entity (e.g., the maximum speed or rotation speed the entity can reach). In this research, since it does not involve estimating the rotation of entities, the maximum speed of entities is the only considered OC.

After calculating the velocity of an entity, if the velocity of the entity exceeds the OC, the position of the entity will be corrected to where it should be if it moves with the maximum velocity. For example, as shown in Figure 3.5, $P_{T_{i-1}}$ and P_{T_i} are the consecutive positions of the entity after the estimation. If the velocity in the period $[T_{i-1}, T_i]$ exceeds the speed limit, the correction is made and ${p_{t_i}}'$ is the corrected position of entity. The distance of the movement of an entity in the period is calculated according to Eq. (3-5), where v is the velocity calculated with the position before correction, v_{oc} is the maximum velocity and dt is the duration of the period.

$$\begin{cases} D = v \cdot dt & (v < v_{oc}) \\ D = v_{oc} \cdot dt & (v > v_{oc}) \end{cases} \tag{3-5}$$

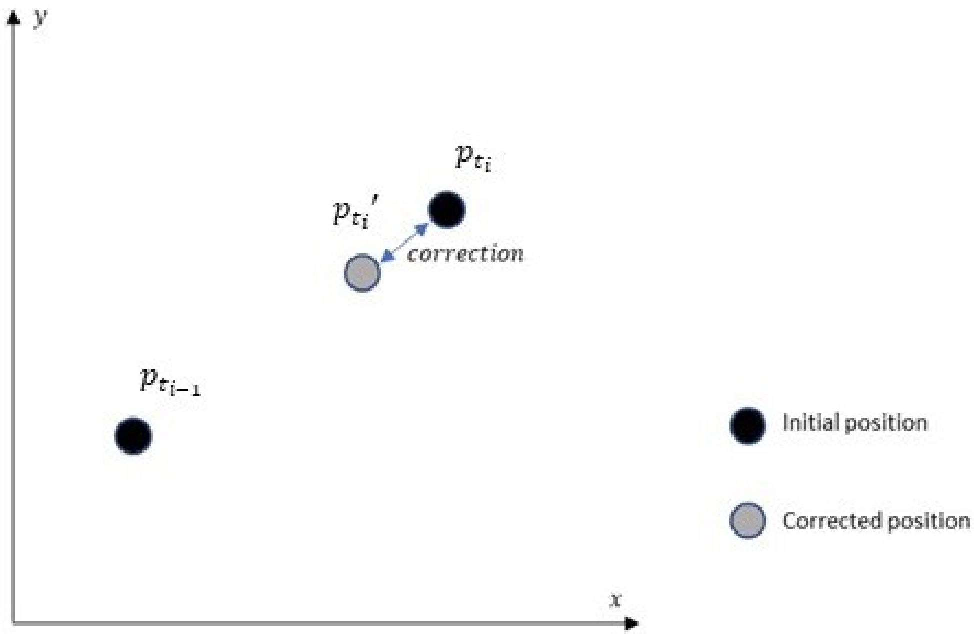


Figure 3.5 Operational constraint correction

Three kinds of equipment are being monitored in this research: truck crane, boom lift and telehandler. The maximum travel speed of the telehandler is 32.8 km/h (9.11 m/s)(Caterpillar Inc. 2012). The maximum travel speed of the boom lift is 6.84 km/h (1.9 m/s) (*JLG Industries, Inc.*). For the truck crane, the maximum travel speed can reach above 80 km/h. However, applying such high speed as OC is useless since it is not possible to accelerate to such high speed on construction sites. Therefore, the estimation of truck crane will not consider the OC. For workers, the OC is set to 13.62 km/h (3.78 m/s), which is the average running speed of men aged 18-34 (EnduranceMachine, 2019).

At last, estimation is made to deal with error caused by the situations that sensors do not capture any data from the tags during the period dt. The entity with all the tags data missed during the period dt is considered as a missing entity at the time T_i. Extrapolation is used to estimate the missing data assuming that the entity will move at the same speed. As shown in Figure 3.6, the velocity in the period $[T_{i-2}, T_{i-1}]$ and position at T_{i-1} are used to calculate the position of the missing entity according to Equation (3-6).

$$\begin{cases} X_{T_i} = X_{T_{i-1}} + V_{X_{i-1}} \times (T_i - T_{i-1}) \\ Y_{T_i} = Y_{T_{i-1}} + V_{Y_{i-1}} \times (T_i - T_{i-1}) \end{cases} \tag{3-6}$$

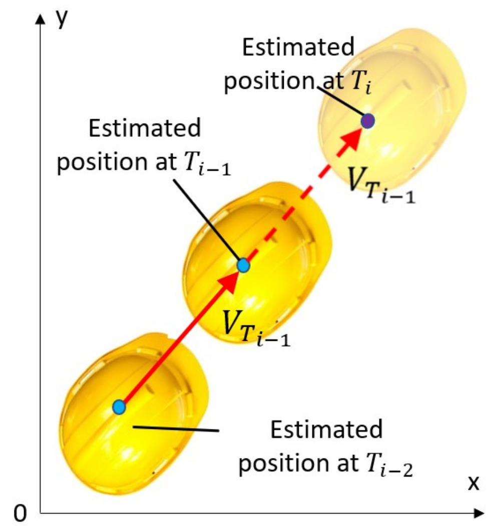


Figure 3.6 Extrapolation to fill missing data

3.4 Proximity Detection

With the positions and velocity estimated in data processing, and the predefined size of safety buffers of equipment, the dangerous zone of equipment is generated. Another step is then developed to reduce unnecessary proximity alerts generated to the workers and equipment operators in the same team. At last, with the positions of workers, dangerous zones of equipment, movement information of entities, and the grouping information, three cases of proximities can be detected. Once proximity is detected, the system generates the corresponding alerts to all related workers and operators.

3.4.1 Generating Dangerous Zone for Equipment

The cylindrical workspace and the buffer workspace (Figure 2.10) are two popular workspaces that are generated and used for collision avoidance (Vahdatikhaki & Hammad, 2015). Wang and Razavi (2016) developed a method generating a cylindrical workspace with dynamic size which generates less false alerts when compared with other methods considering only the position. In this research, the generation of dangerous zone for equipment is based on their method. As explained in Section 2.2.1, Wang and Razavi considered the position and size of equipment, velocity, and reaction time when generating dangerous zones for equipment. Their method can effectively reduce the false alert rate. However, there are limitations in their method. It may not effectively notify the workers when they are at a very dangerous position because the size of the alert zone is only determined by the size of the equipment without considering the nearby area where the task may have influence. Besides, there are situations that the worker will not get alerts even if he is at a dangerous position, because their method does not consider the situations that the worker and equipment are not moving. For example, for the situation that a piece of equipment is not moving and a worker is standing close to the edge of the alert zone of the equipment, the calculated width of the warning zone will be zero because the velocity of the worker and the equipment are zero. If the state of movement of any entity changed suddenly, the worker will not get enough reaction time before entering the alert zone.

Some improvements have been made to their method considering the limitations. In consideration of safety, in this research, two buffers are designed to address the two limitations, as shown in Figure 3.7. R_0 is the radius of the workspace of the equipment. R_1 and R_2 represent the radius of the alert zone and warning zone, respectively. The alert buffer B_1 is added to the alert zone to consider the area where the task has influence. The value of B_1 depends of the size and the type of equipment. The warning zone consists of the warning buffer B_2 and the velocity buffer B_v. The warning buffer (B_2) is designed to ensure that workers will have enough reaction time to the warning alerts before moving into the alert zone. The calculation of this buffer is described in Eq. (3-7). It is based on the assumption that workers can stop walking once he makes a reaction to the perception of alerts.

$$B_2 = reaction\ time\ for\ vibro\ tactile\ alert \times average\ speed\ of\ worker \tag{3-7}$$

The velocity buffer is generated according to the velocity of entities considering the reaction time. Its width (B_v) is equal to the sum of the reaction distance of the equipment ($RD\ of\ equipment$) and the braking distance (BD'). The definitions of $RD\ of\ equipment$ and BD' are described in Eq. (2-6) and Eq. (2-7)

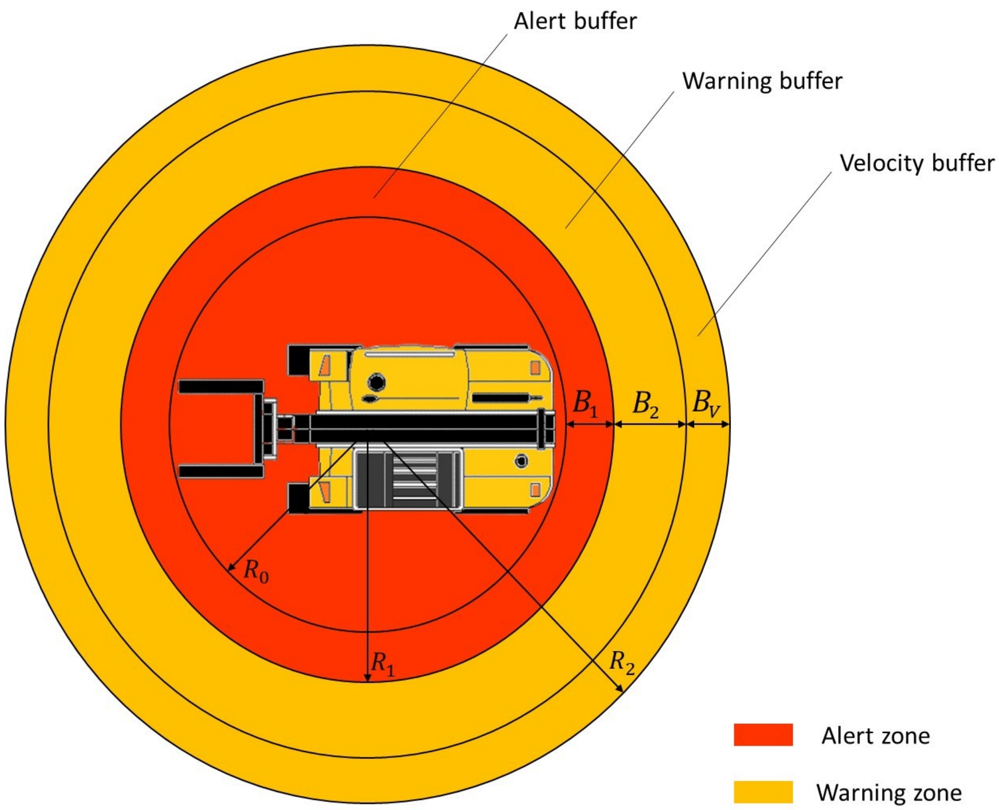


Figure 3.7 Dangerous zone of equipment

As shown in Figure 3.7, the area colored in red is defined as the alert zone, and the area in orange is the warning zone. If the worker enters the warning zone, two kinds of alerts representing different proximity levels will be generated according to the velocity direction of the equipment and the worker. If the worker enters the alert zone, another alert representing the most serious proximity case will be generated. The generation of alerts based on the cases of proximity will be further explained in Section 3.4.3.

3.4.2 Defining Groups Based on Micro-Schedule

Before detecting proximities, another step is developed to reduce unnecessary proximity alerts. On construction sites, there are situations that workers are working with equipment as a team. For example, when a worker is loading materials to a telehandler, the worker and the equipment may be very close. In this case, the movement of equipment should be slow. Besides, the worker and equipment operator should pay attention to each other during the operation. Generating alerts, in this case, may disturb the workers and the equipment operators, which may eventually lead to workers' ignoring the alerts. Therefore, the proximity alerts generated in this case should be considered as false alerts. To deal with these false alerts, in the proposed method, the proximity detection considers the micro-schedule for defining the groups.

A micro-schedule is a schedule listing tasks with the information of the team members in the tasks. Figure 3.8 shows an example of the micro-schedule for four workers and three pieces of equipment. Tasks are listed in the first column. Members who are in the same task within a specific period are shown in the figure.

Task	Team members	7:00~8:00	8:00~9:00	9:00~10:00	10:00~11:00	11:00~12:00	12:00~13:00	13:00~14:00	14:00~15:00	15:00~16:00	16:00~17:00
Welding and Preparing Material	Worker	Worker B;Worker D			Worker D				Worker A;Worker D		
	Equipment	N/A			N/A				N/A		
Prepration of Working at Heights	Worker	Worker A; Worker E	N/A								
	Equipment	N/A									
Hualing	Worker	N/A									
	Equipment	N/A	Telehandler								
Installing Elements	Worker	N/A	Worker A;Worker E						Worker E		
	Equipment	N/A	Boom Lift;Crane						Boom Lift;Crane		

Figure 3.8 Example of micro-schedule

The micro-schedule has two sources. The project manager or the scheduler can provide the micro-schedule of the first source. This micro-schedule is required in advance for making real-time proximity detection. However, the micro-schedule is usually not exactly followed by the workers and equipment. Changes may be applied during the task. Therefore, the system may generate false alerts if a worker starts to help another team to finish their task. In case of sending false alerts to workers and equipment not considered in the same group, a function is added to the system allowing the worker to deactivate the alert. After receiving the first proximity alert, if the alert for the proximity with the same equipment is generated again, the worker can push the button on their wristband to deactivate the proximity events with the equipment until the next time step. A rule is added using the position information of entities. If the worker stays away from the equipment with a long-distance (e.g., more than 100 m) for more than 10 min, the system will consider that the worker has left the team. So, when the worker enters the dangerous zone again, alerts will be generated. The worker can tell the system again that he has returned to the group by deactivating the alert.

The other source of the micro-schedule is the videos of the site. This micro-schedule reflects the reality of the grouping relationship. By observing the video, the groups in different periods can be defined. In post-analysis, this micro-schedule can help to count the real number of alerts to compare with the number of alerts that have been sent to workers and operators.

3.4.3 Identifying the Case of Proximity

When proximity happens, the movements of the two entities are considered for deciding whether the two entities are getting closer. In this research, three cases of proximity are considered as shown in Figure 3.9: (1) the worker is in the warning zone but is not approaching the equipment; (2) the worker is in the warning zone and is approaching the equipment; and (3) the worker is in the alert zone of the equipment. For Case 1 and Case 2, soft vibro-tactile alerts will be generated to tell the specific worker and equipment operator that the worker is within the dangerous zone of the equipment. For Case 3, a strong vibro-tactile alert will be sent to inform workers that there is a highly risky proximity nearby.

Whether the two entities are getting closer can be known by projecting the estimated velocity of equipment and worker on the line connecting their positions, as shown in Figure 3.10. The projected velocity can be used to calculate the relative velocity of the equipment to the worker.

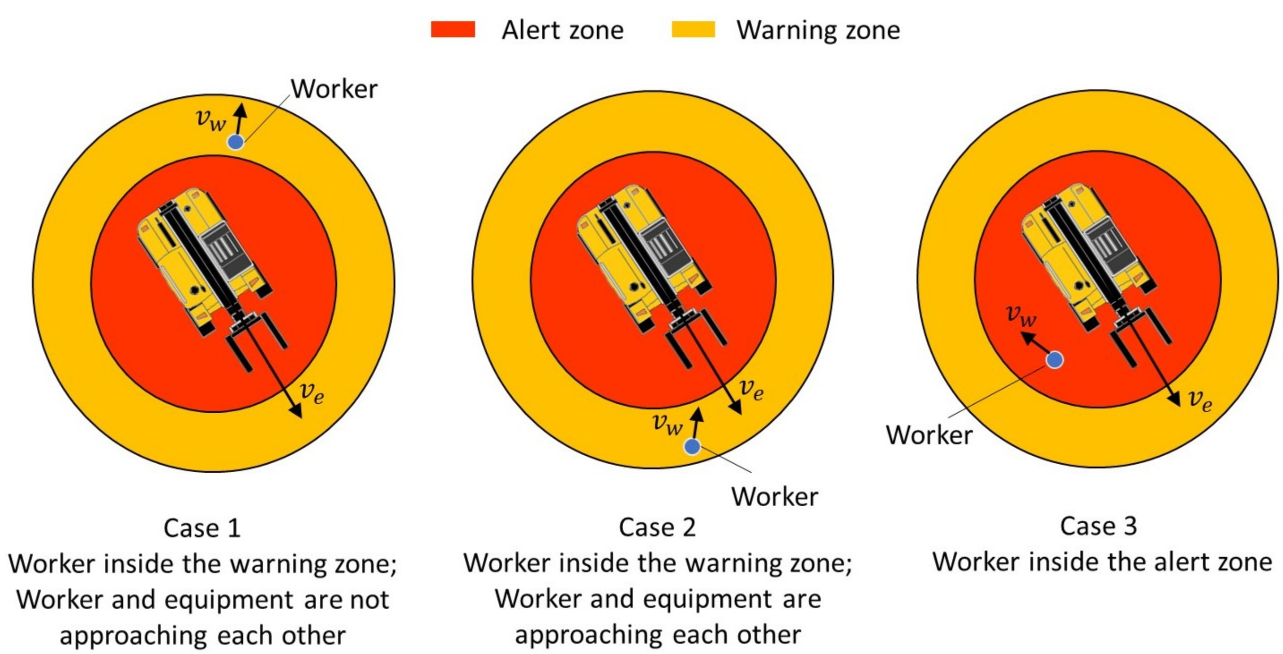


Figure 3.9 Three cases of proximities

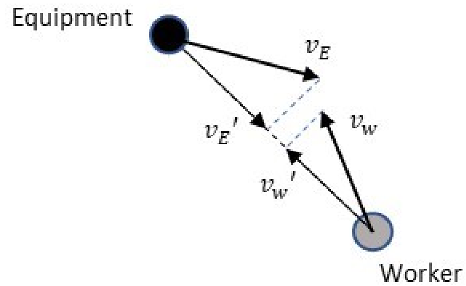


Figure 3.10 Velocity projection

3.4.4 Reducing Unnecessary Alerts

A speed threshold is set for the alert generation. Alerts will be generated only when the velocity of the equipment is higher than the threshold. In the situations that the equipment is moving slowly (i.e., moving slower than 10 km/h), workers can notice the risk with enough reaction time. So, there is no need to generate alerts for such situations (i.e., the alerts can be skipped). In addition, repeated alerts related to the same equipment in a short period may annoy the workers and lead to ignoring alerts. Therefore, after the generation of an alert, the repeated alerts that have been detected in the following short period will be skipped if they are related to the same equipment. This short period is defined as skipping period (Δt). To avoid the situations that workers may miss the alerts and the proximity event is becoming more serious, the changes to a more serious case will generate a new alert (i.e., Case 1 to Case 2, Case1 to Case 3, and Case 2 to Case 3). Figure 3.11 shows the decision tree for the alert generation when proximities are detected. The value of the speed threshold and the duration of the skipping period are discussed in Section 4.3.4.

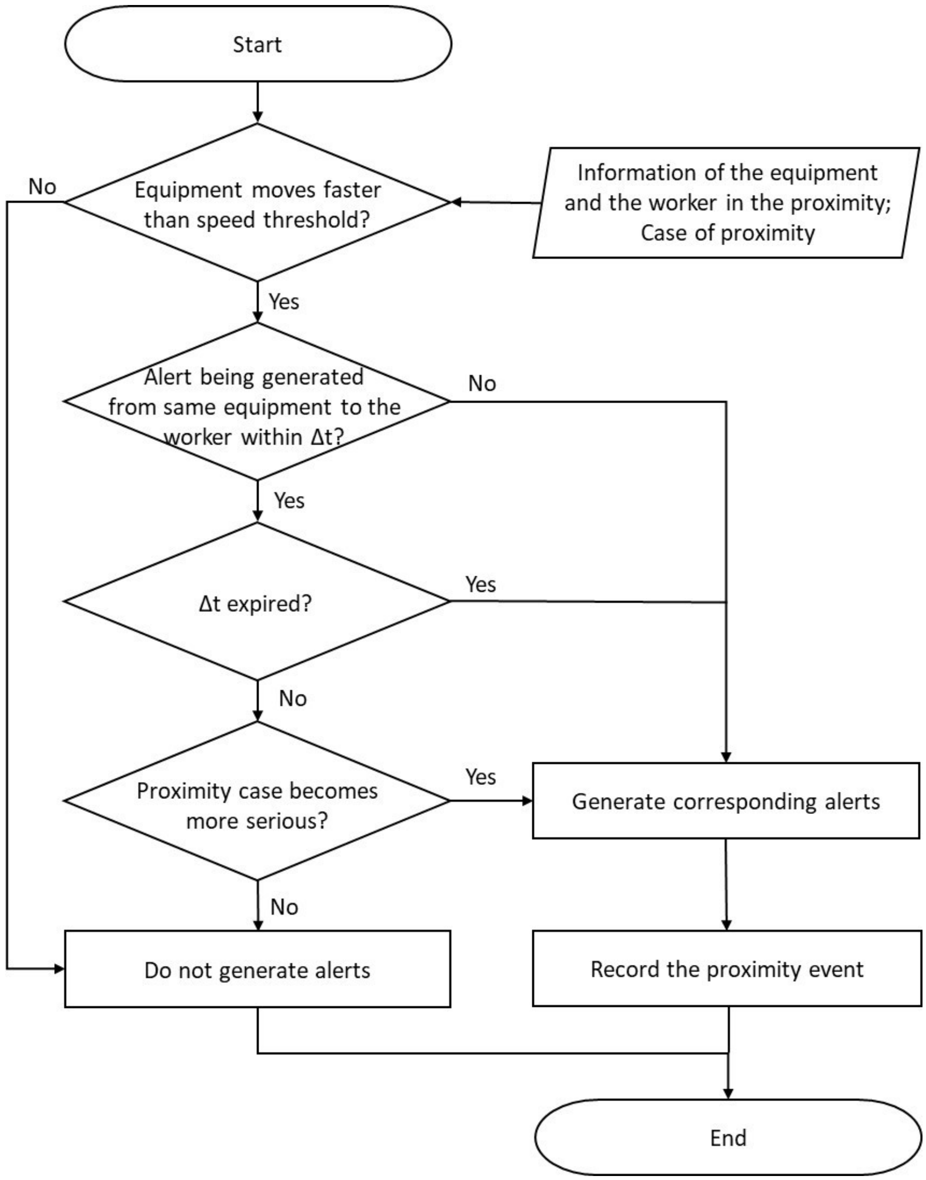


Figure 3.11 Decision tree for generating alerts

3.5 Generating Speeding Alerts

Other than the proximity alerts, the system generates alerts for speeding equipment. If the velocity of the equipment violates the speed regulation on the site, the system will send speeding alerts to the equipment. To differ the speeding alerts from proximity alerts, the speeding alerts will be generated through audio or a LED. Since the operators work in the cab of the equipment, the impact of noise is reduced. In addition, the operators have fix vision, so it is feasible to use visual alerts to catch their attention.

3.6 Defining Virtual Fences for Dangerous Zones

In addition to generating dangerous zones for equipment, virtual fences are also generated to improve safety on construction sites. Hammad et al. (2012) generated virtual fences for two types of areas for improving safety on construction sites: the places where physical guardrails should be installed, and the reserved areas required by tasks. By following their concept, virtual fences are generated for the following stationary danger zones in this research: (a) the storage zones of dangerous materials, (b) the areas with works at height and fall risks, and (c) the areas where dangerous facilities are installed (e.g., high voltage transformer).

3.7 Prototype Development

A prototype system is developed using a BLE RTLS based on AOA called Quuppa (*Quuppa*, 2019). The sensor model in this research is LD-7L. Quuppa LD-7D is a BLE locating sensor that is designed for positioning tags within a distance of 300 m in open space and outdoor environment. The RTLS server (Quuppa Position Engine) is set up on a Linux computer. The operating system of the computer is Ubuntu 18.06. After receiving AOA data from the sensors, the RTLS server captures the position of tags. Two filters are provided by the Quuppa Position Engine, which are the Kalman filter and exponential filter. In this research, the embedded exponential filter is used because it provides smoother data and is recommended by Quuppa. An API is provided by the RTLS server, which allows other software to obtain position data and send commands to the sensors and tags.

The system architecture is shown in Figure 3.12. After receiving the signals from tags, the sensors send the AOA data to the server computer. Then the server computer processes the AOA data to become position data and detects proximity events. Once a proximity event is detected, the server sends vibro-tactile alert information to the sensors to let sensors broadcast BLE commands to the wristbands, which are fully compatible with the BLE RTLS acting as tags with the ability to generate vibration alerts. Upon intercepting command from a sensor, the specific wristbands generate vibro-tactile alerts with the predefined patterns according to the case of proximity.

According to the technicians in Quuppa, when commanding tags, only one sensor will broadcast the BLE signal to the tags and any compatible device. If this sensor does not successfully command the tags or wristband, this broadcasting task will be assigned to another sensor. However, at any time, only one sensor will broadcast the commands. In normal conditions, the tags and wristbands should be able to intercept commands effectively from the RTLS if it is within the communication range of a sensor. The response time of the tags and wristband is affected by the tags' 'response mode'. The 'response mode' has three options: 'slow', 'normal', and 'fast'. The update rate of the wristband may limit the options of the 'response mode'. In order to activate faster 'response mode', the tags' update rate should satisfy the threshold update rate. For example, the update rate should be set to higher than 5 Hz if the 'fast' mode is expected. However, there is no evidence that shows a higher update rate provides a faster response time with the same 'response mode'. Thus, in the premise that response is fast enough, to save the data handling capacity of sensors, the update rate of the alert generating device should be set as low as possible.

A software is developed using Python to perform the data processing and proximity detection as mentioned in Section 3.3 and Section 3.4. The code for the software is attached in Appendix B.

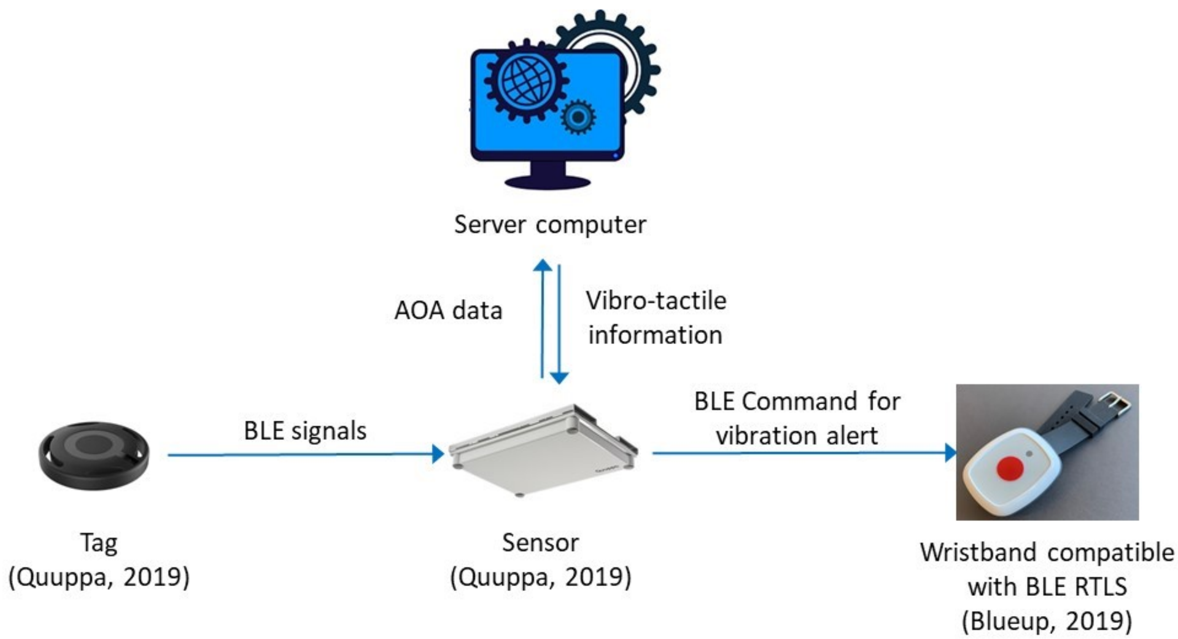


Figure 3.12 System architecture

3.7.1 Software Development

The software is developed based on Python. Using the API to communicate with the RTLS, the software is able to obtain position data of tags and send commands to the RTLS. A simple test was conducted to test the time needed for sending requests to the API for obtaining data and sending commands to the sensors. These two kinds of requests are sent to the API 50 times. The average time is about 17 ms to read data from the API and about 14 ms to send commands to the sensors.

In the data processing, the software obtains filtered position data of tags from the RTLS server. Then, the data are processed through the multiple steps according to Section 3.3 to estimate the pose and velocity of the workers and equipment. At the same time, a user interface plots the 2D map with the position of the workers and the equipment. Then, proximity detection is performed considering the estimated positions, velocities and the grouping information from the micro-schedule. Once a proximity is detected, alerts are generated to the workers and equipment operators in this event. In addition, the information of the event is recorded in a local hard disk for post-analysis. Details of the proposed prototype software are shown below by functions:

(1) Data capturing and recording

Filtered data are captured from Quuppa API through 'request ()' method. The data provided by the API is in JSON format, which is transformed into a 'Dictionary'. The data will be saved in the local hard disk and uploaded to the cloud database at the same time for post analysis and backup.

(2) Data processing

With this function, the filtered data of tags are processed to estimate the position and velocity of entities. The data processing is performed according to Section 3.3.

(3) Proximity detection

This function is developed according to Section 3.4. The position and velocity of entities and the grouping information are used for proximity detection. The three cases of proximities as shown in Figure 3.9 can be detected. Before the activation of the proposed system, the micro-schedule is imported to the system for retrieving the grouping information. Figure 3.8 is an example of the micro-schedule. The micro-schedule is built by Excel and is saved in the comma-separated values (CSV) format. For proximities between entities doing the same task in the same team, alerts will not be generated. Once a proximity between a worker and equipment in a different team is detected, alerts will be generated to the corresponding workers and operators. Simultaneously, the event will be recorded in the local hard disk and uploaded to the cloud database for post-analysis.

(4) Generating alerts

According to the cases of proximity, three kinds of vibro-tactile alerts are generated to workers and equipment operators through a wristband as will be explained in Section 4.6.1.

(5) Event recording

The proximity events will be recorded in a local hard disk in JSON format. At the same time, the events are uploaded to a cloud database (MongoDB, n.d.).The recorded information of the events includes: (a) the time when the event happens, (b) ID of the worker, (c) position of the worker, (d) ID of the equipment, (e) position of the equipment, (f) distance between the worker and the equipment, and (g) the proximity case.

(6) Reports generation

A summary report is generated at the end of the day. The report involves the following information: (a) the total number of different cases of proximities during the day, (b) the number of proximities for each hour, (c) the number of proximities that each piece of equipment has experienced, (d) the number of proximities that each worker has experienced. The report is saved on the local hard disk as an Excel document and also uploaded to the cloud database. In addition, a detailed report is generated, including the information of each proximity with the IDs of the workers and the equipment. It should be mentioned that the IDs are assigned randomly and are not linked to the specific workers.

(7) User interface

The user interface of the software has two parts, which are the graphic 2D map and the small area showing the information of the proximities. The map shows the position of workers and equipment attached with tags in near real-time. Once proximity is detected, information of the proximity, involving the ID of worker and equipment, their distance and the case of the proximity is printed in that area.

3.7.2 Device for Generating Alerts

In this research, the wristbands from Blueup (2019) are used for generating alerts as shown in Figure 3.12. The wristbands are fully compatible with the BLE RTLS used in this research. After intercepting the commands broadcasted by the sensors, two kinds of alerts can be generated by the wristband, which are vibro-tactile alerts through the vibration motor and audio alerts through the beeper. The patterns of the vibration and the audio are customizable. The customization is done in the setup phase of the RTLS. By using the API to let the sensors send commands based on different proximity, the wristbands can generate the corresponding pre-defined alerts.

In addition, the author tried to develop another wristband so that the vibro-tactile alerts can be fully customized. The development of the wristband uses Arduino Nano, Quuppa Tags Module (Figure 3.13) (Quuppa, 2019) and a vibration motor. Figure 3.14 shows the connection of the components of the device. The tag module can act as a tag of the RTLS, which can transmit BLE signals for positioning and intercept the commands from sensors. According to the commands, specific output pins of the tag module can be activated. The duration of the activation is decided by the commands. When the pin is not activated, the output pin has the same electric potential as the VBAT (battery voltage). When the pin is activated, the electric potential of the output pin is the same as GND (ground). An Arduino Nano powers the tag module. The Arduino Nano is an open-source hardware that can be programmed to interact with sensors and other electronic devices. The three digital pins (D3, D4, D5) on the Arduino Nano are set to ‘READ’ mode. They are connected to the Quuppa Tag Module to read the state of the three output pins (PIN 23, PIN 24 and PIN 25). The Quuppa sensors can broadcast BLE signals to command the specific tag module to activate one or more output pins. If any of the output pins of the tag module is activated, the corresponding pin on the Arduino Nano returns ‘LOW’. On the other hand, if an output pin is not activated, the corresponding pin on the Arduino Nano returns ‘HIGH’. By knowing which output pins on the tag module are activated, the Arduino can control the vibration motor which is connected to the analog pin A3 and GND to generate the suitable vibro-tactile alerts. The analog pin can generate a voltage within the range of 0 to 5 V. Therefore, the strength and pattern of the vibration can be controlled by controlling the output voltage of the analog pin. Besides, a 3D printing box with sliding closure is designed to hold the components as shown in Figure 3.15.

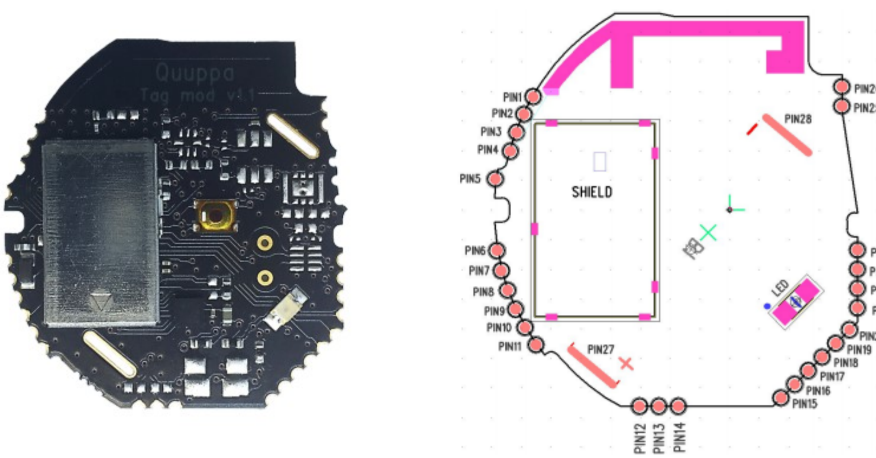


Figure 3.13 Quuppa tags module and pin placement (Quuppa 2017)

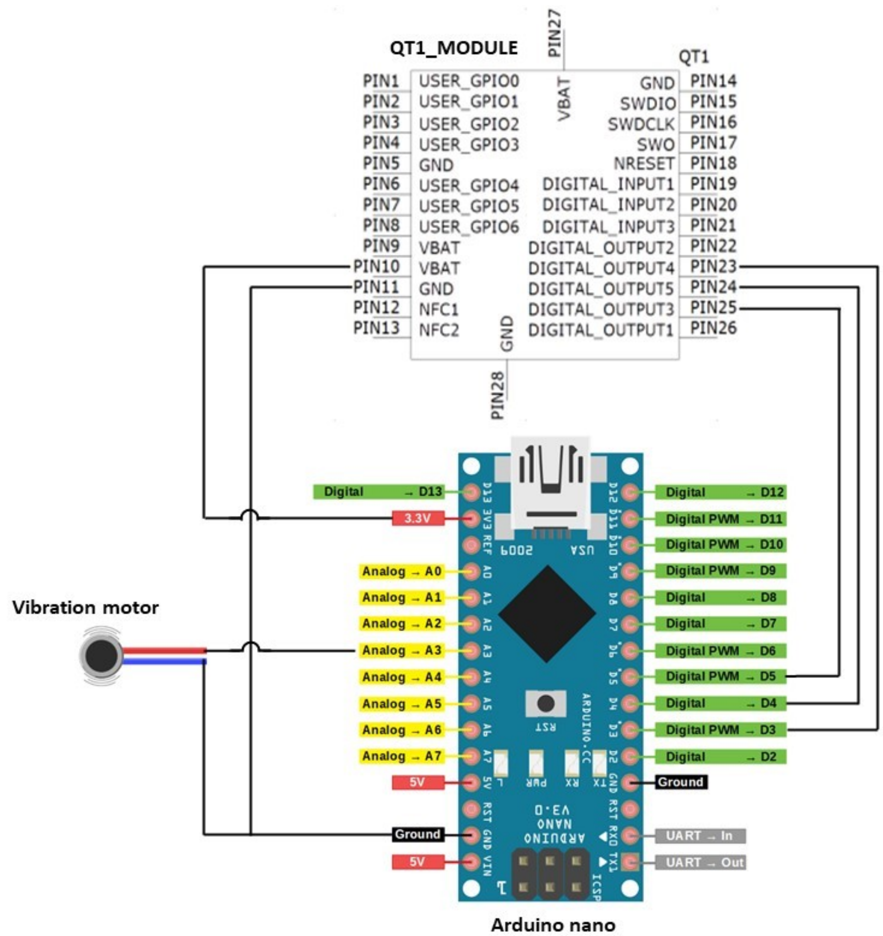


Figure 3.14 Connection of components

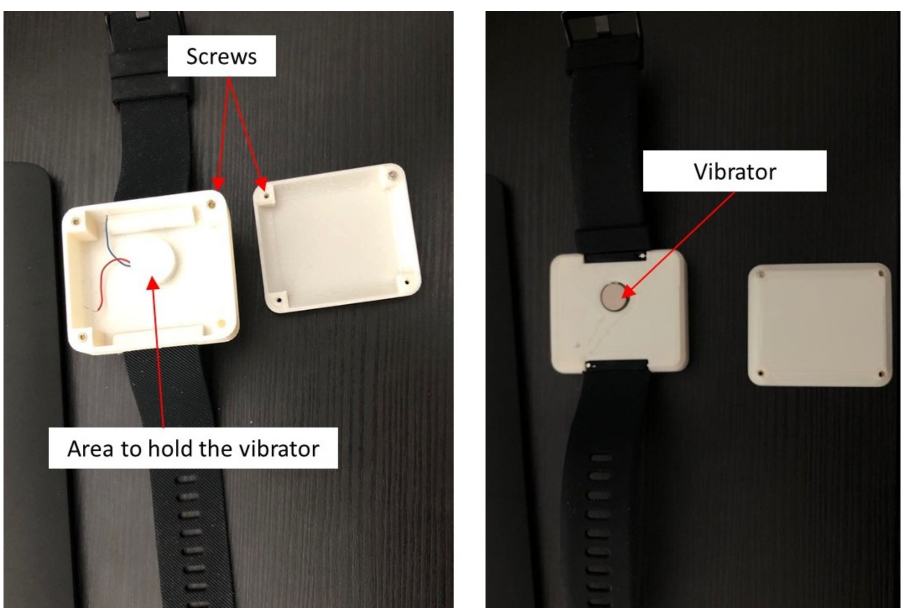


Figure 3.15 3D printing box

3.8 Laboratory Test

A laboratory test is conducted to test the performance of the BLE RTLS. The main objective of this test is evaluating the accuracy in capturing the position of tags in the following cases: (1) Tags are at a location where the RTLS can only make low resolution positioning; (2) Some of the sensors are occluded by objects; (3) Combination of the two previous scenarios.

3.8.1 Test Design

The test was conducted in a confined office with dimensions of 10 m × 4 m. There is a column with a diameter of 1.2 m in the office. Four LT-7D Sensors were installed on tripods at the height of about 1.5 m. The positions of the sensors are shown in Figure 3.16. The RTLS server provides a function to estimate the positioning accuracy in different areas. As shown in the figure, the green area represents a high-resolution positioning area, while the yellow area represents a low-resolution positioning area. For tags in the red area, the position captured by the RTLS will have large errors.

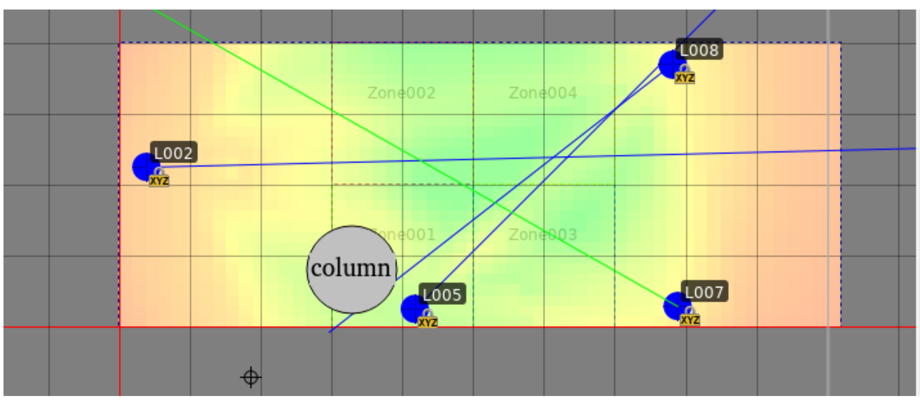


Figure 3.16 Accuracy estimation

Six tags were distributed into two groups. All the tags are placed on the same XY-plane. The tags in the first group (Tag 1, Tag 2, Tag 3) were placed next to each other on a straight line. During the test, the tags of the first group were not moved. The tags in the second group (Tag 4, Tag 5, Tag 6) are placed on another straight line 1.52 m away from the line of the first group. The straight line of the second group was parallel to the line of the first group as shown in Figure 3.17. Six scenarios were designed to simulate the three cases: (1) Tags in group 2 are in a good position with no sensor occluded; (2) Tags in group 2 are in a good position with one sensor occluded; (3) Tags in group 2 are in a good position with two sensors occluded; (4) Tags in group 2 are in a bad position with no sensor occluded; (5) Tags in group 2 are in a bad position with one sensor occluded; (6) Tags in group 2 are in a bad position with two sensors occluded. In this test, the update rate of tags was set to 5 Hz. The RTLS captured position data of the six tags for 1 min. By using the positions captured by the RTLS, the distance from tags in group 1 to the tags in group 2 can be calculated. The evaluation of whether those scenarios have impact on the accuracy was based on this distance.

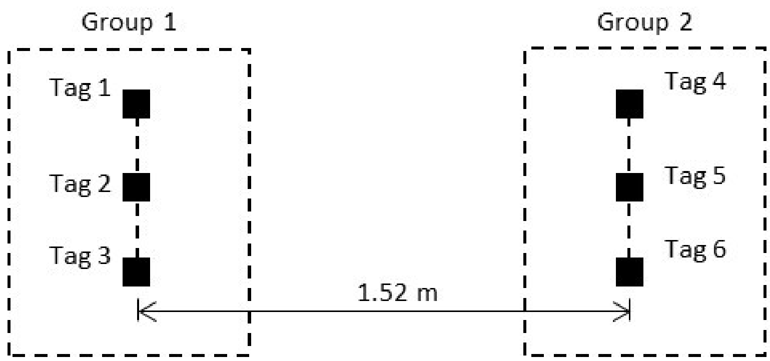


Figure 3.17 Placement of tags in the laboratory test

3.8.2 Results and Discussion

Table 3-1 and Table 3-2 show the average position of each tag over one minute and their standard deviation (SD) in the six scenarios. Overall, the SD of x value and y value are smaller than 0.03 most of the time, which means the positioning results are precise. No obvious evidence shows that SD values change in not ideal scenarios.

Table 3-3 shows the average distance between tags in a different group (tag1 and tag 4, tag2 and tag 5, tag 3 and tag 6). ΔD represents the difference between the measured distance and the ground truth. Overall, the differences between the measurements and the ground truth are less than 1m in all scenarios. For scenario 1, the differences between the measured distances and the ground truth distances are less than 20 cm. The distance of tag 1 and tag 4 and the distance of tag 2 and tag 5 are very close to the ground truth. One potential reason can be that the tag 3 or tag 6 is at a position with relatively lower resolution. As shown in Figure 3.16, there are areas in yellow-green surrounded by green. The positions of these two tags are close to the yellow-green area. In scenario 2 and scenario 3, some sensors were occluded. The results of these two scenarios show that the positioning error increases when more sensors are occluded.

In scenario 4, 5 and 6, only tag 4, tag 5 and tag 6 are moved. After moving the tags, the ground truth distance is still 1.52m. As shown in Table 3-1, the position changes of tag 4, tag 5 and tag 6 do not generate errors to the captured positions of tag 1, tag 2 and tag 3. After moving the three tags to the area with poor positioning performance, the difference between the measured distance and ground truth increased. The positions of tags changed when more obstacles are added. However, the results show that the difference is decreasing when adding more occlusions to the sensors. One potential reason is that the errors of tag 4, tag 5 and tag 6 generated by the occlusions have compensated the errors caused by moving the tags to the bad location. This assumption is made because the positions of tag 1, tag 2 and tag 3 in scenario 5 and scenario 6 are close to that in scenario 2 and scenario 3.

Table 3-1 Average coordinate data

Tag	Scenario (1)			Scenario (2)			Scenario (3)		
	x (m)	y (m)	z (m)	x (m)	y (m)	z (m)	x (m)	y (m)	z (m)
T1	5.916	3.445	0.682	5.947	3.469	0.675	5.946	3.457	0.637
T2	6.006	3.452	0.464	5.996	3.451	0.483	6.016	3.456	0.453
T3	5.865	3.375	0.625	5.847	3.355	0.592	5.882	3.355	0.559
T4	4.418	3.489	0.543	4.373	3.492	0.535	4.382	3.479	0.486
T5	4.466	3.429	0.419	4.453	3.410	0.388	4.543	3.417	0.243
T6	4.211	3.391	0.181	4.137	3.407	0.001	4.126	3.355	0.053
Tag	Scenario (4)			Scenario (5)			Scenario (6)		
	x (m)	y (m)	z (m)	x (m)	y (m)	z (m)	x (m)	y (m)	z (m)
T1	5.937	3.457	0.669	5.949	3.459	0.667	5.955	3.459	0.636
T2	5.993	3.475	0.470	6.006	3.447	0.455	6.047	3.439	0.482
T3	5.883	3.373	0.583	5.867	3.350	0.592	5.920	3.344	0.621
T4	7.795	3.296	0.606	7.789	3.279	0.571	7.769	3.304	0.551
T5	7.813	3.179	0.574	7.803	3.193	0.572	7.798	3.205	0.588
T6	7.657	3.411	0.381	7.636	3.366	0.339	7.667	3.379	0.364

Table 3-2 Standard deviation

Tag	Scenario (1)			Scenario (2)			Scenario (3)		
	x	y	z	x	y	z	x	y	z
T1	0.025	0.012	0.010	0.014	0.009	0.008	0.019	0.012	0.017
T2	0.012	0.020	0.014	0.015	0.031	0.021	0.017	0.031	0.041
T3	0.018	0.013	0.020	0.074	0.013	0.027	0.020	0.015	0.019
T4	0.015	0.012	0.010	0.010	0.011	0.016	0.015	0.014	0.020
T5	0.032	0.022	0.036	0.034	0.026	0.043	0.019	0.040	0.045
T6	0.031	0.041	0.043	0.021	0.020	0.004	0.015	0.016	0.025
Tag	Scenario (4)			Scenario (5)			Scenario (6)		
	x	y	z	x	y	z	x	y	z
T1	0.019	0.009	0.010	0.019	0.012	0.016	0.021	0.020	0.026
T2	0.017	0.032	0.020	0.021	0.041	0.030	0.018	0.033	0.019
T3	0.024	0.013	0.022	0.022	0.009	0.020	0.019	0.012	0.030
T4	0.014	0.023	0.029	0.017	0.022	0.031	0.012	0.018	0.031
T5	0.022	0.029	0.025	0.020	0.028	0.023	0.019	0.024	0.021
T6	0.010	0.019	0.047	0.030	0.040	0.085	0.018	0.024	0.043

Table 3-3 Tags distance

Tags	Scenario (1)		Scenario (2)		Scenario (3)		Scenario (4)		Scenario (5)		Scenario (6)	
	D (m)	ΔD (m)	D (m)	ΔD (m)	D (m)	ΔD (m)	D (m)	ΔD (m)	D (m)	ΔD (m)	D (m)	ΔD (m)
Tag 1 & Tag 4	1.499	0.021	1.573	0.053	1.564	0.044	1.866	0.346	1.849	0.329	1.821	0.301
Tag 2 & Tag 5	1.541	0.021	1.544	0.024	1.474	0.046	1.845	0.325	1.816	0.296	1.767	0.247
Tag 3 & Tag 6	1.654	0.134	1.711	0.191	1.757	0.237	1.775	0.255	1.770	0.250	1.747	0.227
	∑(ΔD)	0.176	∑(ΔD)	0.268	∑(ΔD)	0.327	∑(ΔD)	0.926	∑(ΔD)	0.875	∑(ΔD)	0.775

Note: ΔD = |D - ground truth|

3.8.3 Conclusions

The results show that the BLE RTLS based on AOA is able to make sub-meter positioning in a confined indoor area. The positioning accuracy changes if the tags are in the area with low positioning resolution or if any sensor is occluded. The impact of moving tags to a bad location is more significant than the impact of occluding sensors.

3.9 Summary

This chapter proposed a methodology of providing safety alerts to construction workers using BLE RTLS based on AOA. The requirements of using BLE RTLS based on AOA are listed. The solutions to meet the requirements are explained. A wireless scheme for the installation of the RTLS in a large construction site is presented. With this scheme, the cabling problems of applying RTLS can be mitigated. The proposed proximity alerts generating system has two modules: the data processing module, the proximity detection module. In the data processing module, the data captured by the RTLS are processed to estimate the position and velocity of workers and equipment. Then, with the information provided by the data processing module, the proximity detection module detects proximities and generate alerts. For reducing false alerts, the proximities of workers and equipment in the same team are ignored. The movement of an entity is also considered when detecting unsafe proximities. Three cases of proximity can be identified. Each case has its corresponding alert so the workers and operators who receive the alerts can perceive the level of danger. The system also recognizes and skips repeated alerts within a period of time. A prototype system is developed based on the methodology. The development involves a web-based software and hardware for generating alerts. All data from the RTLS and events from the proximity detection module are saved in a local hard disk and uploaded to the cloud database.

A laboratory test was conducted to test BLE RTLS being used in this research. The RTLS is able to provide sub-meter accuracy positioning in confined indoor space. Moving tags to an area with low positioning resolution has an impact on the positioning accuracy. In addition, occlusions can also affect the performance of the positioning.

Chapter 4: Implementation and Case Studies

4.1 Introduction

This chapter presents the case study of applying BLE RTLS and the proposed method on a construction site. The site is an electric substation with dimensions of 110 m × 70 m. According to the test result in the lab, the electric magnetic noise has no significant effect on the positioning performance of the BLE RTLS (Appendix A). As shown in Figure 4.1, elements of the substation are built on one side of the construction site. The other side of the site is reserved for material storage and a workshop to prepare elements. The office where the server computer is located about 150 m away from the site. Because of the long-distance, using ethernet cables for connecting sensors with the server computer is difficult. In this case, a wireless scheme is used to establish connections. Two tests are conducted to evaluate the feasibility and performance of the RTLS using the wireless scheme in this construction site. Besides, the performance of proximity detection is tested during construction activities. The system detects the proximities between five volunteer workers and three pieces of equipment. At last, the alert generation function is evaluated by two tests and a field trial.

Figure 4.1 Electric substation

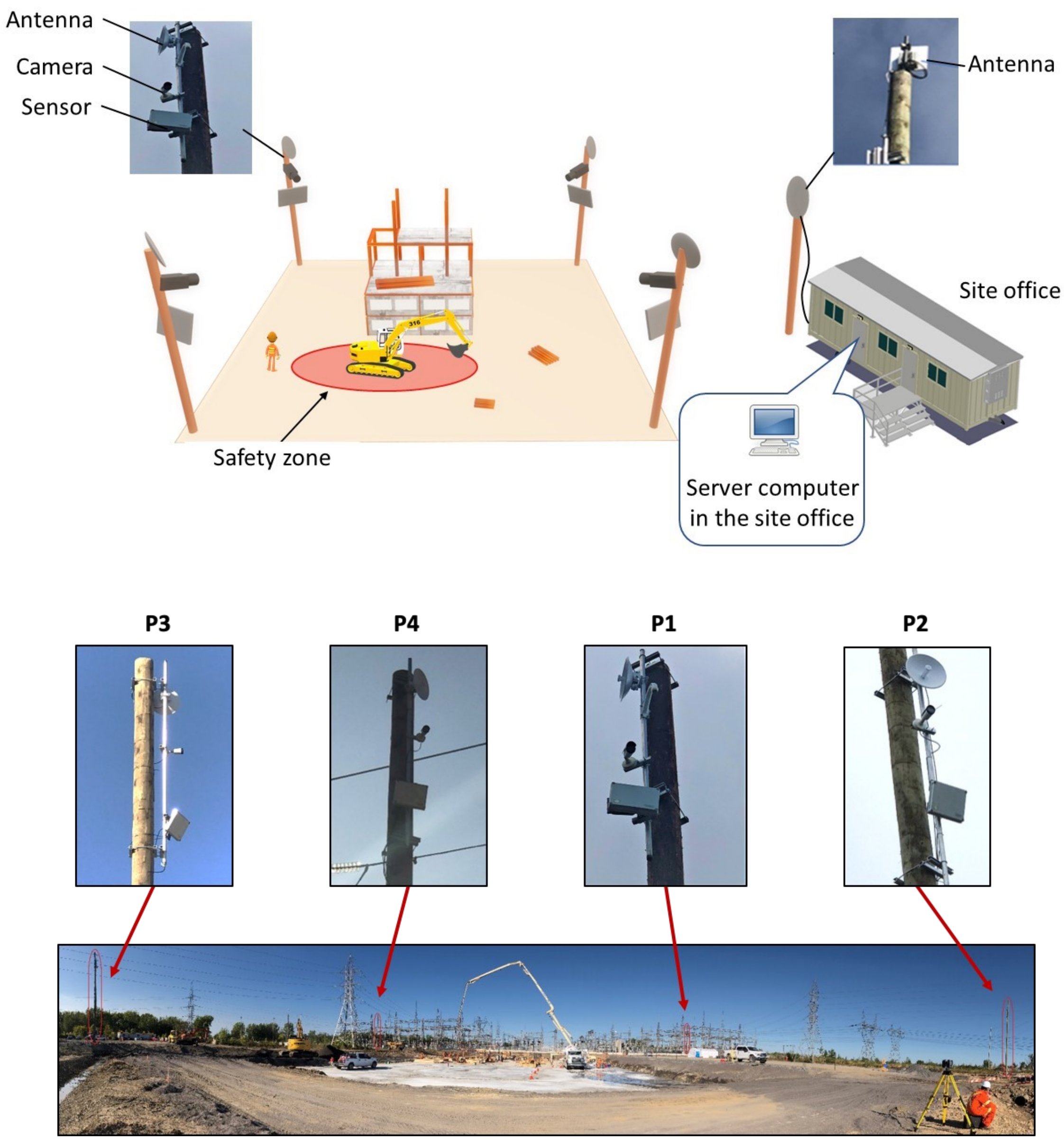


Figure 4.2 Installation of Quuppa sensors and antennas

4.2 RTLS Setup

4.2.1 Installation

The deployment of the RTLS has the following difficulties: (1) The office is far away from the site. Building the network environment for the sensors and server computer using ethernet cables is difficult and expensive; (2) It is difficult to ensure coverage of sensors in a site with large dimensions.

In this research, antennas are used to build the network connection between sensors and the server computer in the site office. As shown in Figure 4.2, four poles are installed near the corners of the site. The RTLS sensors and the antennas are installed on the four poles. In addition, a camera is installed on each pole to capture the videos of the site, which will be used later to compare with the results from the system. Another pole with an antenna on its top is installed near the office. This antenna is connected to the network of the office by ethernet cable.

The model of sensors being used in this case study is Quuppa LD-7L, which is designed for outdoor and is claimed to have the capability of providing sub-meter accuracy tracking within 100 m. In order to ensure the RTLS coverage on the construction site, nine sensors in total are installed on the poles. Sensors on the same pole are installed at different heights and with different orientations to optimize the coverage. Additionally, different poles are deployed with different numbers of sensors as shown in Table 4-1. The distribution of sensors is defined according to the tasks on the construction site. On the poles near the main construction area, more sensors are deployed. A total station was used to do the surveying for the sensors. The positions of the sensors are shown in Table 4-1. Positions given by the total station are in Northings and Eastings Coordinate (NEZ). To make the positions usable in the Quuppa RTLS, the positions are converted into a Cartesian Coordinate (XYZ), where the coordinate of the highest sensors on pole 2 is (0,0,11.469).

Table 4-1 Position of sensors

Pole	Sensor	Northings and Easting Coordinate			Cartesian Coordinate		
		N (m)	E (m)	Z (m)	X (m)	Y (m)	Z (m)
1	S1	310684.32	5032701.96	34.12	-39.55	-103.82	13.31
2	S2	310723.87	5032805.78	32.28	0.00	0.00	11.47
	S3	310724.12	5032805.51	28.09	0.25	-0.26	7.28
3	S4	310587.66	5032746.60	32.50	-136.21	-59.17	11.69
	S5	310587.38	5032746.76	27.51	-136.55	-59.02	6.70
	S6	310587.31	5032746.74	25.88	-136.57	-59.04	5.07
4	S7	310604.60	5032840.33	32.47	-119.27	34.56	11.66
	S8	310604.34	5032840.21	27.08	-119.53	34.44	6.27
	S9	310604.41	5032840.22	25.69	-119.47	34.44	4.88

4.2.2 System Setup

In this part, the RTLS setting is explained. As shown in Figure 4.3, the 'proprietary mode' is chosen as the operation mode of the system. The tracking area is set to 'open', which means the system is configured to do positioning in a large open space. For higher accuracy on the x-y plane, 2D tracking was chosen. The tracking height is set to 1.8 m. The positions of the four poles generate a cell, which is the tracking area. The system can track the position of tags whenever they enter the tracking area. Then nine sensors were added according to their locations.

The next step is identifying the sensors. Another portable sensor named 'focusing locator' is set up using the wireless deployment scheme as shown in Figure 4.4 (Quuppa 2019c). The wireless scheme could successfully identify all the sensors and the quality of the signals was good. Then, the focusing locator is also used for capturing the orientations of the sensors. The focusing locator is placed at a random location inside the tracking area, pointing at one of the sensors to be configured. When the system informs that the measurement is acceptable, the position of the focusing locator measured by the total station is input into the system to calculate the orientation of the sensor. Then, by moving the focusing locator and observing the movement of the focusing locator in the system, the orientation of the sensor can be verified. The focusing locator is then moved to a different location for another measurement of the same sensor to increase the reliability of the measured orientation. In this test, at least five measurements were made for each sensor. Figure 4.5 shows the representation of azimuth, elevation and rotation.

The Quuppa system provides a function to estimate the performance of the system in the tracking area. The estimation result of this setting is shown in Figure 4.6. The green area represents the area where the positioning quality is good, while the yellow area represents acceptable positioning quality. For the red area, the positioning performance is poor and not reliable. From the estimation of the setting in this case study, the performance was estimated to be good inside the cell formed by the four poles.

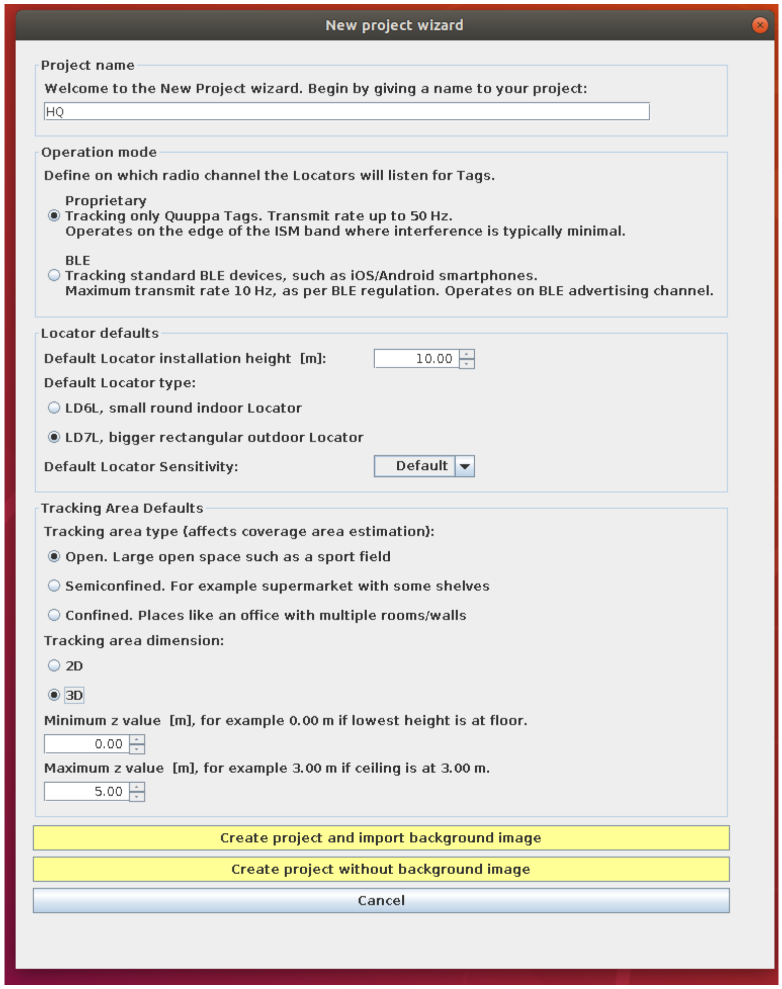


Figure 4.3 Quuppa site planner project setting

Figure 4.4 Wireless deployment scheme (Quuppa 2019c)

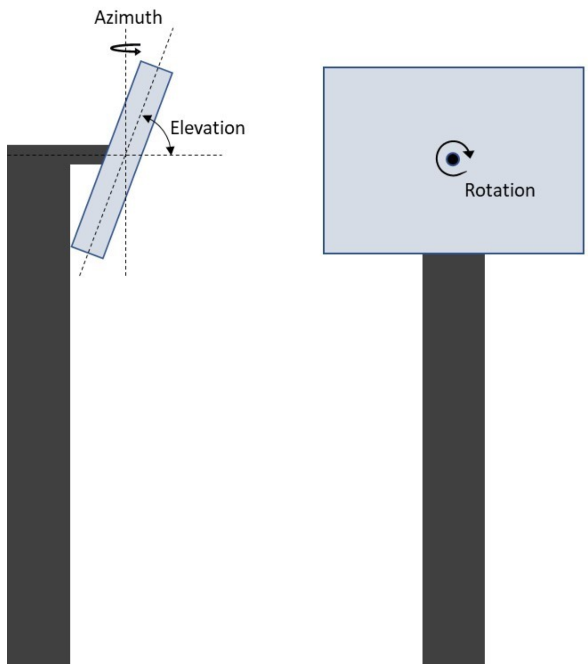


Figure 4.5 Representation of azimuth, elevation and rotation

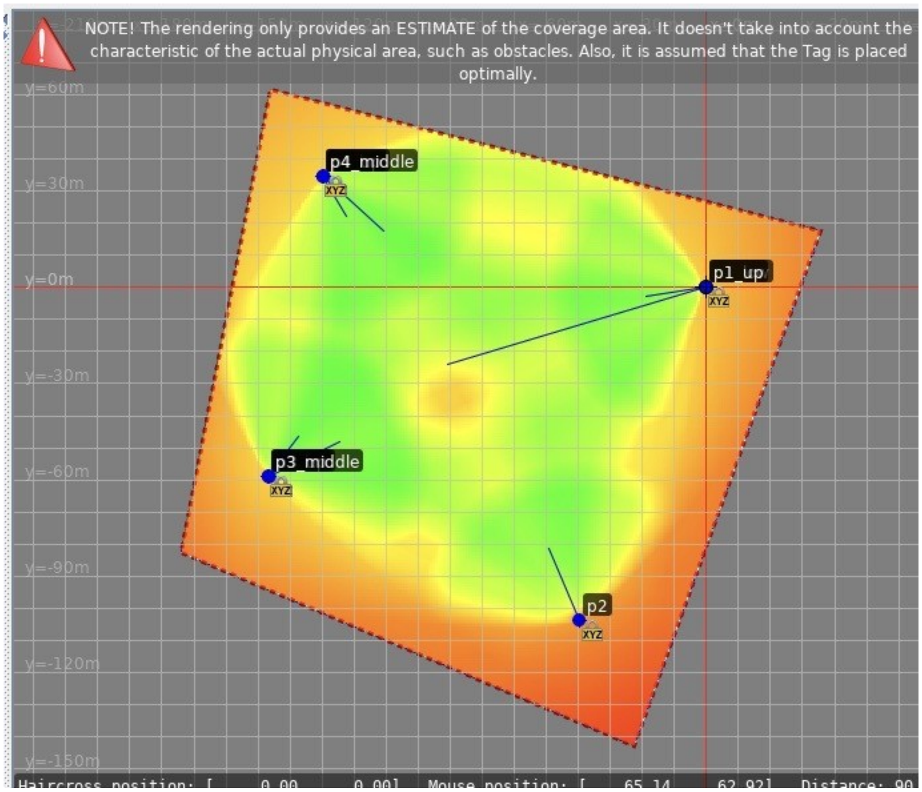


Figure 4.6 Quality estimation of the system

4.3 RTLS Accuracy Test

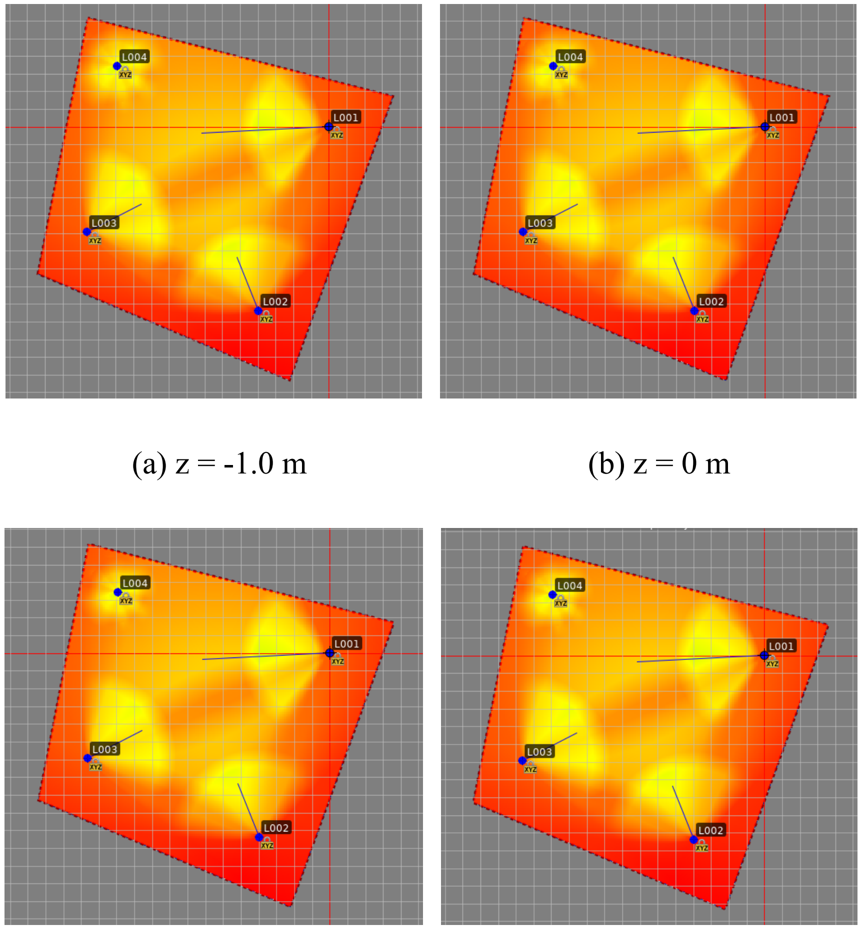
4.3.1 Preliminary Test

4.3.1.1 System Setting and Test Design

A preliminary test was conducted to test the feasibility of applying BLE RTLS on this construction site. The test was conducted on October 2nd, 2019. Four sensors were installed at similar heights on the poles. The positions of the sensors are shown in Table 4-2. The system setting was the same as the setting in Section 4.2, except for the tracking mode. In the preliminary test, the 3D tracking mode was chosen. Since the sensor on pole 4 was not connected properly, it was not deployed during the test. The estimation of the positioning quality at different heights is shown in Figure 4.7.

Table 4-2 Position of sensors in the preliminary test

Pole	Sensor	position captured by total station			transferred data		
		X (m)	Y (m)	Z (m)	X (m)	Y (m)	Z (m)
P1	Sensor 1	310723.91	5032805.86	32.26	0.00	0.00	11.47
P2	Sensor 2	310684.21	5032701.90	34.12	-39.70	-103.96	13.33
P3	Sensor 3	310587.64	5032746.61	32.51	-136.27	-59.25	11.72
P4	Sensor 4	310604.64	5032840.24	32.85	-119.27	34.37	12.06
Average height of ground (m)				20.79			



(a) z = -1.0 m (b) z = 0 m

(c) z = 1.0 m (b) z = 1.5 m

Figure 4.7 Estimation at difference height (preliminary test)

The tags in this test were configured to have 5 Hz transmission rate and high transmission power. After the activation of the BLE RTLS, three site workers wore the hardhats attached with two tags on the sides and the wristbands, which are fully compatible with Quuppa system. The Quuppa was able to capture the position of tags and wristbands. In addition, the user interface of the RTLS can show the position of all entities.

A simple test was made to evaluate the accuracy of the system. In this test, a tag was placed on an iron bar. The position of the tag measured by a total station was considered as the ground truth. Then, the RTLS recorded the position of the tag for one minute. By comparing the position in the duration with the ground truth, the accuracy of the system can be evaluated. The test was conducted twice at different positions.

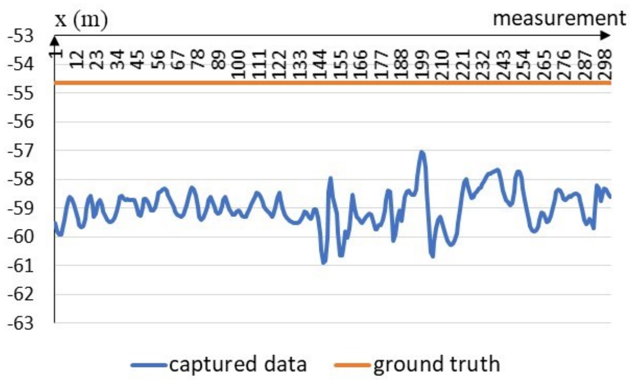
4.3.1.2 Results and Discussion of Preliminary Test

When the tag was at position 1 (P1), 298 measurements were captured by the RTLS for one minute. The ground truth positions of the tag in position 1 and position 2 are shown in Table 4-3. Figure 4.8 shows the comparison between the ground truth and the measured position from the RTLS in the x, y and z directions. The position measured by the RTLS has a sinificant shift compared to the ground truth. Overall, the measured position is 5.02 m away from the ground truth (Table 4-4). The shift in the x-direction is the most significant with a distance of 4.34 m, while the average shifts in the y-direction and z-direction are 1.26 m and 2.10 m, respectively. It is also necessary to note that the ground truth in the z-direction is above 0, but most of the z values captured by the RTLS were below 0. In addition, the measurements in the x-direction have strong fluctuations than those in the other two directions.

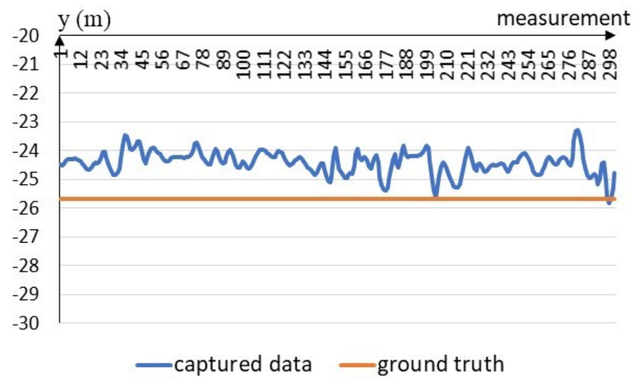
For the results when the tag was at position 2, also 298 measurements were captured by the RTLS. As shown in Figure 4.9, the shifts and the fluctuations in x, y and z directions are smaller. In general, the average position shift is 1.23 m (Table 4-5). The measurements in x-direction still have the most errors. In the y and z directions, the measured data are fluctuating near the ground truth. The average differences in y and z directions are less than 0.3 m.

Table 4-3 Ground truth of position 1 and position 2

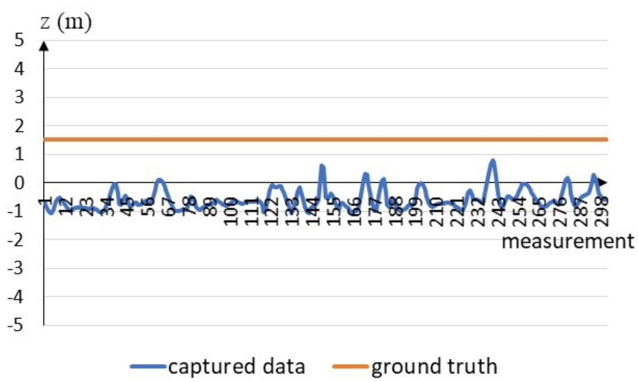
	x_g (m)	y_g (m)	z_g (m)
Position 1	-54.65	-25.67	1.52
Position 2	-57.99	-33.94	1.39



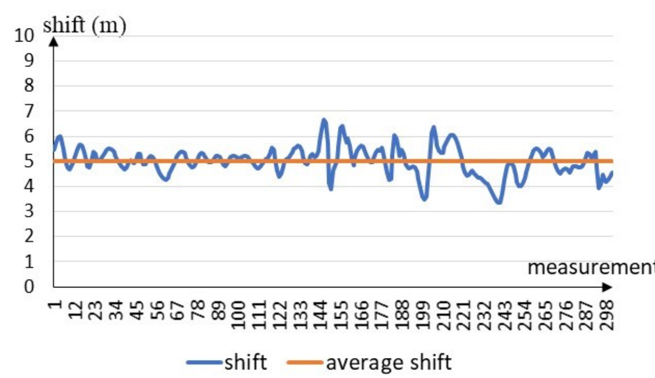
(a) x



(b) y



(c) z

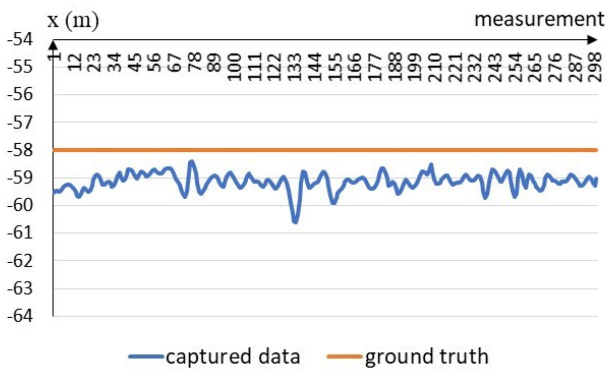


(b) position shift

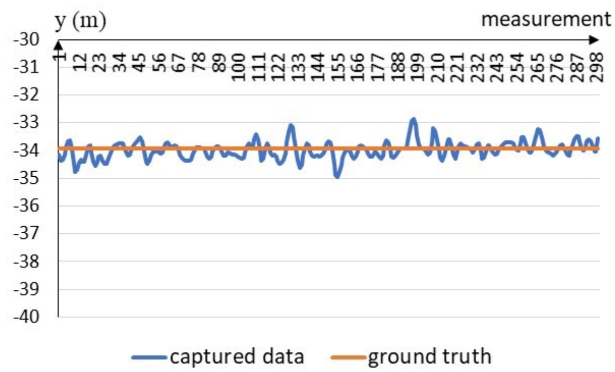
Figure 4.8 Results compared with ground truth (position 1)

Table 4-4 Results compared with ground truth (position 1)

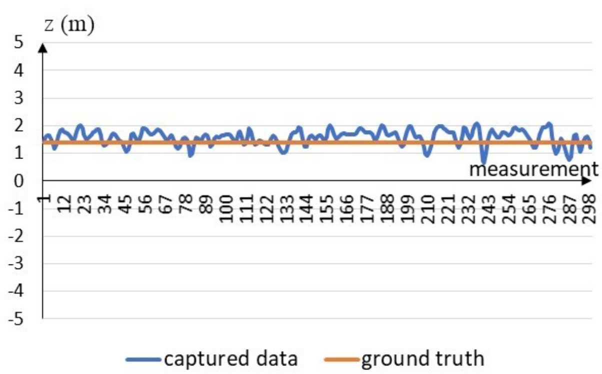
	x	y	z
Average position (m)	-58.99	-24.41	-0.58
Standard deviation	0.63	0.39	0.35
	$\lvert x - x_g \rvert$	$\lvert y - y_g \rvert$	$\lvert z - z_g \rvert$
Difference (m)	4.34	1.26	2.10
Standard deviation	0.63	0.39	0.35
Average position shift	5.02		



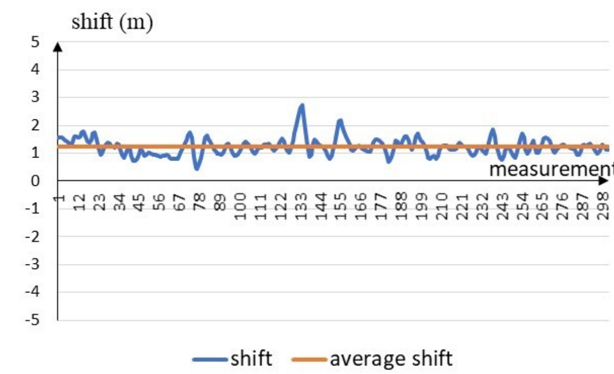
(a) x



(b) y



(c) z



(b) position shift

Figure 4.9 Results compared with ground truth (position 2)

Table 4-5 Results compared with ground truth (position 2)

	x	y	z
Average position (m)	-59.14	-33.98	1.57
Standard deviation	0.30	0.31	0.27
	$\lvert x - x_g \rvert$	$\lvert y - y_g \rvert$	$\lvert z - z_g \rvert$
Difference (m)	1.15	0.24	0.28
Standard deviation	0.30	0.20	0.17
Average position shift	1.23		

The system can detect all the tags and wristbands on the site. However, the system did not provide good positioning performance during the test. The test cannot achieve sub-meter accuracy. The potential reasons that account for this are discussed:

(1) The insufficient number of sensors. As shown in Figure 4.7, most of the area in the construction site is colored in orange and some area is in red or yellow. The three sensors are not capable of covering the whole site and provide accurate positioning. The author attempted to change the orientation of the sensors in the system manually to optimize the positioning quality in the estimation. But even in the best case, only a small area is colored in green. For better positioning quality, more sensors are required.
(2) The 3D tracking mode. In the preliminary test, the 3D tracking mode was chosen. The accuracy in the x-y plane for 3D tracking mode is lower than that for 2D accuracy. Therefore, in the later tests, the 2D tracking mode is applied.
(3) Location of the tag. The accuracy of the same tag at the two positions has a significant difference. This could be because the positioning performance in the two different locations is different.
(4) Metal materials. During the test, the tag was attached directly to a metallic surface. The metallic surface can reflect the BLE signals from the tag. The reflected signals can affect the positioning accuracy.

After discussions with the Quuppa technicians about the case and the results, the following modifications were made. More sensors are deployed. For sensors on the same pole, their orientations are different. 2D tracking mode was applied for higher accuracy on the x-y plane. The setup of the RTLS after the modifications has been presented in Section 4.2.

4.3.2 Accuracy Test with New Setup

An initial test was conducted to evaluate the performance of the RTLS with the new setting. The test was conducted on December 12th, 2019. The test compared the RTLS location data with the ground truth, which was provided by a total station. In this test, the accuracy and missing data rate are analyzed to evaluate the performance.

The setting of the tags is shown in Figure 4.10. Tags in this test were configured to have 33 Hz transmission rate and high transmission power. The 33 Hz is defined by the data handling capacity of the sensors. The data handling capacity of the sensors in this research is 150 packet/s. Four tags were activated during the test. According to Eq. (3-1), the maximum update rate of the tags is 37 Hz. The RTLS in this research provides the option of 33Hz, which is the closest value to 37 Hz.

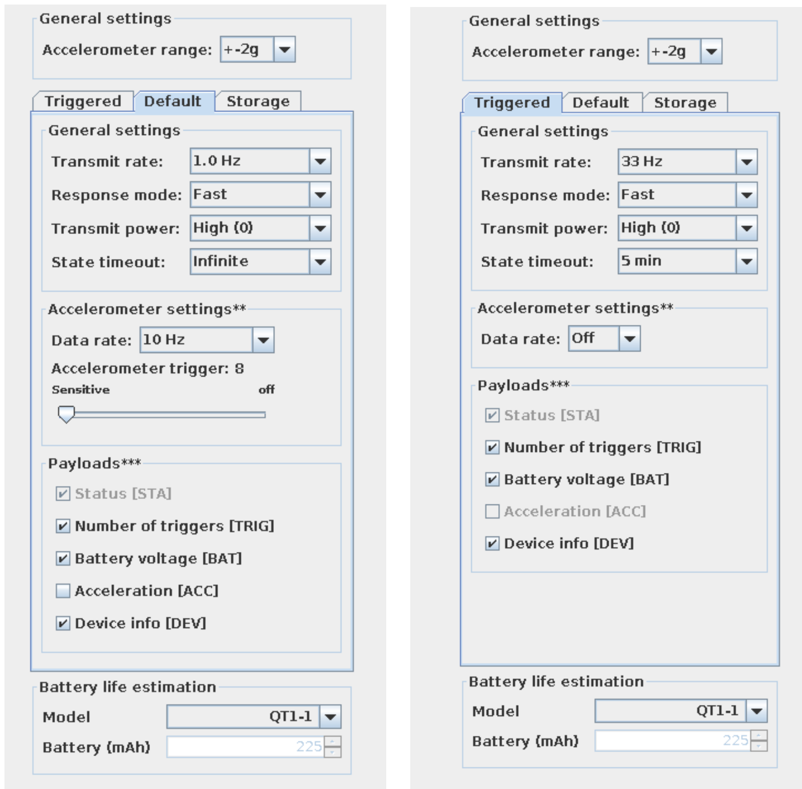


Figure 4.10 Setup of tags in triggered mode and default mode

4.3.2.1 Test Design

Two tags are attached to the hardhat, as shown in Figure 3.3 (a). Another two tags are attached to the top surface of a vehicle. As shown in Figure 4.11, the worker and the vehicle are asked to move across the site parallelly at a similar speed. When they arrive at the four checkpoints, they are asked to stop. Then the surveyor uses a total station to capture the positions of the worker and the vehicle as ground truth. The positions calculated by the system are compared with the ground truth to evaluate the accuracy.

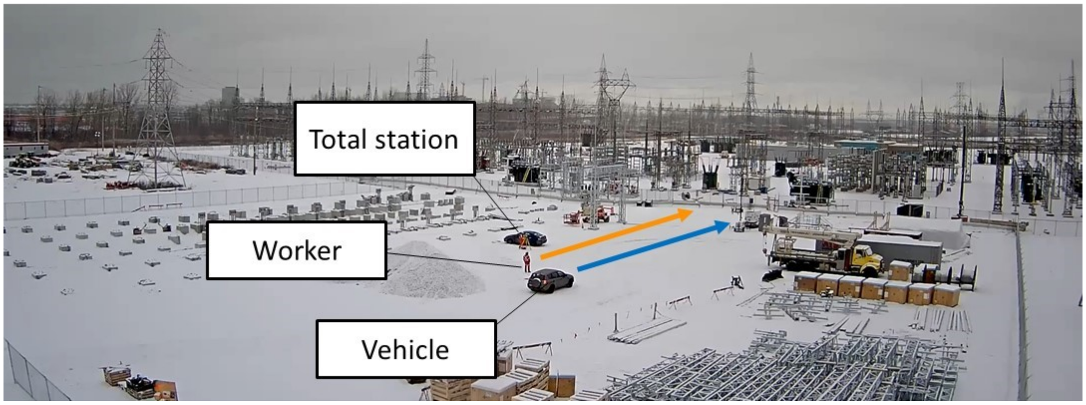


Figure 4.11 Initial test

4.3.2.2 Results and Analysis

The test started at 13:01:20 and ended at 13:06:30. The data in the 310 s duration are analyzed, including the missing data rate and the accuracy are analyzed. The average accuracy of tags attached to the hard hat was 2.15 m, while the average accuracy of tags attached to the vehicle was 2.31 m. It is worthy to mention that the accuracy is an indicator provided by the RTLS to evaluate the performance in positioning the specific tags.

As mentioned in Section 2.2.2, the UWB RTLS wireless scheme leads to a higher missing data rate (MDR). In this research, the MDR is considered to evaluate the ability of the RTLS to collect data on this construction site. The MDR is calculated according to Eq. (2-7).

In the proposed method, all the data of a tag in one second are averaged to represent its position. If the RTLS cannot capture any data of a tag in the second, the tag is considered to be missing. The estimation of the pose of the entity, in this case, should rely on the other tags attached to the same entity. The rate when the system misses a tag in a full second can reflect the reliability of the results. In the data analysis, the missing seconds are defined as the total seconds that the RTLS captures no data from a tag during the period. The missing second rate is calculated according to Eq. (4-1).

$$MSR = \left(1 - \frac{Missing\ seconds}{Total\ duration}\right) \times 100\% \qquad (4\text{-}1)$$

Table 4-6 shows the MDR and MSR of all four tags during the test. Tag 1 and tag 2 are the tags attached to the vehicle, while tag 3 and tag 4 are attached to the hardhat. It can be observed that the performance in collecting data is reasonable. For the tags attached to the vehicle, more missing data are found. The missing second rates for the tags are zero, which means that the RTLS can capture the data of the four tags every second during the test.

Table 4-6 MDR and MSR analysis

	Tag 1	Tag 2	Tag 3	Tag 4
Total Duration (s)	310	310	310	310
Expected Update Rate (Hz)	33	33	33	33
Total Readings	8381	8384	8774	8782
Missing Data Rate	10.57%	10.54%	6.38%	6.30%
Missing Second Rate	0	0	0	0

Figure 4.12 shows the estimated positions of the worker and the vehicle compared with the ground truth. The measured movement of the worker is close to the ground truth, while the movement of the vehicle has shifted to its left. At checkpoints 3 and 4, the position of the vehicle has shifted to the position near the worker.

To make a more detailed analysis, the seconds when the vehicle and worker stopped at the checkpoints are picked out. The distances between the entities and their ground truth are calculated to evaluate the accuracy. The accuracy is shown in Figure 4.13. For the worker, the accuracy is better, as 95.97 % of data from the tags on hardhat have less than 2 m accuracy. For the result of the vehicle, only 47.66 % of data have an error within 2 m.

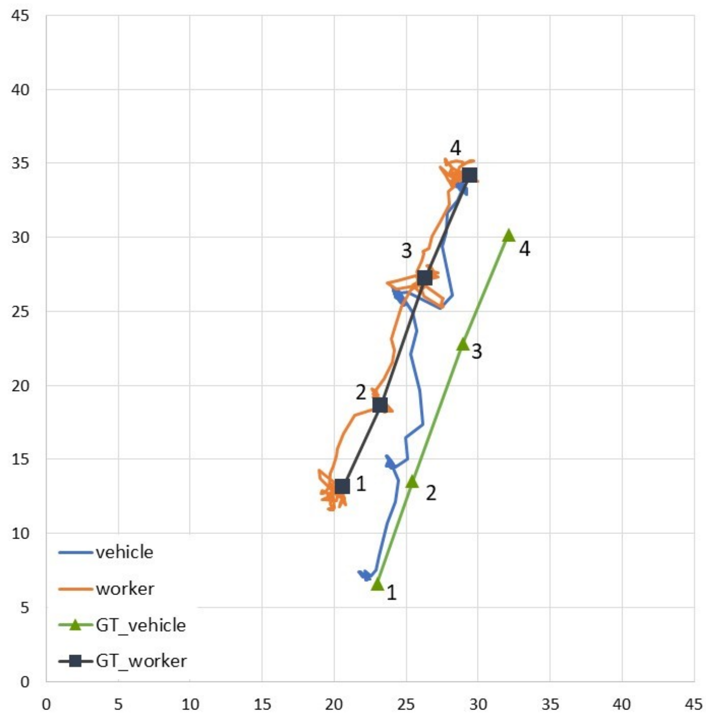


Figure 4.12 Averaged position compared with ground truth

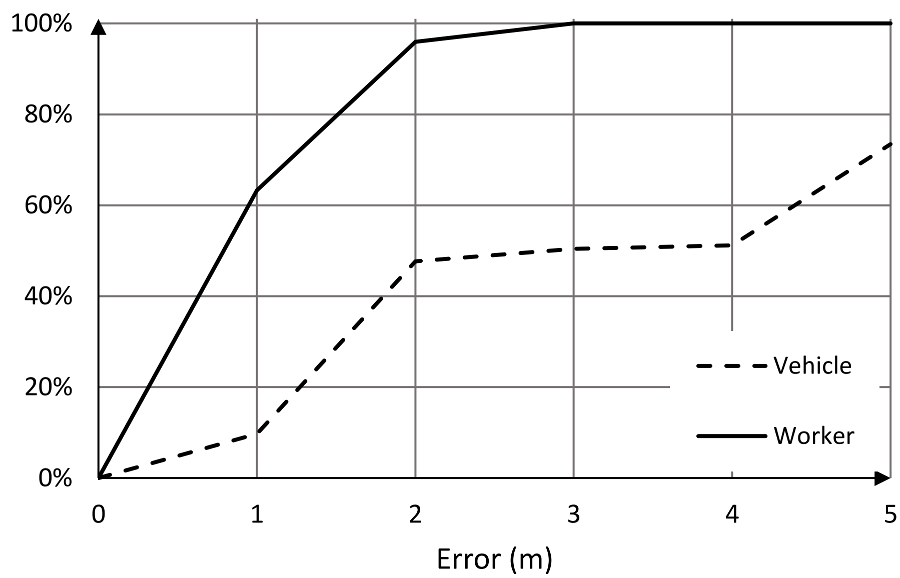


Figure 4.13 Accuracy analysis result (initial test)

Overall, the results show that the system has better performance when tracking the position of workers. The missing data rate is lower and the accuracy is reasonable. Since the worker and the equipment are near each other, and the tags are also placed at similar heights, the differences in the tags' positions and heights are not the main reasons causing errors in the measurement of the vehicle. As mentioned in Section 3.1, metallic material has significantly affected the positioining accuracy. One assumption for the errors is that the surface of the vehicle reflected the Bluetooth signals since the tags were attached directly to the metallic surface. To avoid errors caused by attaching tags directly to the metallic surface, sponge pads were placed between the tags and the equipment in the later tests.

4.3.3 Accuracy Test with Built-Up Metal Structures

Another test was conducted on January 22th to test the performance of the system when more steel structures have been built on the site. The setting of the system was the same. However, the changes in the construction site gave more challenges to the performance of the system. As shown in Figure 4.14, steel structures have been built on the main construction area. In addition, more metal materials have been placed in the zone reserved for materials storage on the site.

Figure 4.14 View of the site on January 28th, 2020

4.3.3.1 Test Design

There are two phases in this test. The first phase was conducted to see the performance of the RTLS when the metal structures exist on the site. In the first phase, two test participants were asked to wear the hardhat attached with tags and walk parallelly in the area where metal structures were built. The second phase was similar to the test on December 16th, 2019. Two participants were asked to attached tags on their hardhat. Three tags were attached to the top surface of a vehicle. To see whether the performance of BLE RTLS is affected when tags are attached directly to the metal material, two tags were attached to the vehicle with a sponge pad in between, while the third tag was attached directly to the metal surface of the vehicle. The two participants and the vehicle were asked to move parallelly across the site and stop when they arrived at the three checkpoints. A total station was used to capture the location of the two workers and the vehicle when they arrived at the checkpoints.

4.3.3.2 Results and Analysis

The phase 1 of the test started at 10:46:30 and ended at 10:59:00. Data during the 750 seconds were collected. The MDR and MSR are analyzed, as shown in Table 4-7. The MDRs of all the tags attached to the hardhats are between 30% and 40%, and the MSRs are between 6% to 10%. Compared to the previous test, more missing data can be found. Although the MDR and MSR have increased significantly, the system can still estimate the position of the workers. The movement paths of the two workers estimated by the system are shown in Figure 4.15. The paths are compared with the videos of the site. It can be observed that the paths estimated by the system are reasonable. When the workers were near the center of the main construction area, the fluctuations of the paths were more violent. On the other hand, when the worker approached the edges of the construction area, where noises from steel materials are less, the path is smoother.

Table 4-7 Data missing rate in phase 1

Entities	Worker 1		Worker 2	
Tag	W_1_L	W_1_R	W_2_L	W_2_R
MDR	39.16%	34.76%	35.94%	32.15%
MSR	9.33%	6.53%	7.07%	6.00%

Figure 4.15 Result of phase 1

The phase 2, which repeated the steps of the test in December, was conducted for two purposes. First, to evaluate how much errors the metal structures have brought to the accuracy of the system. Second, to see whether adding sponge between tags and metallic surface improves the locating performance.

The phase 2 started at 13:40:10 and ended at 13:46:40. Data in the 390 seconds duration were collected. The missing data rate and the missing second rate are shown in Table 4-8. The missing data rate and missing second rate are higher. Among all, the tag 'V2' attached directly to the metallic surface has the highest MDR. The tag 'V3' on the vehicle contributes to the lowest MDR and MSR. This proves that adding a sponge pad between the tag and the metallic surface can help to improve the performance of the RTLS.

Table 4-8 MDR and MSR analysis in phase 2

Entity	Worker 1		Worker 2		Vehicle		
Tag	L	R	L	R	V1	V2 (no pad)	V3
MDR	46.53%	39.26%	46.00%	35.05%	47.05%	61.56%	34.21%
MSR	11.63%	7.44%	14.65%	6.74%	8.60%	16.74%	3.72%

The accuracy of the system is affected by the metal structures and materials on the site. The accuracy is shown in Figure 4.16. For worker 1, accuracy is the highest. However, only 70% of the data of worker 1 have less than 5 m accuracy. For worker 2, the accuracy is similar to that for the vehicle, with 45% of data have an error within 5 m.

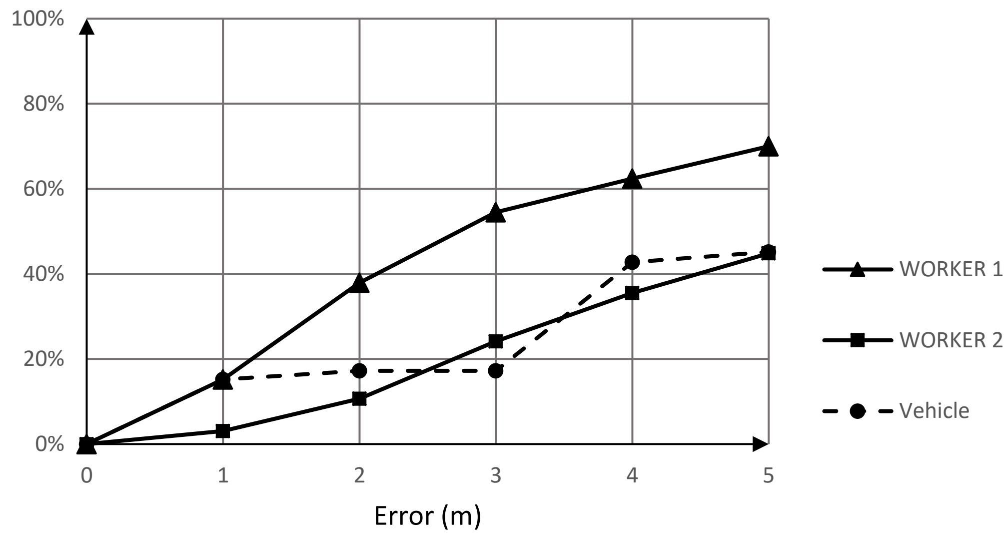


Figure 4.16 Accuracy analysis result (second test)

4.3.4 Factors Leading to Errors

The results of the previous tests show that the accuracy of the RTLS is not as expected. The following factors can be the reasons:

(1) Metal materials on construction site: The presence of more metal materials on the construction site has caused more Bluetooth signal reflections, which lead to more errors in positioning. However, there is no effective solution to reduce the errors generated by the radio signal reflections from metal materials for the radio-based RTLS. The only way is to avoid using the radio-based

RTLS when the site has a large quantity of metal materials (e.g., using the RTLS during excavation and foundation building phase).

(2) Occlusions by built-up structures: The occlusion is another main cause of errors. The sensors were installed with a height of about 10 m, while the structures were about 8 m to 9 m high. Therefore, the structures became obstacles between the sensors on the pole and the tags. To overcome this problem, one solution is to build a pole near the center of the construction site with sensors having different orientations on the pole. Another solution is adding more poles and deploying more sensors near the edges of the site. A good plan for the distribution of sensors can also mitigated the occlusion problems from the built-up structures. Besides, there are solutions that were proposed recently which can deal with the non-line-of-sight situations using only RTLS (Ansaripour et al., 2020). By applying these methods, it is possible to mitigate the impacts from occlusions.

(3) Not enough density of sensors: In this research, nine sensors were distributed on the four poles near the corners of the construction site. However, deploying nine sensors on four poles is not enough for providing best positioning performance in this construction site. As shown in Figure 4.6, there are areas which have bad or medium estimated accuracy. According to the Quuppa technicians, to make a more accurate positioning, the sensors should be deployed at more different locations on the site to increase the coverage of the RTLS (i.e., increasing the density of sensors).

(4) Attaching tags to the metallic surface: Although a sponge pad was added between the tags and metallic surfaces, the errors caused by attaching tags to metallic surfaces are not fully eliminated. It is required to design a better scheme for attaching the tags on the equipment surface to reduce the errors, which will be one of the future works.

(5) Error caused by wireless communication: In this research, antennas are used to connect the sensors and cameras with the server computer. When analyzing the video and the RTLS data, there are lost data. One potential reason is that some data packets are lost during wireless communications. To better understand how and why the data are missed, it is required to take another test to monitor the data flow during the data communication. After knowing the causes of the missing data, a more stable wireless connection scheme can be developed (e.g., finding a better antenna model).

(6) Limited capacity of the RTLS in handling tags' data: The update rate of tags was set to 5 Hz, and the prototype system detects proximity every second. The low frequency of data sampling limits the efficiency of averaging data over time. Besides, the system's low update rate may lead to an up to one-second delay in generating alerts.

4.4 Virtual Test

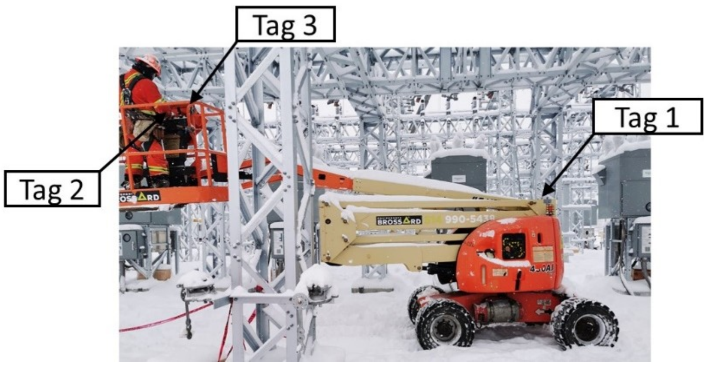
Before deploying the system on the construction site, a virtual test was conducted to test the functions of the prototype system. A virtual API is developed using Tomcat 8 and Python. This virtual API can simulate the API of the BLE RTLS and automatically generate data for tags of three workers and three pieces of equipment. The output format is the same as the API of the BLE RTLS. The movements of the workers and equipment are randomly generated considering the OC.

In this test, the prototype processes the tags data from the virtual API and detect proximities. The program was run for 10 minutes and 22 proximities were detected. The prototype system could identify the proximities. Among all proximities, 15 proximities were Case 2, while six proximities were Case 3 and one proximity was Case 1. Compared with the raw data recorded from the virtual API, the velocities and distance were very close. The case of the proximities can be identified correctly.

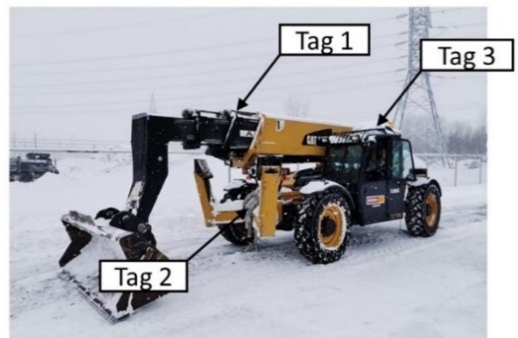
4.5 Proximity Detection in Real Construction Activities

Five anonymous volunteer workers joined the test for proximity detection. They were asked to attach tags on the sides of their hardhats, as shown in Figure 4.17. This test focuses on the proximity detection between the workers and the three kinds of equipment, which are boom lift (equipment 1), telehandlers (equipment 2), and crane (equipment 3), as shown in Figure 4.18. Three tags are attached to the body of every piece of equipment. Tapes are used to reinforce the attachment for all the tags. Workers and equipment are asked to complete their tasks according to their schedule as usual. The system records the data and detects the proximities. After applying the RTLS system during daily activities, the system could show the positions of the workers and equipment, as shown in Figure 4.19. In this figure, Equipment-2 (telehandler) and Worker-D are indicated. The system monitored the distance between workers and equipment not belonging to the same group. Whenever a worker enters the alert zone of equipment, the IDs of the equipment and the worker are shown on the user interface, as shown in Figure 4.19.

Figure 4.17 Tags attached to a hardhat



(a) Boom lift



(b) Telehandler

(c) Crane

Figure 4.18 Tags attached on equipment

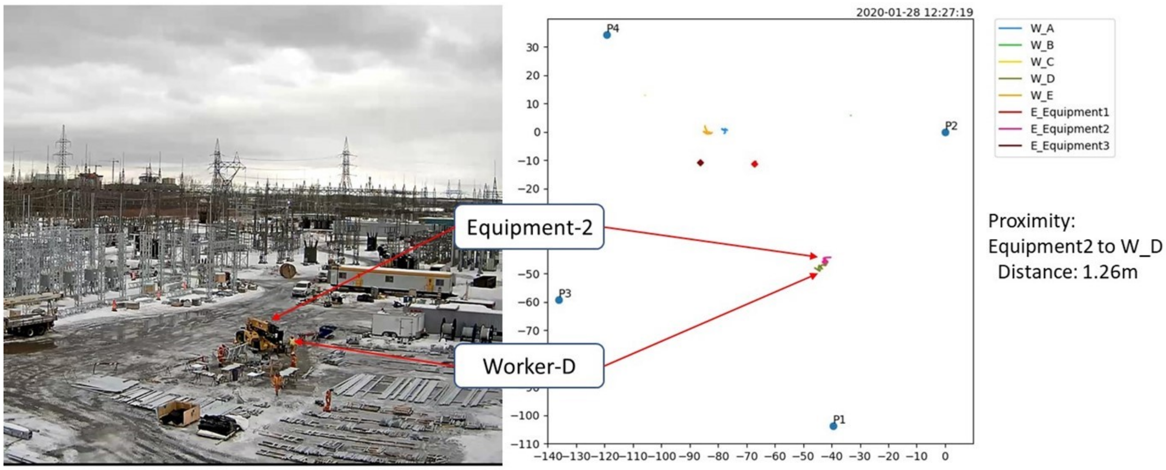


Figure 4.19 Monitoring position of equipment and workers

A post-analysis was conducted using the recorded location data to verify the function of considering grouping information from the micro-schedule. A software was developed for generating safety reports by using the recorded data. A micro-schedule developed based on the video was imported into the system. As a result, before considering the micro-schedule, the skipping period and the speed threshold, 37,238 proximities are detected. Then, after applying the information from the micro schedule, the system detects 4,109 proximities. The number of detected proximities is still big. Therefore, the skipping period and the speed threshold are needed to be considered to reduce the unnecessary alerts.

To define the duration of the skipping period and the value of the speed threshold, a sensitivity analysis is conducted as shown in Figure 4.20. The speed threshold is set to 5 km/h (1.39 m/s) and 10 km/h (2.78 m/s). The number of proximities decreases significantly if the skipping period is considered. After setting the duration higher than 120 s, the numbers do not change significantly. Therefore, the duration of the skipping period is set to 120 s. Setting the speed threshold to 5 km/h will result in decreasing the number of proximities if the skipping period is shorter than 60 s. However, if the speed threshold is set to 10 km/h, fewer proximities are detected. In order to reduce the number of unnecessary alerts, the speed threshold is set to 10 km/h.

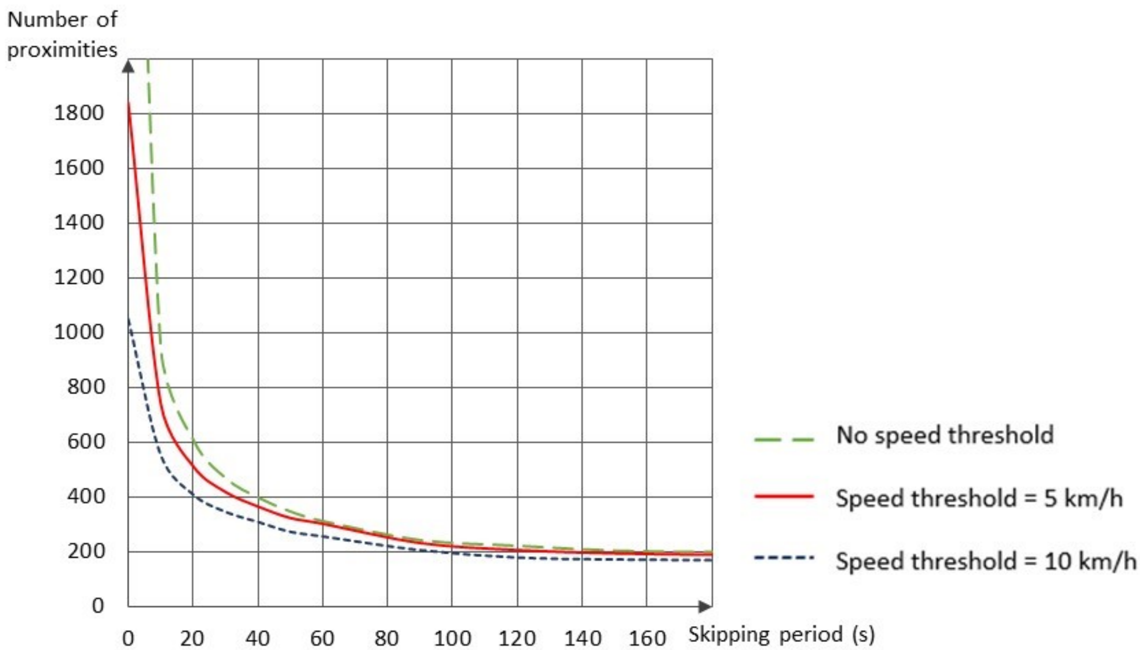


Figure 4.20 Sensitivity analysis for duration of skipping period and speed threshold

Table 4-9 shows the results of proximity detection after applying the speed threshold and the skipping period. As a result, 179 proximities are detected in ten hours with 37 proximities of Case 1, 105 proximities of Case 2 and 37 proximities of Case 3. One of the volunteer workers (Worker_D) left the site after 8:00 a.m. No proximities related to this worker can be detected. All the proximities are related to Equipment-2, which is the telehandler. The task of the telehandler on that day was hauling. Although the telehandler worked alone most of the time, there were situations that the equipment got close to the area where the workers were working to load or unload the materials. In addition, the worker may also move nearer to the equipment to help. For the other two pieces of equipment (crane and boom lift), no proximities are detected, even before the speed threshold and skipping period were considered. It was because no other workers (except the workers in the same team) are allowed to get close to them according to the safety regulation. This also reflects the fact that the workers on the construction site have followed this regulation properly. By analyzing the proximities distributed by time (Table 4-10), more proximities are detected from 8:00 to 10:00 and from 14:00 to 15:00. During these periods, the equipment moved more frequently near the zone reserved for welding and preparing elements and the zone where the installation was happening, leading to more proximity events.

Overall, there are more than 100 proximities detected in this test. However, this number is based on the fact that the alert generation function was not applied because it is not ready. Therefore, there are situations where the workers remained in a position where the system can still detect proximity because the worker was not notified. Besides, the high proximities number may also be resulted from the site planning. One solution to reduce the number of proximities is to improve the site planning. In addition, the low accuracy in positioning on this construction site is the main reason that leads to more false alerts. As shown in Figure 4.15, the estimated path of an entity in the site is always fluctuating. Because of this, the estimated velocity is higher, leading to generating a bigger dangerous zone for equipment. Therefore, to decrease the number of false alerts, it is required to increase the accuracy of the RTLS on the site. The factors leading to low accuracy and the solutions to increase the accuracy are discussed in Section 4.3.4.

Table 4-9 Number of proximities distributed by workers and cases

	Worker A	Worker B	Worker D	Worker E	Total per case
Case1	11	3	21	2	37
Case2	32	6	55	12	105
Case3	12	4	12	9	37
Total	55	13	88	23	179

Table 4-10 Number of proximities distributed by hours

Period	7:00-8:00	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00
No. of Proximities	0	30	27	19	20
Period	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00
No. of Proximities	21	17	28	2	15

4.6 Tests for Alert Generation

The alert generating function was disabled because it is not ready in the previous test. To evaluate the performance of the alert generation function, two tests and a field trial are conducted. In this part, the alerts representing the three cases of proximities are defined. A preliminary test is conducted to test the wristbands' ability to receive commands from the sensors and generate alerts on construction sites. Then, a test is conducted to evaluate the perception of the three alerts and their ability to transfer proximity information to the workers. At last, a field trial is conducted to test the prototype system with alerts generating function activated. The field trial was held on October 6th, 2020. Figure 4.21 shows the view of the site. There are two tests in the field trial. In the first test, the RTLS's alert generation function to the entities who enter a reserve zone is evaluated. In the second test, the performance of the prototype system in generating alerts for proximities is evaluated.

Figure 4.21 View of the site on October 6th, 2020

4.6.1 Defining Vibro-Tactile Alerts

According to the results from the experiment of Saket et al. (2013), a more intense vibro-tactile alert provides a more urgent feeling to the receiver. Besides, the vibro-tactile alerts can be affected by many internal factors (i.e., type of vibration motor, length, strength, and signal pattern).

In this research, the vibro-tactile alerts are generated from the Blue-up wristband. Since the circuit and the vibration motor are fixed, the vibro-tactile alerts' strength is not customizable. The length and patterns of the vibration are the only two factors that are customizable. Three patterns can be chosen in the system, which are 'blink slow', 'blink fast' and 'always on'. The three patterns are shown in Figure 4.22. In this research, pattern (1) represents the proximity of Case 1 and pattern (2) and (3) represent the proximities of Case 2 and Case 3, respectively. For the length of the alerts, if the alerts are too short, they may be missed by the receiver. On the other hand, a too-long alert annoys the receiver and may eventually lead to ignoring the alerts. The length of the vibro-tactile alerts in this research was initially set to 3 s. The length may be modified according to the feedback from the test participants in the later tests.

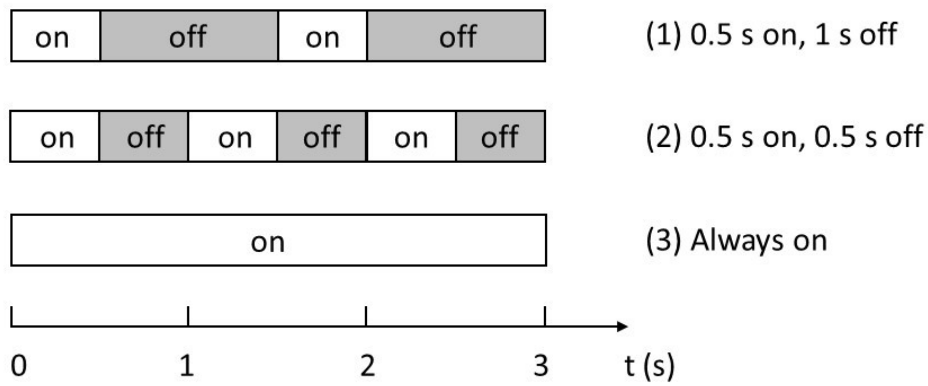


Figure 4.22 Vibration patterns

4.6.2 Preliminary Test

A preliminary test was conducted to test the wristbands' ability to receive commands on the construction site. The test conductor wore a wristband and stood at multiple locations on the site, as shown in Figure 4.23. Considering that the metal materials can affect the Bluetooth signal transmission, six testing points (a-f) were set near the main construction area, and one testing point (j) was inside the construction area to evaluate the impacts. In addition, three testing points (g, h, i) were set near the edge of the detection area, where the positioning performance is not the best, as shown in Figure 3.16, to test whether the position of the wristband has effects on receiving commands. When the conductor arrived at a testing point, he controlled the system to send a command to the wristband to generate an alert of Pattern (3) (i.e., always on) with a delay of 10 s. Once the conductor received the vibro-tactile alert, he recorded the time difference between the time when he controlled the system to generate an alert and the time when he perceived the alert by a stopwatch. At every testing point, the steps were repeated four times. As a result, the average time difference for all testing points is 10.17 s. The differences between the average time difference at each testing point are less than 0.33 s. Despite the human body's reaction time and the time used for sending a command to the server remotely, the wristband can receive commands from the system and make a reaction with less than 0.17 s average delay. No significant delay can be found in the measurements near the metal structures area. It means that the metal materials did not affect the sensors in transmitting BLE commands to the wristband. The less than 0.17 s delay indicates using the BLE RTLS and the Blue-up wristband for generating timely proximity safety alerts is feasible and promising.

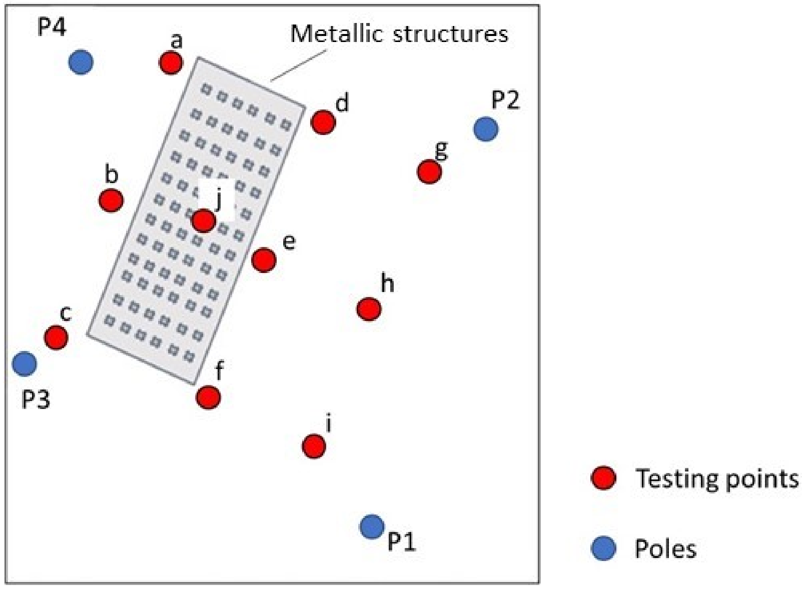


Figure 4.23 Testing points in the preliminary test

Table 4-11 Average delay of receiving alerts

Checkpoints	a	b	c	d	e
Average time to receive alerts (s)	10.18	10.16	10.15	10.31	10.16
Average delay (s)	0.18	0.16	0.15	0.31	0.16
Checkpoints	f	g	h	i	j
Average time to receive alerts (s)	10.17	10.12	10.13	10.08	10.26
Average delay (s)	0.17	0.12	0.13	0.08	0.26
Total average time to receive alerts (s)		10.17	Total average delay (s)		0.17

4.6.3 Alert Perception Test

4.6.3.1 Test Design

A test is conducted to test the perception of the defined alerts representing the three cases of proximities. To simulate the construction environment, the participants were asked to wear the hardhat, safety glasses, safety suit, and safety boots. Before the test, the test conductor explained the three kinds of alerts to the participants and let them experienced them until the participants could recognize the three kinds of alerts. The variables of the test are shown in Table 4-12.

During the test, participants were asked to wear a wristband and walk along the pre-defined route (Figure 4.24) for about five-minutes construction site. During the test, 15 alerts (five alerts for every pattern) were generated to the participants at a random time in random order. Since the participants were informed that they would receive an alert from the wristband, they would pay more attention to their arm with the wristband waiting for the alerts. Therefore, workloads were given to the participants to distract their attention and evaluate the alerts' performance under workloads. The weight of the box was set to 5 kg. Whenever the participant perceived an alert, he/she told the test conductor about the type of the alert. The missed alerts and the alerts which were not recognized correctly, were recorded. At the end of the alert perception test, the participants were asked to finish a survey (Appendix C) and give feedback on the three kinds of alerts.

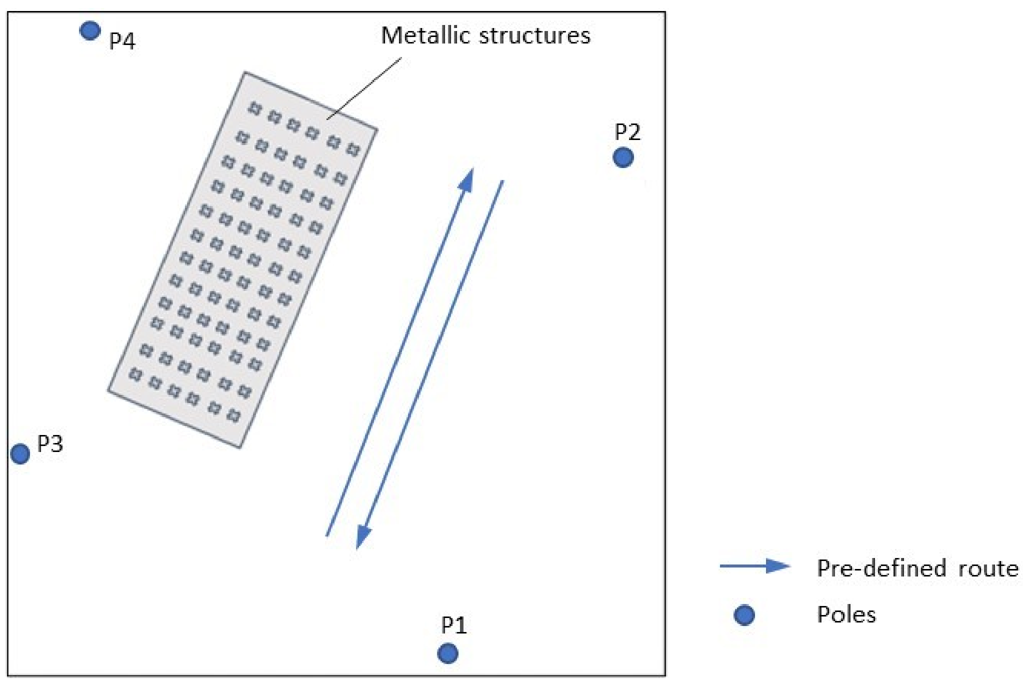


Figure 4.24 Pre-defined route

Table 4-12 Perception test variables

Time	5 min
Participants	4 people
Total number of alerts	5 alerts of Case 1
	5 alerts of Case 2
	5 alerts of Case 3

4.6.3.2 Results of Alert Perception Test

Four participants (three males and one female) joined the test. The participants can remember the three kinds of alerts after a short experience of the alerts. As a result, the participants could perceive most of the alerts with no delay after the alerts were generated. Figure 4.25 shows the accuracy of the participants' recognition of different alerts. The alerts representing Case 3 were all recognized correctly, while the other two kinds of alerts' accuracies were 90%. It is worthy of mention that the same participant contributed all the misrecognition. The participant's main confusion about the alerts was between the alerts of Case 1 and Case 2.

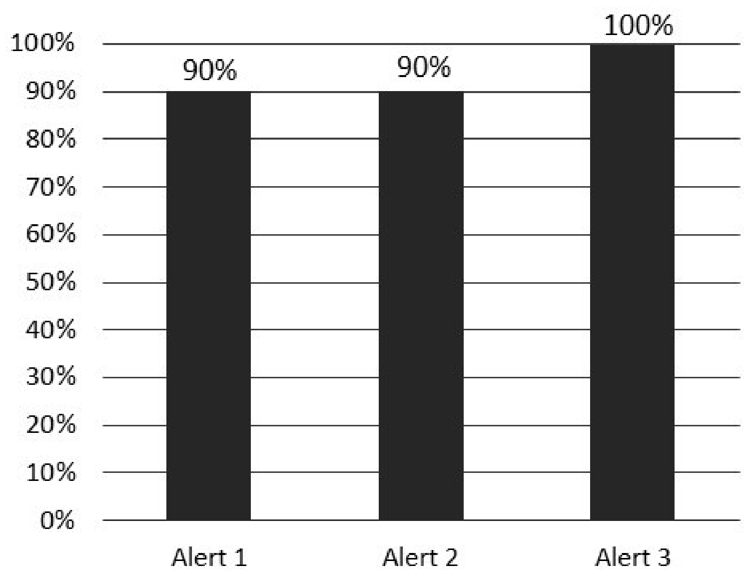


Figure 4.25 Alert recognition accuracy

The survey results are shown in Appendix C:. All participants think that the three alerts representing different cases are easy or very easy to be perceived. In addition, all the participants agree that the alert representing case 3 (Alert 3) is the most urgent one and the alert representing Case 2 (Alert 2) is the second. One participant thinks that the workload had generated a noticeable impact on alert perception, while the other participants believe that the impact was small. Last but not least, one participant thinks that the alerts' duration is too long, and he has concerns that the alert may annoy the workers because of the long duration.

4.6.4 Field Trial of Proximity Alert Generation

This field trial aimed to evaluate the performance of the system in detecting proximity and generating corresponding alerts. In the field trial, the participants were required to wear the hardhat attached with two tags, safety glasses, safety suit, safety boots, and the wristband. A vehicle was used to act as the equipment. The vehicle was attached with three tags on its top surface. There were two tests in the field trial. During the tests, tags were set to have a 5 Hz update rate. The first test aimed to evaluate the system's performance in generating alerts to workers who have entered a reserved dangerous area. And the second test is about generating alerts for proximities between workers and equipment.

4.6.4.1 First Test Design: Worker Entering the Reserved Dangerous Area

The system's performance in detecting the situations when a worker breaks through the virtual fence of the reserved dangerous area is evaluated in this test. On the site of the electric substation, the dangerous zones are: (a) area where has fall risks, (b) area with works at height, and (c) area near the activated high voltage facilities, as shown in Figure 4.26. To avoid risks, this test is not conducted in an area where risks exist. Instead, this test is conducted on a safe empty area. The simulated dangerous zone has dimensions of about 6 m × 5 m, as shown in Figure 4.27. Four traffic cones are placed on an open place of the construction site. By placing a hardhat attached with tags on the traffic cones, the position of each traffic cone can be estimated. Then, the traffic cones' positions are input into the BLE RTLS to create the reserved dangerous area. Edges ①, ②, ③, and ④ are the virtual fence of the dangerous zone. In this research, the detection of breaking through virtual fences of the dangerous zone is done by an embedded function in the BLE RTLS. This function can generate alerts to the specific wristband if the system finds that the wristband is inside the predefined dangerous area.

During the test, the participant was asked to move toward the center of the dangerous zone from a position outside of the dangerous zone and from different directions. If the RTLS detects any wristband entering the dangerous zone, alerts will be generated to the wristband. The update rate of the wristband was set to 5 Hz, which is the minimum update rate required by the 'fast response mode' as explained in Section 3.7. Once the participant receives the alert, he should stop moving and raise his hand and tell the conductor that he perceives the alert. When the alert is generated, the position of the wristband and the time of the detection is recorded. Then, this position is compared with the position of the entity estimated by the prototype system.

(a)

(b)

Figure 4.26 Dangerous zones on the construction site of electric substation

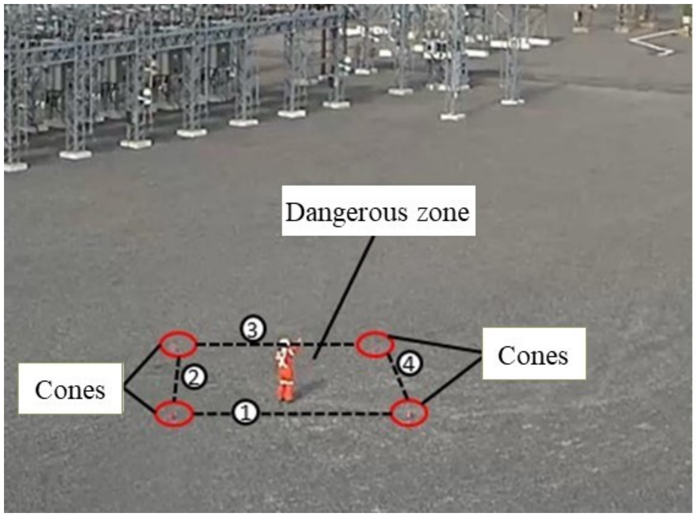


Figure 4.27 First test: worker entering reserved dangerous area

4.6.4.2 First Test: Results and Discussion

The participant made 25 attempts to enter the dangerous zone. Seven attempts of entering were from edge ①. And entering from the other three edges, each had six attempts. During the test, 64% of the attempts triggered the alerts, with seven attempts from edge ①, three attempts from edge ②, two attempts from edge ③ and four attempts from edge ④. Among all attempts where alerts could be generated, only ten attempts could generate in-time-alerts (alerts are generated within one second after the participant entered the dangerous zone). The average delay in the generation of the alerts is about two seconds. Besides, during the test, there are two times that participants are far away from (more than 5 m away from) the reserved area but received the alerts.

Figure 4.28 shows the position of the wristband captured by the RTLS and the participant's position estimated by the prototype system where the alerts were triggered during the test. The average distance between these two positions is 1.9 m. One reason for the large difference between these two positions is that the average positioning accuracy of the wristband (2.49 m) was lower than the average accuracy of the other tags (2.26 m), because the wristbands are usually covered by the sleeves of the safety suit. Besides, the wristband elevation is about 0.7m to 1 m lower than the predefined positioning elevation (1.8 m), where the system can locate the tags most accurately. In addition, the position data of the wristband have no further processing (i.e., averaging over time and averaging over multiple tags). The position of the wristband is always fluctuating and not stable. These factors can also explain the two false-positive alerts where the participant perceived the alerts before reaching the reserved area, as shown in Figure 4.28. According to the mechanism of the embedded alert generating function embedded in the RTLS, alerts are generated once the RTLS detects any presence of a wristband in the reserved zone. Therefore, when the participant moved close to the dangerous zone, the position's fluctuation may delay or advance the generation of alerts.

There are ten attempts that alerts were generated before the participant entered the reserved dangerous area, while in the real situation, the participant perceived the alert near the edge of the reserved area or after they entered the area. This can be explained by the reaction time, the execution time, and stopping distance for the participant to perceive the alerts and raise their hand. In addition, the inaccurate estimation of the entity's position lead by the more-than-two-meter accuracy in positioning of tags can also be one of the reasons.

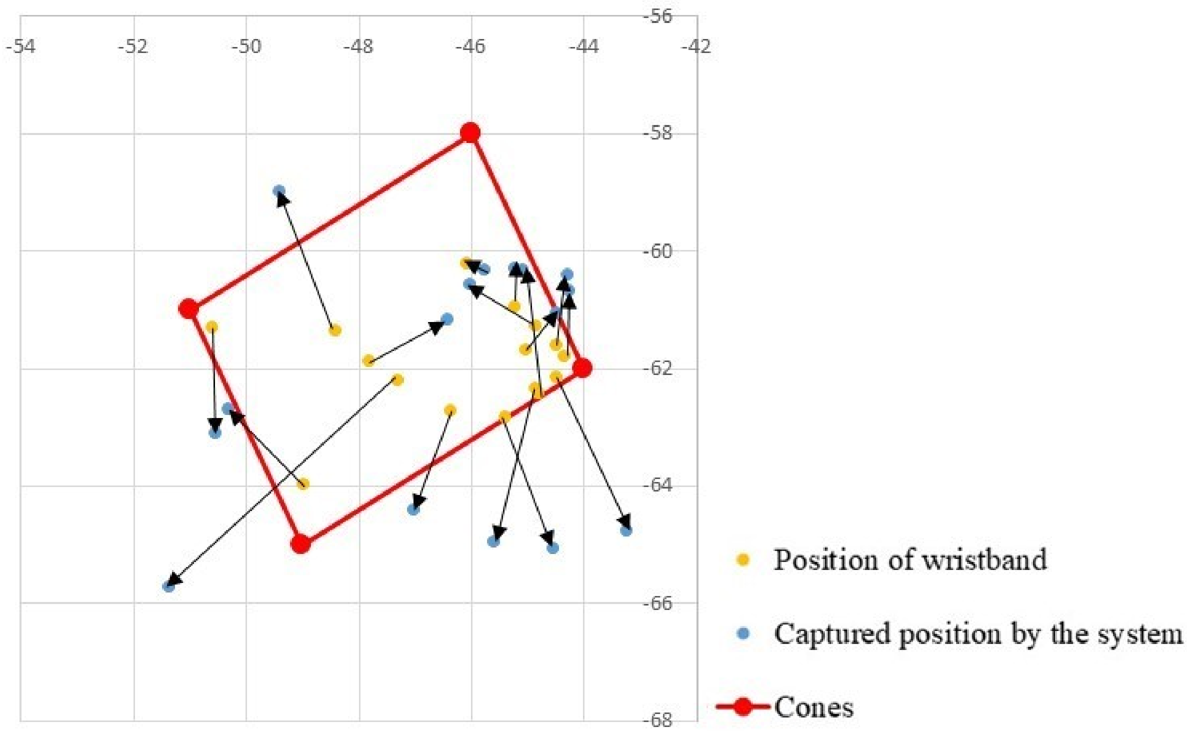
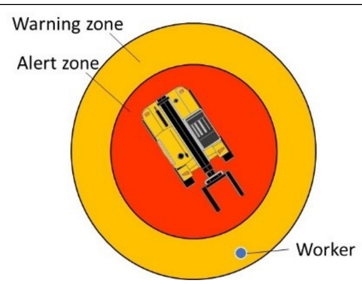
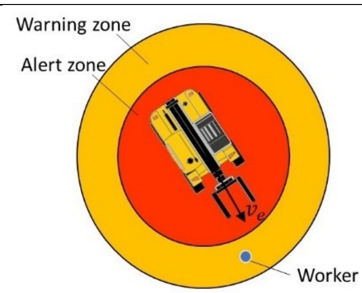
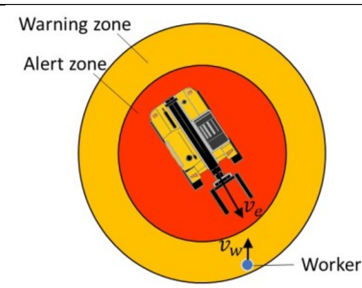
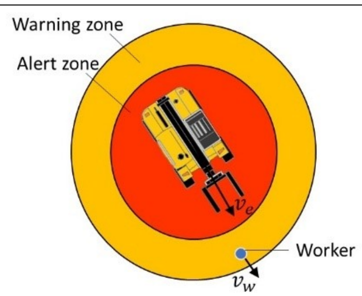
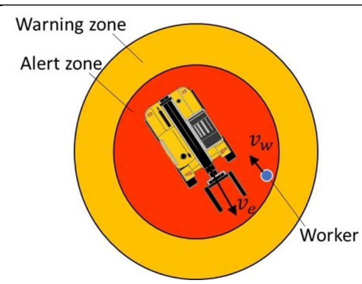


Figure 4.28 Captured position of tags and participant when the alerts are triggered

4.6.4.3 Second Test Design: Proximity Alerts with Equipment

The second test is conducted to test the performance of the prototype system in generating alerts for the proximity events. Table 4-13 lists five common scenarios of proximities between equipment and workers. In Scenarios 2, 3 and 4, the equipment is moving toward the worker. The workers are expected to receive an alert representing the proximity of Case 2 and then an alert representing Case 3 once they enter the alert zone. In Scenario 5, the case when a worker is too close to the equipment is simulated. The occurrences and the damage of proximities of Case 1 are less than the proximities of the other two cases because the worker should have received an alert representing the Case 2 or Case 3 before he receives the alert of Case 1. He will not receive the alert of Case 1 if he exits the warning zone within 10 s as mentioned in Section 3.4.3. One general scenario of the proximity of Case 1 is that the worker does not move far enough from the work zone of a piece of stationary equipment (e.g., boom lift, crane) and stops to work again.

Table 4-13 Main five scenarios

Scenario 1	Scenario 2	Scenario 3
 A stationary worker inside the warning zone of a stationary equipment	 A stationary worker inside the warning zone; equipment approaches the worker	 Worker and the equipment are approaching each other
Scenario 4	**Scenario 5**	
 Worker and equipment move in the same direction; $v_E > v_w$	 Worker inside the alert zone of the equipment	

In this test, the Scenarios 3, 4 and 5 are tested because these three scenarios are more dangerous and common. Each scenario was tested four times for each participant. In the expectation of Scenarios 3 and 4, the participants would receive alerts representing Case 2 first and then receive alerts representing Case 3. For Scenario 5, the participant would only receive alert representing Case 3. The alerts generated by the system and the alerts that the participant can perceive in each attempt were recorded to see whether the prototype system can generate alerts correctly. A vehicle is used to simulate the equipment. To define the dangerous zone, the parameters are set as follows:

(1) R_0 is set to 2.5 m because the dimensions of the vehicle are 4.55 m × 1.84 m (*Vehicle Specs | 2021 CX-5 | Mazda Canada*, n.d.).
(2) B_1 is set to 1.5 m;
(3) The walking speed is set to 1.46 m/s, which is the average walking speed of people aged 14 to 64 (Knoblauch et al., 1996). The pedestrians' average reaction time to vibro-tactile alert from a wristband is about 300 ms (Jiang & Hannaford, 2015). Considering the execution time for the pedestrians to stop, the reaction time is set to 1 s;
(4) The total reaction time for a driver (involving reaction time of operator and execution time) is 2.5 s (Technology Associates, 2014);
(5) The signature braking distance of the vehicle from 60 MPH (26.82 m/s) to 0 is 125 ft (38.1 m) (Mazda n.d.). Therefore, the deceleration of the vehicle is about 9.44 m/s^2.

4.6.4.4 Second Test: Results and Discussions

Three participants joined this test. The system can detect different cases' proximities and generate alerts to the participants. During the test, each participant has made six attempts for each scenario (Scenarios 3, 4 and 5). All the attempts could trigger the alerts. Besides, the participants could correctly perceive and recognize all the alerts generated by the system. However, the generated alerts were sometimes different from the expected alerts. In Scenario 3, 72% of the attempts generated the expected alerts with the expected order. For Scenarios 4 and 5, this percentage is 89% and 66%, respectively. Overall, the accuracy of alerts generation is 77.78% for Alert 2 and 59.72% for Alert 3, as shown in Table 4-14. There are 19.44% of Alerts 2 that were delayed for more than 2 s, and 2.78% of Alerts 2 were missing. For Alert 3, the missing alert rate and delayed alert rate are higher.

Table 4-14 Alert generating accuracy

	Alert 2	Alert 3
Accuracy	77.78%	59.72%
Missing alert rate	2.78%	16.67%
Delayed alert rate (more than 2 s)	19.44%	23.61%

The results of this test are affected by the un expected accuracy of the RTLS. There are factors generating errors to the RTLS as explained in Section 4.3.4. Therefore, there are fluctuations in the estimated paths of entities, which bring more errors to the proximity detections. The fluctuations can also explain the lower accuracy in the generation of Alert 3. Because of the fluctuations, the estimated speed of the equipment become higher and a larger warning zone is generated. Because the size of the warning zone (with an average radius of 8.4 m) was much larger than the alert zone (with fix radius of 4 m), when detecting workers moving to the equipment with fluctuations, it is easier for the system to detect proximities of Case 2. To improve the accuracy of proximity detection, the best way is to improve the accuracy of the RTLS. As mentioned in Section 4.3.4, several improvements can be applied to increase the accuracy. In addition to the RTLS, the compensation for missing data using extrapolation when more than 35% of data were missing could affect the proximity detection. One potential solution to address this limitation in future work is fusing data with the data from other kinds of sensors (e.g., inertial measurement unit, computer vision).

4.7 Summary

In this chapter, the wireless setup of the BLE RTLS on a construction site is explained. The BLE RTLS based on AOA is functional when using the wireless scheme for data communication. Two tests were conducted on the construction site to evaluate the performance of the system in different environmental conditions (without and with the built-up metal structures). In the initial test, when no metal structures were built on the site, the missing data rate of tags was low and the system provided about 2m positioning accuracy for the workers. For tags attached directly on the metallic surface, the missing data rate was increased and more errors were found in the positioning results. In the second test, where metal structures had been built on the site, similar steps were repeated to evaluate the performance of the system. The missing data rate increased. Moreover, a significant impact on accuracy was found. Based on the results from these two tests, the factors that lead to the unexpected performance of the RTLS are listed with the corresponding solutions.

In the case study, a test was conducted during construction activities. The proposed system can detect proximities between workers and equipment. A sensitivity analysis was conducted to define the value of the skipping period and the speed threshold for alert generation. After considering the grouping information retrieved from a micro-schedule, the skipping period and the speed threshold, the system can reduce the number of false alerts and unnecessary alerts.

Two other tests were conducted to test the performance of the RTLS in generating alerts and the workers' perceptions of the alerts. The test results show that the BLE RTLS has the ability to generate alerts to the wristbands without delay. About the alert perception, even if the workers are handling tasks, they could perceive the alert effectively and recognize the severity of the risk that they are encountering.

A field trial is conducted to test the alert generation for the situations that workers enter the reserved dangerous zone on the construction site, and the proximities between workers and equipment. As a result, the system can generate alerts for entering the reserved stationary dangerous zones and proximities between workers and equipment. However, the accuracy of the alerts is affected by the accuracy of the locations caused by the factors that are explained in Section 4.3.4. Although the system may generate delayed alerts or detect a wrong case of proximity events, during the test, there were always alerts generated to notify the workers when they were inside the dangerous zone of equipment. By following the solutions listed with the factors which generate errors, as mentioned in Section 4.3.4, it is possible to improve the accuracy of the RTLS and improve the performance in proximity detection and dangerous zone entering detection.

Chapter 5: Conclusions and Future Work

5.1 Summary of Research

This research investigated the applicability of applying a wireless scheme of BLE RTLS for generating near real-time alerts to workers on construction sites. The literature review mentioned that the previous research about the applications of RTLS technologies could improve construction safety. However, there are limitations (e.g., high cost or low accuracy, difficult cabling) that affect the feasibility on construction sites. The BLE RTLS based on AOA has the potential to address most of the limitations when providing high-quality positioning with relatively low cost.

This research focuses on developing a method that can effectively detect proximities and generate alerts to workers and evaluating the feasibility of applying the BLE RTLS based on AOA on construction sites. The requirements of using BLE RTLS based on AOA were listed with the corresponding solutions. A wireless scheme for the installations of the RTLS, which mitigates the cabling problem on construction sites, was presented. A method for detecting proximities and generating alerts on construction sites was developed. The method consists of two parts, which are the data processing module and the proximity detection module. The data processing module can estimate the position and velocity of workers and equipment. It can also make corrections for the missing data and errors. In the proximity detection module, a dangerous zone is generated using the position and velocity of equipment. With the position and velocity of workers and equipment, three cases of proximity can be detected and the corresponding alerts can be generated. By considering the grouping information from the micro-schedule, the skipping period and the speed threshold, the number of false alerts and unnecessary alerts can be reduced. At last, the events are saved once detected, and a report for the proximity events will be generated at the end of a day/week to the local hard disk and the cloud database.

Based on the method, a prototype system was developed using the BLE RTLS based on AOA. In the case studies, the BLE RTLS was set up on a construction site of an electric substation. Two tests were conducted to test the accuracy of the system in positioning. The results show that the performance of the RTLS on this construction site is not as expected. Six factors that may lead to the low accuracy in positioning are listed with the corresponding advice. A case study was then conducted to define the duration of skipping period and the value of speed threshold, and evaluate the performance of proximity detection during real construction activities. The system was activated during daily construction activities. The system is able to detect proximities and the proximities are compared with the video of the site. A sensitivity is conducted with the collected data to define the skipping period and the speed threshold. After considering the grouping information from micro-schedule, skipping period and speed threshold, the number of false alerts is reduced.

Then, the three vibro-tactile alerts representing the three cases of proximities, which can effectively notify the workers about their encountering proximities, were defined. Two tests were conducted on the construction site to evaluate the performance of the alert generation function of the RTLS, and the perception of the three defined alerts by the workers. As a result, the system can provide near real-time alerts to workers on construction sites. The three alerts can be easily perceived and recognized correctly. Besides, according to the survey results, all participants agree that the three kinds of alerts can properly represent the proximity cases' severities. Using these three kinds of alerts as the notifications to workers about their encountering proximities is promising.

A field trial is conducted to test the alert generation for the situations that workers enter the reserved dangerous zone on the construction site, and the proximities between workers and equipment. As a result, the system can generate alerts for entering the reserved stationary dangerous zones and proximities between workers and equipment. However, the accuracy of the alerts is affected by the accuracy of the locations caused by the factors that are explained in Section 4.3.4. Although the system may generate delayed alerts or detect a wrong case of proximity events, during the test, there were always alerts generated to notify the workers when they were inside the dangerous zone of equipment. By following the solutions listed with the factors which generate errors, as mentioned in Section 4.3.4, it is possible to improve the accuracy of the RTLS and improve the performance in proximity detection and dangerous zone entering detection.

5.2 Research Contributions and Conclusions

The main contributions of this research are: (1) developing and evaluating a wireless scheme for the deployment of BLE RTLS on construction sites; (2) developing a method which identifies the cases of proximity based on the position and movement of the entities; (3) developing a method which can reduce the number of false alert by considering the grouping information between workers and equipment; (4) designing the vibro-tactile alerts representing the three cases of proximities, which can be easily perceived, and notifying the involved workers about the information of the event; (5) developing the prototype system based on the method using a wireless scheme of BLE RTLS; (6) evaluating the feasibility of applying BLE RTLS on construction sites for proximity detection.

The conclusions of this research are shown as follows:

(1) The proposed method can estimate the position and velocity of entities on construction sites. Without the impacts from metal materials and obstacles on the construction, for the position estimation of workers, about 96% of estimation have errors within 2 m, while more than 60% of estimation can reach submeter accuracy. The accuracy of the equipment position estimation is lower because attaching tags to metallic surface generate errors. However, by adding a sponge pad between the tags and the metallic surface, this problem can be mitigated. In addition, attaching tags on the metallic surface, not enough density of sensors and missing data during wireless communication are also factors that lead to missing data and errors.
(2) By using the estimated information, the method can detect the proximity between equipment and workers and generate alerts when necessary. By considering the grouping information from micro-schedule, the skipping period and the speed threshold, the system can reduce the number of false alerts and unnecessary alerts.
(3) Vibro-tactile alerts can be perceived within 1 s by the workers wearing a wristband. Workers can recognize the type of alerts correctly.

5.3 Limitations and Future Work

The limitations and the future work to address the limitations are listed as follows:

(1) The presence of more metal materials and the built-up metal structures on the construction site significantly affect the positioning performance. The metal materials can reflect the radio signals and generate errors to the RTLS, which is an inevitable limitation of the radio-based RTLS. However, in this construction site, the main materials on the construction site were metal. Therefore, many errors were caused by the metal materials during the test. The accuracy is better in the excavation and foundation building phase. As a part of the future work, the RTLS will be deployed in another environment where has fewer metal materials. More tests

will be conducted to evaluate the performance of the RTLS and test the feasibility of applying the proposed proximity warning system in the long term.

(2) The built-up structures can occlude the sensors, which generate more errors to the RTLS. In this research, installing sensors on the four poles is not enough to provide high resolution positioning in all areas on the construction site. In the future, a better plan to distribute the sensors at different points surrounding the site will be developed to mitigate the errors caused by occlusions.

(3) There are errors that may be caused by missing data in the wireless data communication. The data missing may be resulted from the limited broadband speed and unstable wireless connection. Extra test about the data communication with the wireless connection will be conducted in the future to find the factors causing the missing data.

(4) The estimations for the position and the velocity are based on the RTLS data. Therefore, when the missing data rate is high and the accuracy of the RTLS is low, these estimations are less reliable. In addition, with a high missing data rate, the compensation for missing data using extrapolation is ineffective. By adding more rules to the data processing in the future (e.g., rules to deal with the impacts from occlusions), these problems can be mitigated. The RTLS data can be fused with data from other sensors which is less affected by the occlusions and metal materials. When the RTLS data is less reliable, the estimation can be done by using the more reliable data from the other sensors.

(5) For proximity detection module, in this methodology, false-positive alerts may be generated because of the use of cylindrical dangerous zone. By applying the buffer workspaces, as mentioned in Section 2.2.1, the number of unnecessary alerts can be reduced. However, the application of buffer workspace requires accurate information about the orientation of the equipment and states of the parts of the equipment. More rules are required in this method to estimate the pose and the motion state of entities since the orientation estimation using only RTLS is not accurate and reliable. Therefore, as a part in the future work, a method will be developed to capture the poses of equipment with more degrees of freedom using data from other sensors (e.g., inertial measurement unit). Then with the pose of equipment, a more detailed dangerous zone of the equipment will be generated.

(6) In addition, although the grouping information in the micro-schedule is considered, as mentioned in Section 3.4, the micro-schedule prepared by the project manager in advance is not fully complied by workers and equipment operators. This may also lead to false alerts and missed alerts. For example, workers may help the equipment in some tasks for a short period (e.g., 15 min). But the system keeps sending alerts to them, which may annoy the workers and affect their tasks. As a part of the future work, a function that allows the workers to snooze the alerts will be developed. In addition, a more intelligent method, which considers the information from BIM for making adjustments on the micro-schedule will be developed.

References

450 Series Spec Sheet. (n.d.). JLG Industries, Inc. https://www.jlg.com/-/media/JLG/Current-Materials-No-Password/Products/Americas%20-%20ANSI/Engine-Powered-Boom-Lifts/Articulating-Booms/450-Series-II/450%20Series%20Spec%20Sheet%20A9220%20Compliant.pdf

Afaneh, M. (2019, July 30). *Bluetooth Low Energy: A Primer*. Interrupt. https://interrupt.memfault.com/blog/bluetooth-low-energy-a-primer

Ansaripour, A., Heydariaan, M., Gnawali, O., & Kim, K. (2020). ViPER: Vehicle pose estimation using ultra-wideband radios. *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*.

Bensky, A. (2016). *Wireless Positioning Technologies and Applications, Second Edition*. Artech House.

BlueBeacon SafeBand. (2019). Blueup S.r.l.s.

Bluetooth Technology Website | The official website of Bluetooth technology. (n.d.). Bluetooth® Technology Website. Retrieved 9 December 2020, from https://www.bluetooth.com/

Cao, Z., Chen, R., Guo, G., & Pan, Y. (2018). iBaby: A low cost BLE pseudolite based indoor baby care system. *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 1–6.

Cat® TL1055C Telehandler Spec Sheet. (2012). Caterpillar Inc. https://www.cat.com/en_US/products/new/equipment/telehandlers/telehandlers/1000003202.html

Chae, S., & Yoshida, T. (2010). Application of RFID technology to prevention of collision accident with heavy equipment. *Automation in Construction*, *19*(3), 368–374. https://doi.org/10.1016/j.autcon.2009.12.008

Cheng, T., Venugopal, M., Teizer, J., & Vela, P. A. (2011). Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments. *Automation in Construction*, *20*(8), 1173–1184. https://doi.org/10.1016/j.autcon.2011.05.001

de Souza, G. A., Torres, L. A., Dani, V. S., Villa, D. S., Larico, A. T., Maciel, A., & Nedel, L. (2018). Evaluation of Visual, Auditory and Vibro-Tactile Alerts in Supervised Interfaces. *2018 20th Symposium on Virtual and Augmented Reality (SVR)*, 163–169.

Ding, L. Y., Zhou, C., Deng, Q. X., Luo, H. B., Ye, X. W., Ni, Y. Q., & Guo, P. (2013). Real-time safety early warning system for cross passage construction in Yangtze Riverbed Metro Tunnel based on the internet of things. *Automation in Construction*, *36*, 25–37.

EnduranceMachine. (2019, January 8). What is the Average Human Running Speed? *Kettle100*. https://www.kettle100.com/what-is-the-average-human-running-speed/

Eurostat. (2019, November). *Accidents at work statistics—Statistics Explained*. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Accidents_at_work_statistics

Feige, S. (2009). Can You Feel It?–Using vibration rhythms to communicate information in mobile contexts. *IFIP Conference on Human-Computer Interaction*, 800–803.

Government of Canada, C. C. for O. H. and S. (2020, July 5). *Noise - Occupational Exposure Limits in Canada: OSH Answers*. https://www.ccohs.ca/

Hammad, A., Setayeshgar, S., Zhang, C., & Asen, Y. (2012). Automatic generation of dynamic virtual fences as part of BIM-based prevention program for construction safety. *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 1–10.

Hwang, J., & Hwang, W. (2011). Vibration perception and excitatory direction for haptic devices. *Journal of Intelligent Manufacturing*, *22*(1), 17–27.

Jang, W.-S., & Skibniewski, M. J. (2009). Embedded system for construction asset tracking combining radio and ultrasound signals. *Journal of Computing in Civil Engineering*, *23*(4), 221–229.

Jiang, I., & Hannaford, B. (2015). Comparison of reaction times while walking. *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, 345–349.

JISHA. (2020). *JISHA:OSH Statistics in Japan*. https://www.jisha.or.jp/english/statistics/

Kelm, A., Laußat, L., Meins-Becker, A., Platz, D., Khazaee, M. J., Costin, A. M., Helmus, M., & Teizer, J. (2013). Mobile passive Radio Frequency Identification (RFID) portal for automated and rapid control of Personal Protective Equipment (PPE) on construction sites. *Automation in Construction*, *36*, 38–52.

Knoblauch, R. L., Pietrucha, M. T., & Nitzburg, M. (1996). Field studies of pedestrian walking speed and start-up time. *Transportation Research Record*, *1538*(1), 27–38.

Lee, H.-S., Lee, K.-P., Park, M., Baek, Y., & Lee, S. (2012). RFID-based real-time locating system for construction safety management. *Journal of Computing in Civil Engineering*, *26*(3), 366–377.

Lee, K.-P., Lee, H.-S., Park, M., Kim, H., & Han, S. (2014). A real-time location-based construction labor safety management system. *Journal of Civil Engineering and Management*, *20*(5), 724–736. https://doi.org/10.3846/13923730.2013.802728

Li, H., Chan, G., Wong, J. K. W., & Skitmore, M. (2016). Real-time locating systems applications in construction. *Automation in Construction*, *63*, 37–47.

Maalek, R., & Sadeghpour, F. (2013). Accuracy assessment of Ultra-Wide Band technology in tracking static resources in indoor construction scenarios. *Automation in Construction*, *30*, 170–183.

Maalek, Reza, & Sadeghpour, F. (2016). Accuracy assessment of ultra-wide band technology in locating dynamic resources in indoor scenarios. *Automation in Construction*, *63*, 12–26.

Marks, E. D., & Teizer, J. (2013). Method for testing proximity detection and alert technology for safe construction equipment operation. *Construction Management and Economics*, *31*(6), 636–646. https://doi.org/10.1080/01446193.2013.783705

Monfared, S., Nguyen, T.-H., Petrillo, L., De Doncker, P., & Horlin, F. (2018). Experimental demonstration of BLE transmitter positioning based on AOA estimation. *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 856–859.

Occupational Safety and Health Administration. (2020). *Commonly Used Statistics | Occupational Safety and Health Administration*. https://www.osha.gov/data/commonstats

Park, M.-W., & Brilakis, I. (2012). Construction worker detection in video frames for initializing vision trackers. *Automation in Construction*, *28*, 15–25.

Peon, A. R., & Prattichizzo, D. (2013). Reaction times to constraint violation in haptics: Comparing vibration, visual and audio stimuli. *2013 World Haptics Conference (WHC)*, 657–661.

Pradhananga, N., & Teizer, J. (2013). Automatic spatio-temporal analysis of construction site equipment operations using GPS data. *Automation in Construction*, *29*, 107–122.

Quuppa Academy Training Material. (2019). Quuppa.

Quuppa Tag Module User Guide v. 1.8. (2019). Quuppa.

Real Time Locating System (RTLS) | powered by Quuppa. (2020). Quuppa. https://quuppa.com/

Ruff, T. M. (2007). *Recommendations for evaluating and implementing proximity warning systems onsurface mining equipment*.

Ruiz, A. R. J., & Granja, F. S. (2017). Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis. *IEEE Transactions on Instrumentation and Measurement*, *66*(8), 2106–2117.

Saket, B., Prasojo, C., Huang, Y., & Zhao, S. (2013). Designing an effective vibration-based notification interface for mobile phones. *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, 149–1504.

Siddiqui, H. (2014). *UWB RTLS for construction equipment localization: Experimental performance analysis and fusion with video data* [PhD Thesis]. Concordia University.

Siddiqui, H., Vahdatikhaki, F., & Hammad, A. (2019). Case study on application of wireless ultra-wideband technology for tracking equipment on a congested site. *ITcon*, *24*, 167–187.

Sinclair, J. D., & Haflidson, W. O. (1995). Construction noise in Ontario. *Applied Occupational and Environmental Hygiene*, *10*(5), 457–460.

Skibniewski, M. J., & Jang, W.-S. (2009). Simulation of accuracy performance for wireless sensor-based construction asset tracking. *Computer-Aided Civil and Infrastructure Engineering*, *24*(5), 335–345.

Soltanmohammadlou, N., Sadeghi, S., Hon, C. K., & Mokhtarpour-Khanghah, F. (2019). Real-time locating systems and safety in construction sites: A literature review. *Safety Science*, *117*, 229–242.

Suter, A. H. (2002). Construction noise: Exposure, effects, and the potential for remediation; a review and analysis. *Aiha Journal*, *63*(6), 768–789.

Tang, H. (2014). *DOA estimation based on MUSIC algorithm*.

Technology Associates. (2014). *Brake Reaction Time*. https://www.webpages.uidaho.edu/niatt_labmanual/chapters/geometricdesign/theoryandconcepts/BrakeReactionTime.htm

Teizer, J. (2015). Wearable, wireless identification sensing platform: Self-monitoring alert and reporting technology for hazard avoidance and training (SmartHat). *Journal of Information Technology in Construction (ITcon)*, *20*(19), 295–312.

Teizer, J., Allread, B. S., Fullerton, C. E., & Hinze, J. (2010). Autonomous pro-active real-time construction worker and equipment operator proximity safety alert system. *Automation in Construction*, *19*(5), 630–640. https://doi.org/10.1016/j.autcon.2010.02.009

Teizer, J., Venugopal, M., & Walia, A. (2008). Ultrawideband for automated real-time three-dimensional location sensing for workforce, equipment, and material positioning and tracking. *Transportation Research Record*, *2081*(1), 56–64.

U.S. Bureau of Labor Statistic. (2018). *Fatal occupational injuries by event or exposure for all fatal injuries and major private industry sector*. https://www.bls.gov/iif/oshwc/cfoi/cftb0330.htm

Vahdatikhaki, F., & Hammad, A. (2014). Framework for near real-time simulation of earthmoving projects using location tracking technologies. *Automation in Construction*, *42*, 50–67.

Vahdatikhaki, F., & Hammad, A. (2015). Dynamic equipment workspace generation for improving earthwork safety using real-time location system. *Advanced Engineering Informatics*, *29*(3), 459–471.

Vahdatikhaki, F., Hammad, A., & Siddiqui, H. (2015). Optimization-based excavator pose estimation using real-time location systems. *Automation in Construction*, *56*, 76–92.

Van der Ham, M. (2015). *Real time localization of assets in hospitals using quuppa indoor positioning technology*.

Vehicle Specs | 2021 CX-5 | Mazda Canada. (n.d.). Retrieved 10 November 2020, from https://www.mazda.ca/en/vehicles/cx-5/specs/

Wang, J., & Razavi, S. N. (2016). Low false alarm rate model for unsafe-proximity detection in construction. *Journal of Computing in Civil Engineering*, *30*(2), 04015005.

Wang, Y., Yang, X., Zhao, Y., Liu, Y., & Cuthbert, L. (2013). Bluetooth positioning using RSSI and triangulation methods. *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, 837–842.

What Is MongoDB? (n.d.). MongoDB. Retrieved 7 August 2020, from https://www.mongodb.com/what-is-mongodb

Wu, W., Yang, H., Chew, D. A., Yang, S., Gibb, A. G., & Li, Q. (2010). Towards an autonomous real-time tracking system of near-miss accidents on construction sites. *Automation in Construction*, *19*(2), 134–141.

Yang, J., Arif, O., Vela, P. A., Teizer, J., & Shi, Z. (2010). Tracking multiple workers on construction sites using video cameras. *Advanced Engineering Informatics*, *24*(4), 428–434.

Yi, W., Chan, A. P. C., Wang, X., & Wang, J. (2016). Development of an early-warning system for site work in hot and humid environments: A case study. *Automation in Construction*, *62*, 101–113. https://doi.org/10.1016/j.autcon.2015.11.003

Zhang, C., Hammad, A., & Rodriguez, S. (2012). Crane pose estimation using UWB real-time location system. *Journal of Computing in Civil Engineering*, *26*(5), 625–637.

Zhang, Cheng, & Hammad, A. (2012). Multiagent approach for real-time collision avoidance and path replanning for cranes. *Journal of Computing in Civil Engineering*, *26*(6), 782–794.

Zhang, Cheng, Hammad, A., Soltani, M., Setayeshgar, S., & Motamedi, A. (2012). Dynamic virtual fences for improving workers safety using BIM and RTLS. *Proceedings of the 14th International Conference on Computing in Civil and Building Engineering*.

Zhou, C., & Ding, L. Y. (2017). Safety barrier warning system for underground construction sites using Internet-of-Things technologies. *Automation in Construction*, *83*, 372–389.

Zhu, Z., Park, M.-W., Koch, C., Soltani, M., Hammad, A., & Davari, K. (2016). Predicting movements of onsite workers and mobile equipment for enhancing construction site safety. *Automation in Construction*, *68*, 95–101.

Appendix A: Testing Quuppa Sensors Under Electromagnetic Noise

In order to avoid facing a problem on the construction site where there are high voltage power cables and electric substation nearby, a test has been done in the lab with different levels of noise. The effect and the results for the test are discussed below.

Figure A1 shows the camera and one of the sensors used in the test. Figure A2 shows the overall view of four sensors that are attached to the. After installing all sensors, a corner of the lab was selected as the reference point (0, 0, 0), and then the coordinates for each sensor was measured with a laser meter to the wall and calculated based on the reference point. The measured coordinates were used as an input to the software which will recognize and assign a location for each sensor.

Figure A1 RTLS sensor and camera being tested

Figure A2 Setting up BLE RTLS in the lab

After locating each sensor in the software calibration was needed. The focusing locator LT-6D was only used for the calibration process. Since there is no need to have focusing locator after calibrating, a normal cable (not shielded) was used. Steps through calibration are, first, measuring the coordinates of locator while standing in the middle of four sensors. Second, giving the coordinate to the software as a known location for the locator. Third, using the focusing locator to get orientations. Figure A3 shows the estimation of positioning quality. The background colors indicate the performance of the positioning. The area colored in green means a high-resolution zone with high accuracy and the area in yellow means low-resolution zone and finally, the red area shows a zone with incorrect position data.

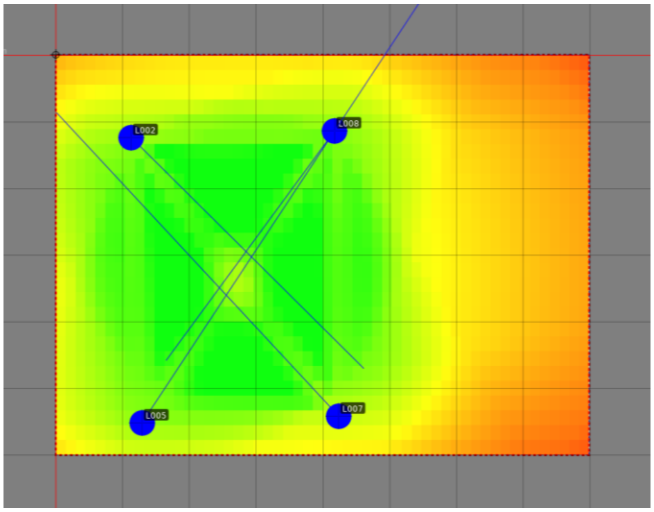


Figure A3 Quality estimation

The tags were configured to transmit signals five times per second to the locators and were placed in the high-resolution area. This experiment was done in an environment with a noise level of 50

KV to 250 KV. The experiment started at 50 KV with an increase of 10 KV in each step. Starting time and ending time of each step are shown in Table A1. Data of x, y, z and accuracy from Quuppa system, and time of every capturing were recorded for at least 1 minute in every voltage step in JSON format during the experiment. It is worthy to note that the accuracy of data is provided by Quuppa system, which estimates accuracy of the position in meters.

Table A1 Voltage steps

Voltage (kv)	start time	end time	duration	estimate number of captured data (duration/0.2sec)	Number of captured data
50	11:01:00.00	11:02:00.00	00:01:00.00	300.00	300
60	11:03:00.00	11:04:00.00	00:01:00.00	300.00	300
70	11:04:00.00	11:05:00.00	00:01:00.00	300.00	300
80	11:05:00.00	11:06:00.00	00:01:00.00	300.00	300
90	11:06:00.00	11:07:00.00	00:01:00.00	300.00	300
100	11:24:04.00	11:25:04.00	00:01:00.00	300.00	300
110	11:25:04.00	11:26:16.00	00:01:12.00	360.00	360
120	11:26:16.00	11:27:22.14	00:01:06.14	330.70	331
130	11:27:22.14	11:28:29.51	00:01:07.37	336.85	337
140	11:28:29.51	11:29:37.58	00:01:08.07	340.35	340
150	11:40:00.00	11:41:00.00	00:01:00.00	300.00	328
160	11:41:00.00	11:42:15.00	00:01:15.00	375.00	375
170	11:42:15.00	11:43:17.00	00:01:02.00	310.00	310
180	11:43:17.00	11:44:17.00	00:01:00.00	300.00	300
190	11:44:17.00	11:45:22.69	00:01:05.69	328.45	328
200	11:45:22.69	11:46:33.56	00:01:10.87	354.35	354
210	11:59:00.00	12:00:10.99	00:01:10.99	354.95	355
220	12:00:10.99	12:01:16.20	00:01:05.21	326.05	326
230	12:01:16.20	12:02:22.39	00:01:06.19	330.95	331
240	12:02:22.39	12:03:27.36	00:01:04.97	324.85	324
250	12:03:27.36	12:05:00.00	00:01:32.64	463.20	464

Result and conclusion

Tag-1 was picked as an example to illustrate how different noises generated by higher voltage affect the Quuppa system performance. Figure A3 shows the data of Tag-1 in different voltages in the sequence of time. The coordinate data were fluctuating around a certain value. Generally, the deviation was less than 20 cm which is the same as the normal condition. While adding more voltage to the environment, no specific change showed up. The accuracy provided by Quuppa system is about 0.13 to 0.17 m, the same as that in normal condition. A small deviation happened at the point of 210KV. It is because the tags were moved manually to make sure they were triggered.

Otherwise, tags may enter default mode which transmits a different signal to sensors if tags did not move for a long time.

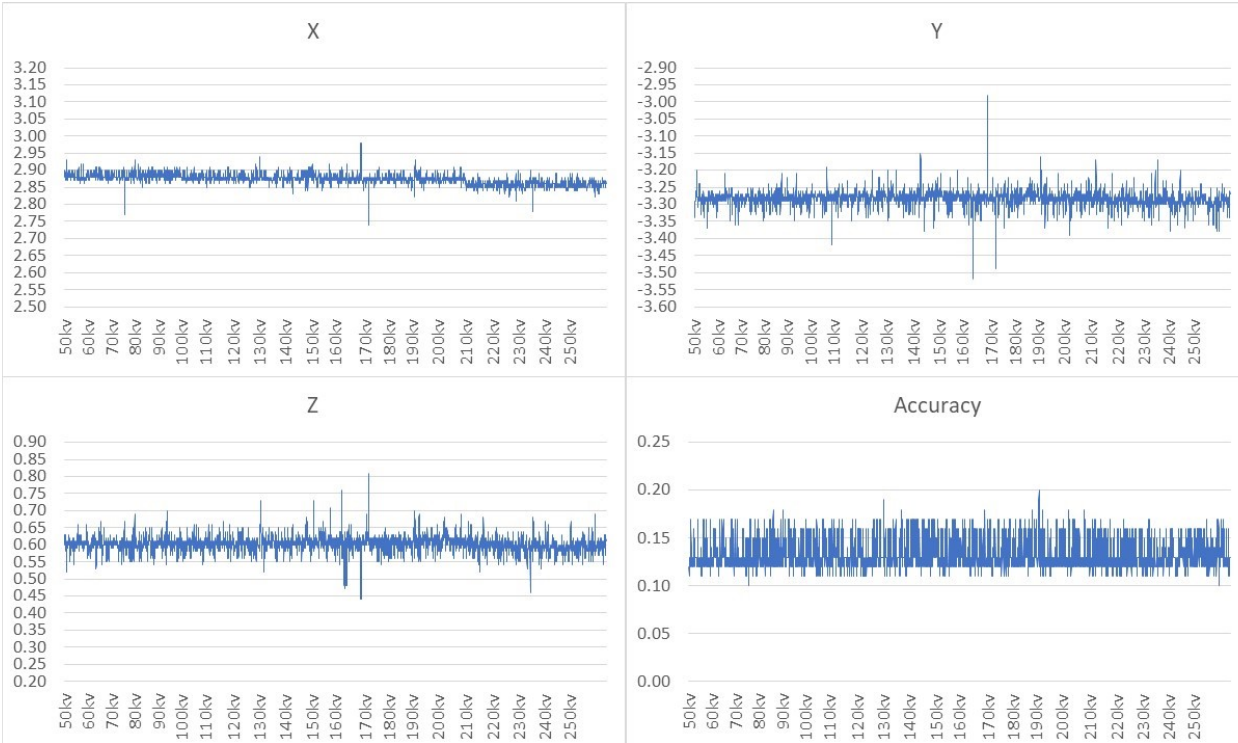


Figure A3 x, y, z values and the positioning accuracy of Tag-1

Table A2 lists the average values and standard deviations for Tag-1 over each test period. Figure A4 and Figure A5 show the average value and standard deviation for all data in each period. As shown in Figure A4, the averages of x, y, z and measurement accuracy provided by Quuppa system have no significant change when adding voltage. Standard deviations for the four indexes are less than 0.04 in all conditions. Even though an obvious increase of standard deviation showed up when the voltage reached 160 KV, the standard deviation is still low enough to be accepted. Therefore, the conclusion is that higher voltage does not have a significant effect on Quuppa system.

Table A2. Average and standard deviation of x, y, z value and the accuracy of tag 1

voltage	X_average	Y_average	Z_average	Accuracy_average	X_SD	Y_SD	Z_SD	Accuracy_SD
50kv	2.882	-3.286	0.602	0.130	0.011	0.019	0.018	0.013
60kv	2.882	-3.283	0.600	0.132	0.009	0.017	0.019	0.013
70kv	2.882	-3.286	0.601	0.129	0.013	0.016	0.021	0.013
80kv	2.883	-3.282	0.602	0.134	0.012	0.017	0.019	0.015
90kv	2.885	-3.284	0.606	0.132	0.010	0.018	0.019	0.014
100kv	2.878	-3.280	0.602	0.133	0.009	0.020	0.019	0.014
110kv	2.879	-3.281	0.605	0.132	0.010	0.016	0.019	0.014
120kv	2.878	-3.280	0.602	0.133	0.011	0.019	0.019	0.015
130kv	2.877	-3.281	0.608	0.132	0.010	0.020	0.022	0.015
140kv	2.878	-3.277	0.608	0.137	0.013	0.023	0.019	0.017
150kv	2.876	-3.277	0.612	0.135	0.012	0.019	0.021	0.017
160kv	2.874	-3.283	0.605	0.131	0.012	0.036	0.030	0.014
170kv	2.873	-3.282	0.608	0.133	0.012	0.026	0.024	0.014
180kv	2.874	-3.276	0.604	0.135	0.009	0.017	0.018	0.015
190kv	2.875	-3.282	0.612	0.132	0.013	0.026	0.023	0.016
200kv	2.874	-3.283	0.612	0.133	0.009	0.021	0.019	0.015
210kv	2.858	-3.287	0.600	0.129	0.008	0.023	0.022	0.016
220kv	2.858	-3.288	0.595	0.132	0.011	0.018	0.020	0.014
230kv	2.858	-3.287	0.594	0.130	0.011	0.024	0.025	0.012
240kv	2.857	-3.295	0.590	0.132	0.009	0.023	0.019	0.013
250kv	2.860	-3.293	0.595	0.132	0.010	0.021	0.019	0.014

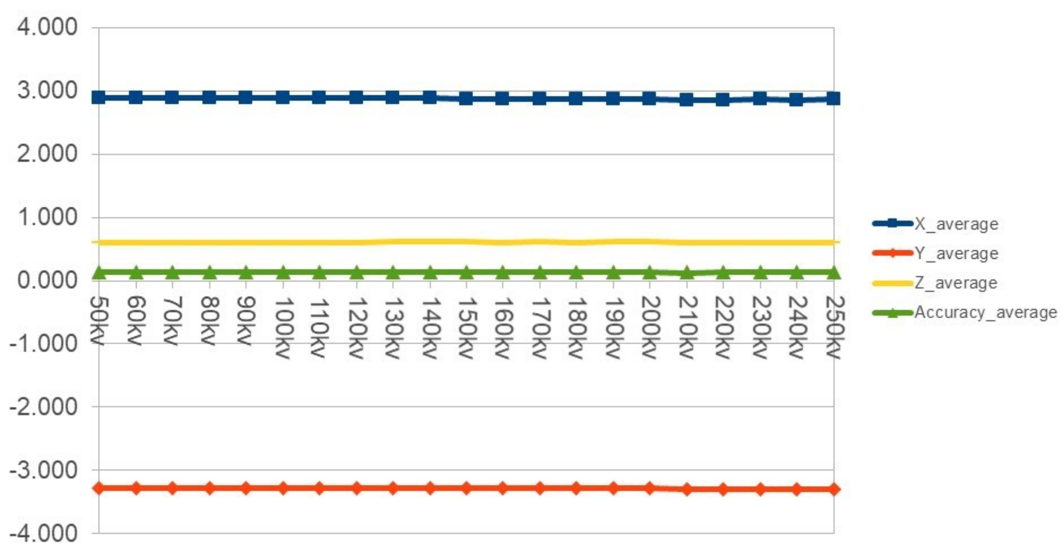


Figure A4. Average x, y, z value and the accuracy of Tag-1

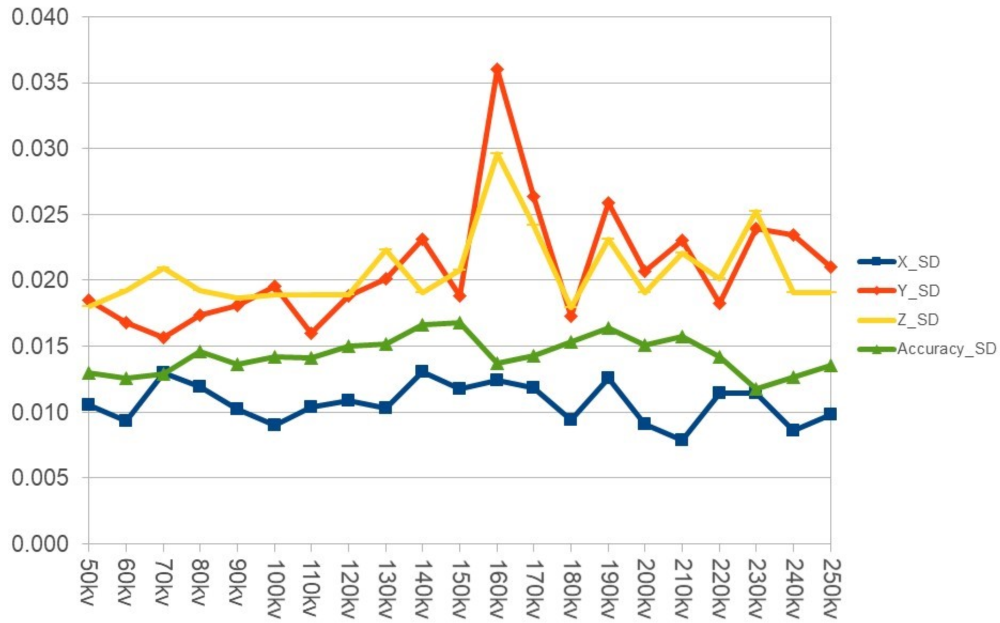


Figure A5. Standard deviation of x, y, z and accuracy of Tag11

Appendix B: Prototype System's Python Code

```
from datetime import datetime as dt
import requests
import json
from change_html import generate_html
import matplotlib

matplotlib.use('Qt5Agg')
import matplotlib.pyplot as plt
from pylab import *

mpl.rcParams['font.sans-serif'] = ['SimHei']
mpl.rcParams['axes.unicode_minus'] = False

from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
from PyQt5.QtWidgets import QWidget, QPushButton, QApplication
from PyQt5.QtCore import QTimer
import sys

from PyQt5 import QtCore, QtWidgets, QtGui
from PyQt5.QtWidgets import *
from PyQt5.QtWidgets import QWidget
from PyQt5.QtGui import QIcon
from PyQt5.QtCore import Qt


class App(QWidget):
    def __init__(self, parent=None):
        super(App, self).__init__(parent)
        self.initUI()


    def initUI(self):
        self.setWindowTitle('prototype')
        self.setFixedSize(1200, 700)
        self.setMinimumSize(1200, 700)
        self.setMaximumSize(1200, 700)

        # test
        self.testBtn2 = QPushButton('readdata')
        #self.testBtn2.clicked.connect(self.captured_data_from_api)
        self.testBtn2.clicked.connect(self.test1)
        self.test_position_data={}
        self.test_first_reading = True

        #schedule
```

```
        self.text_schedule=QLineEdit('./Book2.csv')
        self.group_info = {}
        self.groups = {}
        #proximities
        self.proximities={}

        # prototype system
        self.equip_list = list()
        self.worker_list = list()
        self.temp_equip_list = list()
        self.temp_worker_list = list()
        self.equip_dict = {}
        self.worker_dict = {}
        self.buffer=10 #buffer>=10/5
        self.calculation_ts=0
        self.start_ts = 0

        self.w_line={}
        self.e_line={}

        self.w_color_set = ['gold', 'royalblue', 'olivedrab', 'orange', 'gray', 'pink', 'darkgreen', 'lime',
'deepskyblue',
                    'tan']
        self.e_color_set = ['red', 'darkred', 'deeppink', 'm', 'purple']


        # buttons
        self.startBtn = QPushButton('start')
        self.startBtn.clicked.connect(self.start)
        self.endBtn = QPushButton('end')
        self.endBtn.clicked.connect(self.end)

        # graphic module
        self.figure = plt.figure()
        self.canvas = FigureCanvas(self.figure)
        # start/stop the system


        # time module
        self.t_timer = QTimer(self)
        self.t_timer.timeout.connect(self.step1)
        self.t_timer2= QTimer(self)
        self.t_timer2.timeout.connect(self.step2)



        # layout
        layout = QVBoxLayout()
        layout.addWidget(self.text_schedule)
        layout.addWidget(self.startBtn)
        layout.addWidget(self.endBtn)
```

```
    layout.addWidget(self.canvas)
    layout.addWidget(self.testBtn2)

    self.setLayout(layout)
    self.startBtn.setEnabled(True)
    self.endBtn.setEnabled(False)




def init_parametre(self):
    #############################
    # schedule
    self.text_schedule = QLineEdit('./Book2.csv')
    self.group_info = {}
    self.groups = {}
    # proximities
    self.proximities = {}

    # prototype system

    self.equip_list = list()
    self.worker_list = list()
    self.temp_equip_list = list()
    self.temp_worker_list = list()
    self.equip_dict = {}
    self.worker_dict = {}
    self.buffer = 10  # buffer>=10/5
    self.calculation_ts = 0
    self.start_ts = 0

    self.w_line = {}
    self.e_line = {}

    self.w_color_set = ['gold', 'royalblue', 'olivedrab', 'orange', 'gray', 'pink', 'darkgreen', 'lime',
                        'deepskyblue',
                        'tan']
    self.e_color_set = ['red', 'darkred', 'deeppink', 'm', 'purple']

# prototype system
def end(self):
    ax = self.figure.add_subplot()

    ax.clear()
    self.canvas.draw()

    self.t_timer2.stop()
    self.t_timer.stop()  # timer stop
    self.startBtn.setEnabled(True)
    self.endBtn.setEnabled(False)
```

```
    self.init_parametre()


def start(self):
    ms = 1
    print('test_start')
    while ms != 0:
        t = dt.now()
        ms = int(t.microsecond / 1000)
    print(dt.now())
    self.get_groups()
    self.start_ts = int(dt.now().timestamp()*1000)
    self.t_timer.start(1000)
    self.startBtn.setEnabled(False)
    self.endBtn.setEnabled(True)


def step1(self):
    self.t_timer2.stop()

    self.calculation_ts=float(dt.now().timestamp()*1000).__round__(0)

    t1=dt.now()
    #self.averaging_over_time2(self.temp_equip_list,self.equip_list,calculation_ts)
    #self.averaging_over_time2(self.temp_worker_list, self.worker_list, calculation_ts)

    self.data_processing(self.temp_equip_list, self.equip_dict, self.calculation_ts,self.buffer)
    self.data_processing(self.temp_worker_list, self.worker_dict, self.calculation_ts, self.buffer)
    t2=dt.now()
    print('time used for averaging over time ')
    print(t2-t1)

    """
    print('==================')
    print(dt.now())
    for item in self.worker_dict:
        print(item)
        print(self.worker_dict[item])

    for item in self.equip_dict:
        print(item)
        print(self.equip_dict[item])
    print('==================')
    '''
    self.plotting()
    self.reset_temp_list()

    #print(self.t_timer2.isActive())
    print('step1')

    #self.step2()
```

```
    self.t_timer2.start(185)



def step2(self):
    print('       step2')

    #print(dt.now())
    self.captured_data_from_api()



def captured_data_from_api(self):
    #print('update location')

    #generating API data
    t1 = dt.now()
    #capture data from API
    data_str = requests.get('http://localhost:8080/simulator/')
    #data_str=requests.get('http://localhost:8080/qpe/getTagPosition?version=2')
    data_json = json.loads(data_str.text)
    #print(data_json['responseTS'])
    tags_data = data_json['tags']

    for item in tags_data:
        type = item['name'].split('_')[0]
        object_name = item['name'].split('_')[1]
        tag_part = item['name'].split('_')[2]
        smoothed_posi = item['smoothedPosition']
        ts=item['positionTS']
        time=dt.fromtimestamp(int(item['positionTS'])/1000)
        #print(object_name+' '+tag_part)
        #print(smoothed_posi)
        self.record_temp_data(type,object_name,tag_part,smoothed_posi,ts)

    t2 = dt.now()
    #print('time used for captured data fro API:' )
    #print(t2-t1)
    #print(t2 - t1)

def record_temp_data(self,type,object_name,tag_part,smoothedPosition,time):

    if type == 'E':
        list = self.temp_equip_list
    if type =='W':
        list = self.temp_worker_list

    object_exist=False
    for item in list:
        if item['name']==object_name:
            object_exist=True
```

```
            add_posi = {
                'position': smoothedPosition,
                'time': time
            }
            if item['part'].keys().__contains__(tag_part):
                item['part'][tag_part].append(add_posi)
            else:
                item['part'][tag_part] = [add_posi]


        if object_exist==False:
            add_object= {'name':object_name,
                         'part':{}
                         }
            add_posi={
                'position':smoothedPosition,
                'time':time
            }
            add_object['part'][tag_part] = [add_posi]
            list.append(add_object)

    def reset_temp_list(self):
        self.temp_worker_list=[]
        self.temp_equip_list=[]

    def data_processing(self,temp_list,output_dict,calculation_ts,buffer):
        for item in temp_list:
            #print(item['name'])
            #calculate position
            if output_dict.keys().__contains__(item['name']):
                data_averaged_overtime = self.averaging_over_time(item,calculation_ts)
                """add_data = {'calculation_ts': calculation_ts}
                parts = list(item['part'])
                X = 0
                Y = 0
                Z = 0
                COUNT = 0
                for part in parts:
                    _x = 0
                    _y = 0
                    _z = 0
                    count = 0
                    for item2 in item['part'][part]:
                        _x = _x + item2['position'][0]
                        _y = _y + item2['position'][1]
                        _z = _z + item2['position'][2]
                        count = count + 1
                    X = X+_x / count
                    Y = Y+_y / count
                    Z = Z+_z / count
                    COUNT = COUNT + 1
```

```
            add_data['posi'] = [X / COUNT, Y / COUNT, Z / COUNT]'''
        else:
            output_dict[item['name']]=[]#CREAT LIST
            data_averaged_overtime = self.averaging_over_time(item,calculation_ts)

        output_dict[item['name']].append(data_averaged_overtime)

        if output_dict[item['name']].__len__()>buffer:
            del output_dict[item['name']][0]
            #print(item['name']+' len:'+ str(len(output_dict[item['name']])))

        #calculate speed
        if output_dict[item['name']][-1]['Missing_obj']==False:
            if output_dict[item['name']].__len__() > 2:
                self.calculating_speed(output_dict[item['name']])
            else:
                output_dict[item['name']][-1]['velocity'] = [0, 0]
        else:#compensate missing data
            del_obj = self.compensate_missing_obj(output_dict[item['name']])
            if del_obj==True:
                del output_dict[item['name']]

    def averaging_over_time(self,object_data,calculation_ts):
        add_data = {'calculation_ts': calculation_ts}
        parts = list(object_data['part'])
        X = 0
        Y = 0
        #Z = 0
        COUNT = 0
        for part in parts:
            _x = 0
            _y = 0
            #_z = 0
            count = 0
            for item2 in object_data['part'][part]:
                _x = _x + item2['position'][0]
                _y = _y + item2['position'][1]
                #_z = _z + item2['position'][2]
                if item2['position'][0]!=0 or item2['position'][1]!=0:
                    count = count + 1
            if count !=0:
                X = X + _x / count
                Y = Y + _y / count
                #Z = Z + _z / count
                COUNT = COUNT + 1
            if COUNT != 0 :
                #add_data['posi'] = [X / COUNT, Y / COUNT, Z / COUNT]
                add_data['posi'] = [X / COUNT, Y / COUNT]
                add_data['Missing_obj'] = False
            else:
                add_data['posi'] = [0, 0]
```

```
            add_data['Missing_obj'] = True

        return add_data

    def calculating_speed(self, target_list):
        new_posi = target_list[-1]['posi']
        new_ts = float(target_list[-1]['calculation_ts'])/1000
        old_posi = target_list[-2]['posi']
        old_ts = float(target_list[-2]['calculation_ts'])/1000
        #need change
        if ((new_posi[0]==0) & (new_posi[1]==0)) or ((old_posi[0]==0) & (old_posi[1]==0)):
            target_list[-1]['velocity']=target_list[-2]['velocity']
        else:
            v_x = (new_posi[0]-old_posi[0])/(new_ts-old_ts)
            v_y = (new_posi[1] - old_posi[1]) / (new_ts - old_ts)
            target_list[-1]['velocity']=[v_x,v_y]

    def compensate_missing_obj(self,target_list):

        target_list[-1]['velocity'] = [0, 0]
        target_list[-1]['posi'] = target_list[-2]['posi']

        compensate_range = 5 if len(target_list)<5  else len(target_list)

        for i in range(2,compensate_range+1):
            if target_list[-i]['Missing_obj']==False:
                target_list[-1]['velocity']=target_list[-2]['velocity']
                new_ts = float(target_list[-1]['calculation_ts']) / 1000
                old_ts = float(target_list[-2]['calculation_ts']) / 1000
                time_difference=new_ts - old_ts
                est_x = target_list[-2]['posi'][0]+target_list[-2]['velocity'][0]*time_difference
                est_y = target_list[-2]['posi'][1] + target_list[-2]['velocity'][1] * time_difference
                target_list[-1]['posi']=[est_x,est_y]
                return False

        for i in range (0,target_list.__len__()):
            if target_list[i]['Missing_obj']==False:
                return False
        return True

    def plotting (self):
        ax = self.figure.add_subplot()
        ax.clear()
        #set plotting parameter
        ax.set_xlim((0, 100))
        ax.set_ylim((0, 100))
        ax.set_xticks(np.arange(ax.get_xlim()[0], ax.get_xlim()[1], 10))
        ax.set_yticks(np.arange(ax.get_ylim()[0], ax.get_ylim()[1], 10))
        ax.set_aspect('equal')

        # register line data
```

```
for key in self.w_line:
    if self.worker_dict.keys().__contains__(key)==False:
        self.w_color_set.append(self.w_line[key][3])
        del self.w_line[key]

for item in self.worker_dict:
    #print(self.worker_dict[item])
    if self.w_line.keys().__contains__(item)==False:
        w_color = self.w_color_set[0]
        del self.w_color_set[0]

    else :
        w_color = self.w_line[item][3]
    self.w_line[item] = [[], [], [], w_color]  # x,y,t,color


    for item2 in self.worker_dict[item]:
        self.w_line[item][0].append(item2['posi'][0])
        self.w_line[item][1].append(item2['posi'][1])
        self.w_line[item][2].append(item2['calculation_ts'])
        if len(self.w_line[item][0])>5:
            del self.w_line[item][0][0],self.w_line[item][1][0],self.w_line[item][2][0]

    print(item)
    print(len(self.w_line[item][0]))
    print(self.w_line[item])


for key in self.e_line:
    if self.equip_dict.keys().__contains__(key)==False:
        self.e_color_set.append(self.e_line[key][3])
        del self.e_line[key]

for item in self.equip_dict:
    #print(self.equip_dict[item])

    if self.e_line.keys().__contains__(item) == False:
        e_color = self.e_color_set[0]
        del self.e_color_set[0]

    else:
        e_color = self.e_line[item][3]
    self.e_line[item] = [[], [], [], e_color]  # x,y,t,color

    for item2 in self.equip_dict[item]:
        self.e_line[item][0].append(item2['posi'][0])
        self.e_line[item][1].append(item2['posi'][1])
        self.e_line[item][2].append(item2['calculation_ts'])
        if len(self.e_line[item][0])>5:
            del self.e_line[item][0][0],self.e_line[item][1][0],self.e_line[item][2][0]
    print(item)
```

```
        print(self.e_line[item])
        print(len(self.e_line[item][0]))

    #proximity detection
    self.proximity_detection(2,5)

    #plot line
    for key in self.w_line:
        print( 'plotting '+key)
        ax.plot(self.w_line[key][0], self.w_line[key][1], label=key, c=self.w_line[key][3])
    for key in self.e_line:
        print( 'plotting '+key)
        ax.plot(self.e_line[key][0], self.e_line[key][1], label=key, c=self.e_line[key][3])
    ax.legend(bbox_to_anchor=(1.05, 1), loc='upper left', borderaxespad=0.)

    # creating posi of points
    point_w_x = [self.w_line[key][0][-1] for key in self.w_line]
    point_w_y = [self.w_line[key][1][-1] for key in self.w_line]
    point_e_x = [self.e_line[key][0][-1] for key in self.e_line]
    point_e_y = [self.e_line[key][1][-1] for key in self.e_line]
    color_w = [self.w_line[key][3] for key in self.w_line]
    color_e = [self.e_line[key][3] for key in self.e_line]
    W_points = ax.scatter(point_w_x, point_w_y, marker=".", s=5, c=color_w)
    E_points = ax.scatter(point_e_x, point_e_y, marker="D", s=15, c=color_e)

    #creat events text
    event_text = 'Event:\n'+self.creat_event_text()

    ax.text(ax.get_xlim()[1] + 2, (ax.get_ylim()[1] + ax.get_ylim()[0]) / 2, event_text,
                horizontalalignment='left', verticalalignment='top')

    #plot time
    p_text = str(dt.now().strftime('%Y-%m-%d %H:%M:%S'))

    ax.text(ax.get_xlim()[1], ax.get_ylim()[1], str(p_text),
                horizontalalignment='right', verticalalignment='bottom')

    if self.calculation_ts-self.start_ts>=3000:
        self.canvas.draw()

def read_schedule(self):
    path = self.text_schedule.text()
    self.text_schedule.setEnabled(False)
    print("reading schedule")
    with open(path) as myfile:
        data = myfile.read()
        lst = data.split('\n')
        lst.remove('')
        task = dict()

        # creat keys list
```

```
hours = len(lst[0].split(',')) - 2
_schedule_list = list()
_schedule_list.append('task')
_schedule_list.append('resources_type')
for i in range(2, hours + 2):
    _schedule_list.append(lst[0].split(',')[i])
    st = _schedule_list[i].split('~')[0]
    et = _schedule_list[i].split('~')[1]
    if len(st) < 5:
        st = '0' + st
    if len(et) < 5:
        et = '0' + et
    _schedule_list[i] = st + "~" + et

for j in range(1, len(lst)):
    temp_list = lst[j].split(',')
    if temp_list[0] == '':
        temp_list[0] = lst[j - 1].split(',')[0]
    for k in range(2, len(temp_list)):
        if temp_list[k].__contains__(';'):
            temp_elements = temp_list[k].split(';')
            temp_list[k] = temp_elements
        else:
            temp_list[k] = [temp_list[k]]

        if (temp_list[k] == [""]) & (k != 2):
            temp_list[k] = temp_list[k - 1]
        elif ((temp_list[k] == [""]) & (k == 2)):
            temp_list[k] = ['/']

    if task.keys().__contains__(temp_list[0]) != True:
        task[temp_list[0]] = {}
        for i in range(2, hours + 2):
            task[temp_list[0]][_schedule_list[i]] = {'Equipment': [], 'Worker': []}
            if temp_list[1] == 'Worker':
                task[temp_list[0]][_schedule_list[i]]['Worker'] = temp_list[i]
                if task[temp_list[0]][_schedule_list[i]]['Worker'] == ['/']:
                    task[temp_list[0]][_schedule_list[i]]['Worker'] = []
            elif temp_list[1] == 'Equipment':
                task[temp_list[0]][_schedule_list[i]]['Equipment'] = temp_list[i]
                if task[temp_list[0]][_schedule_list[i]]['Equipment'] == ['/']:
                    task[temp_list[0]][_schedule_list[i]]['Equipment'] = []

    else:
        for i in range(2, hours + 2):
            # task[temp_list[0]][_schedule_list[i]] = {'Equipment': [], 'Worker': []}
            if temp_list[1] == 'Worker':
                task[temp_list[0]][_schedule_list[i]]['Worker'] = temp_list[i]
                if task[temp_list[0]][_schedule_list[i]]['Worker'] == ['/']:
                    task[temp_list[0]][_schedule_list[i]]['Worker'] = []
            elif temp_list[1] == 'Equipment':
```

```
                    task[temp_list[0]][_schedule_list[i]]['Equipment'] = temp_list[i]
                    if task[temp_list[0]][_schedule_list[i]]['Equipment'] == ['/']:
                        task[temp_list[0]][_schedule_list[i]]['Equipment'] = []
        return task

    def get_groups(self):
        self.group_info=self.read_schedule()
        for item in self.group_info:
            for item2 in self.group_info[item]:
                _temp_group_info = self.group_info[item][item2]
                if _temp_group_info['Equipment']!=[] and _temp_group_info['Worker']!=[]:
                    if self.groups.keys().__contains__(item2)==False:
                        self.groups[item2]=[]
                    for equipment in _temp_group_info['Equipment']:
                        for worker in _temp_group_info['Worker']:
                            self.groups[item2].append([equipment,worker])

    def proximity_detection(self,D_Warning,D_Alert):
        #find the period
        # self.calculation_ts
        for period in self.groups:
            _ST = period.split('~')[0]
            _ET = period.split('~')[1]
            date = dt.now().strftime('%Y-%m-%d')
            ST = dt.fromisoformat(date + ' ' + _ST).timestamp()
            ET = dt.fromisoformat(date + ' ' + _ET).timestamp()
            if float(ST) <= float(self.calculation_ts) and float(self.calculation_ts) < float(ET):
                grouping_info = self.groups[period]
            elif float(self.calculation_ts) >= float(ET):
                grouping_info=[]
                break
            else:
                grouping_info=[]

        for equipment in self.e_line:
            print(len(self.e_line[equipment][0]))
            if len(self.e_line[equipment][0])<=2:
                continue

            e_x=self.e_line[equipment][0][-1]
            e_y=self.e_line[equipment][1][-1]
            e_t=self.e_line[equipment][2][-1]
            e_v_x = (self.e_line[equipment][0][-1] - self.e_line[equipment][0][-2]) / (
                    self.e_line[equipment][2][-1] - self.e_line[equipment][2][-2])
            e_v_y = (self.e_line[equipment][1][-1] - self.e_line[equipment][1][-2]) / (
                    self.e_line[equipment][2][-1] - self.e_line[equipment][2][-2])

            for worker in self.w_line:
                if len(self.w_line[worker][0])<=2:
                    continue
                w_x = self.w_line[worker][0][-1]
```

```
            w_y = self.w_line[worker][1][-1]
            w_t = self.w_line[worker][2][-1]
            w_v_x = (self.w_line[worker][0][-1] - self.w_line[worker][0][-2]) / (
                    self.w_line[worker][2][-1] - self.w_line[worker][2][-2])
            w_v_y = (self.w_line[worker][1][-1] - self.w_line[worker][1][-2]) / (
                    self.w_line[worker][2][-1] - self.w_line[worker][2][-2])

            if e_t!=self.calculation_ts or w_t!=self.calculation_ts: continue

            # if not in the same group
            if grouping_info.__contains__([equipment, worker]) == False:
                W_info = [w_x, w_y, w_v_x, w_v_y]
                E_info = [e_x, e_y, e_v_x, e_v_y]
                _results = self.detect_risks(W_info, E_info, D_Warning, D_Alert)
                # result: [proximity,case,distance, relative speed e to w]
                # if proximities happen
                if _results['case'] != 0:
                    temp_key = equipment + '_' + worker
                    append_obj = {'case': _results['case'], 'calculation_ts': e_t,
                                  'distance': _results['distance'], 'W_info': W_info,
                                  'E_info': E_info}
                    # record event
                    if self.proximities.keys().__contains__(temp_key) == False:
                        self.proximities[temp_key] = []
                        self.proximities[temp_key].append(append_obj)

                    elif self.proximities[temp_key][-1]['case'] < _results['case']:
                        self.proximities[temp_key].append(append_obj)
                        # activate alert
                        # upload event

                    elif int(self.calculation_ts) - int(self.proximities[temp_key][-1]['calculation_ts']) >= 10000:
                        self.proximities[temp_key].append(append_obj)
                        # activate alert
                        # upload event

                    if len(self.proximities[temp_key])>3:
                        del self.proximities[temp_key][0]



    def proximity_detection2(self, D_Warning, D_Alert):
        # find the period
        # self.calculation_ts

        for equipment in self.e_line:
            print(len(self.e_line[equipment][0]))
            if len(self.e_line[equipment][0]) <= 2:
                continue

            e_x = self.e_line[equipment][0][-1]
```

```
e_y = self.e_line[equipment][1][-1]
e_t = self.e_line[equipment][2][-1]
e_v_x = (self.e_line[equipment][0][-1] - self.e_line[equipment][0][-2]) / (
        self.e_line[equipment][2][-1] - self.e_line[equipment][2][-2])
e_v_y = (self.e_line[equipment][1][-1] - self.e_line[equipment][1][-2]) / (
        self.e_line[equipment][2][-1] - self.e_line[equipment][2][-2])

for worker in self.w_line:
    if len(self.w_line[worker][0]) <= 2:
        continue
    w_x = self.w_line[worker][0][-1]
    w_y = self.w_line[worker][1][-1]
    w_t = self.w_line[worker][2][-1]
    w_v_x = (self.w_line[worker][0][-1] - self.w_line[worker][0][-2]) / (
            self.w_line[worker][2][-1] - self.w_line[worker][2][-2])
    w_v_y = (self.w_line[worker][1][-1] - self.w_line[worker][1][-2]) / (
            self.w_line[worker][2][-1] - self.w_line[worker][2][-2])

    if e_t != self.calculation_ts or w_t != self.calculation_ts: continue

    for period in self.groups:
        _st = period.split('~')[0]
        _et = period.split('~')[1]
        date = dt.now().strftime('%Y-%m-%d')
        st = dt.fromisoformat(date + ' ' + _st).timestamp()
        et = dt.fromisoformat(date + ' ' + _et).timestamp()
        # if in the period
        if float(st) <= float(w_t) and float(w_t) < float(et):
            # if not in the same group
            if self.groups[period].__contains__([equipment, worker]) == False:
                W_info = [w_x, w_y, w_v_x, w_v_y]
                E_info = [e_x, e_y, e_v_x, e_v_y]
                _results = self.detect_risks(W_info, E_info, D_Warning, D_Alert)
                # result: [proximity,case,distance, relative speed e to w]
                # if proximities happen
                if _results['case'] != 0:
                    temp_key = equipment + '_' + worker
                    append_obj = {'case': _results['case'], 'calculation_ts': et,
                                  'distance': _results['distance'], 'W_info': W_info,
                                  'E_info': E_info}
                    # record event
                    if self.proximities.keys().__contains__(temp_key) == False:
                        self.proximities['temp_key'] = []

                    if self.proximities['temp_key'][-1]['case'] < _results['case']:
                        self.proximities['temp_key'].append(append_obj)
                        # activate alert
                        # upload event

                    elif int(et) - int(self.proximities['temp_key'][-1]['calculation_ts']) > 10:
                        self.proximities['temp_key'].append(append_obj)
```

```
                # activate alert
                # upload event

            if len(self.proximities['temp_key'] > 30):
                del self.proximities['temp_ket'][0]

def detect_risks(self,W_info, E_info,proximity,alert_distance):
    #x [1]
    #y [2]
    #vx[3]
    #vy[4]
    print('detecting event')
    result = dict()
    print('proximity: ' + str(proximity))
    print("alert_distance" + str(alert_distance))

    distance = np.sqrt((W_info[0] - E_info[0]) ** 2 + (W_info[1] - E_info[1]) ** 2)
    Unit_E_2_W = [(W_info[0] - E_info[0]) / distance, (W_info[1] - E_info[1]) / distance]
    project_E = E_info[2] * Unit_E_2_W[0] + E_info[3] * Unit_E_2_W[1]
    project_W = W_info[2] * Unit_E_2_W[0] + W_info[3] * Unit_E_2_W[1]
    v_relative_E2W = project_E - project_W

    result['distance']=distance
    result['relative speed'] = v_relative_E2W


    if distance > proximity:
        result['case'] = 0
    else:
        if distance <= alert_distance:
            result['case'] = 3
        elif v_relative_E2W > 0:  # becoming closer
            result['case'] = 2
        else:
            result['case'] = 1
    return result

    #result: [proximity,case,distance, relative speed e to w]

def creat_event_text(self):
    output_text = ''
    for key in self.proximities:

        #if self.calculation_ts-self.proximities[key][-1]['calculation_ts']<=10000:
        if self.calculation_ts - self.proximities[key][-1]['calculation_ts'] <= 2000:

            output_text = output_text + key + ': case' + str(self.proximities[key][-1]['case']) + '   ' + str(
                round(float(self.proximities[key][-1]['distance']),2))+' m\n'

    return output_text
```

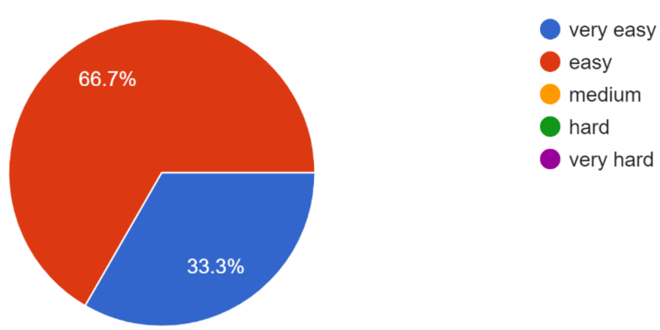
Appendix C: Survey on Proximity Alerts and Results

Survey

Questions					
	easy		normal		hard
Does the alert for case 1 easy to be perceived?	o	o	o	o	o
Does the alert for case 2 easy to be perceived?	o	o	o	o	o
Does the alert for case 3 easy to be perceived?	o	o	o	o	o
Please rank the three cases of alerts upon the urgency you can feel from the alerts	( )	>	( )	>	( )
	0	1	2	3	4
Please evaluate the impact the workloads have generate to the perception of alerts	o	o	o	o	o
	too short		proper		too long
How do you feel about the length of the alerts?	o		o		o
Do you have any comments or advice about the three alerts?	______________________________				

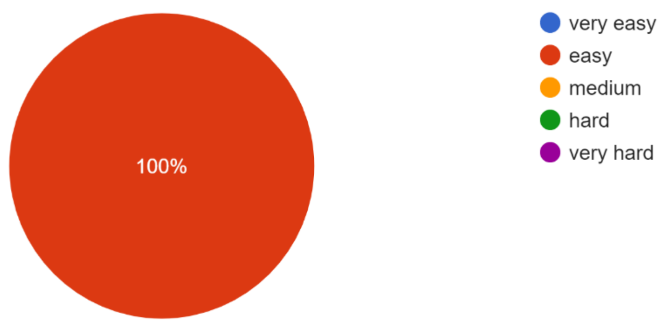
Results (extracted from Google Forms)

Does the alert for case 1 (the mildest alert) easy to be perceived?
（3 条回复）



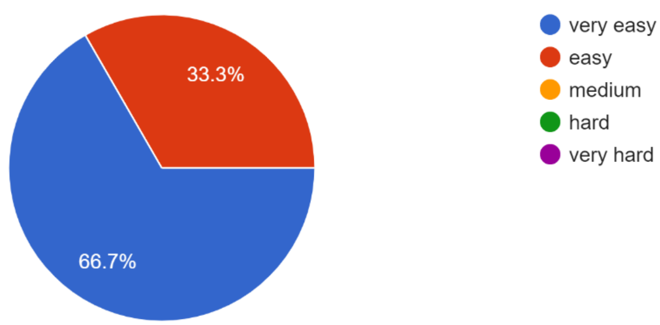
Does the alert for case 2 easy to be perceived?

（3 条回复）

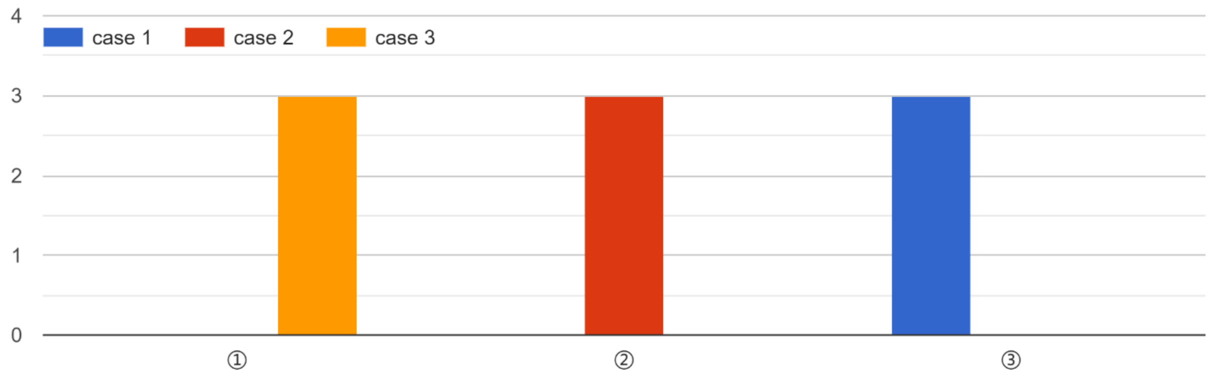


Does the alert for case 3 (the most intense alert) easy to be perceived?

（3 条回复）

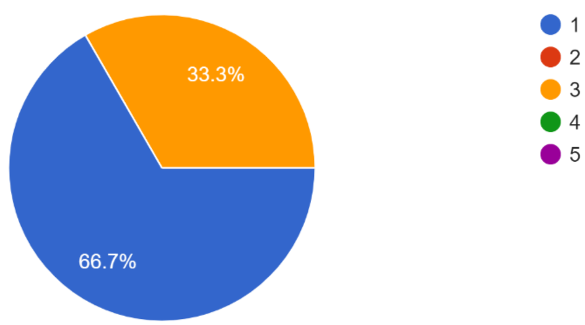


Please rank the three cases of alerts upon the urgency you can feel from the alerts. (① represents the most urgent one and ③ represents the mildest)



Please evaluate the impact the workloads have generate to the perception of alerts? ('1' represents small impact and '5' represents very serious impact)

（3 条回复）



How do you feel about the length of the alerts?
（3 条回复）