Models for Efficient Automated Site Data Acquisition

Magdy Ibrahim

A Thesis
In the Department of
Building, Civil, and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Building Engineering) at
Concordia University
Montreal, Quebec, Canada

July 2015

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Magdy Ibrahim**

Entitled: **Models for Efficient Automated Site Data Acquisition**

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

DOCTOR OF PHILOSOPHY (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

______________________________ External Examiner
Dr. C. Haas

______________________________ External to Program
Dr. A. Hammad

______________________________ Examiner
Dr. T. Zayed

______________________________ Examiner
Dr. Z. Zhu

______________________________ Thesis Supervisor
Dr. O. Moselhi

Approved by ______________________________
Dr. F. Haghighat, Graduate Program Director

July 13, 2015

____
Dr. A. Asif, Dean
Faculty of Engineering & Computer Science

ABSTRACT

Models for Efficient Automated Site Data Acquisition

Magdy Ibrahim, Ph.D.

Concordia University, 2015

Accurate and timely data acquisition for tracking and progress reporting is essential for efficient management and successful project delivery. Considerable research work has been conducted to develop methods utilizing automated site data acquisition for tracking and progress reporting. However, these developments are challenged by: the dynamic and noisy nature of construction jobsites; the indoor localization accuracy; and the data processing and extraction of actionable information. Limited research work attempted to study and develop customized design of wireless sensor networks to meet the above challenges and overcome limitations of utilizing off-the-shelf technologies.

The objective of this research is to study, design, configure and develop fully customized automated site data acquisition models, with a special focus on near real-time automated tracking and control of construction operations embracing cutting edge innovations in wireless and remote sensing technologies. In this context, wireless and remote sensing technologies are integrated in two customized prototypes to monitor and collect data from construction jobsites. This data is then processed and mined to generate meaningful and actionable information. The developed prototypes are expected to have wider scope of applications in construction management, such as improving construction safety, monitoring the condition of civil infrastructure and reducing energy consumption in buildings.

Two families of prototypes were developed in this research; Sensor Aided GPS (SA-GPS) prototype, which is designed and developed for tracking outdoor construction operations such as earthmoving; and Self-Calibrated Wireless Sensor Network (SC-WSN), which is designed for indoor localization and tracking of construction resources (labor, materials and equipment). These prototypes along with their hardware and software are encapsulated in a computational framework. The framework houses a set of algorithms coded in C# to enable efficient data processing and fusion that support tracking and progress reporting. Both the hardware prototypes and software algorithms were progressively tested, evaluated and re-designed using Rapid Prototyping approach. The validation process of the developed prototypes encompasses three steps; (1) simulation to validate the prototypes' design virtually using MATLAB, (2) laboratory experiments to evaluate prototypes' functionality in real time, and (3) testing on scaled case studies after fine-tuning the prototype design based on the results obtained from the first two steps.

The SA-GPS prototype consists of a microcontroller equipped with GPS module as well as a number of sensors such as accelerometer, barometric pressure sensor, Bluetooth proximity and strain gauges. The results of testing the developed SA-GPS prototype on scaled construction jobsite indicated that it was capable of estimating project progress within 3% mean absolute percentage error and 1% standard deviation on 16 trials, in comparison to the standalone GPS which had approximately 12% mean absolute percentage error and 2% standard deviation. The SC-WSN prototype incorporates two main features. The first is the use of the Kalman filtering and smoothing for the RSSI signal to provide more stable and predictable signal for estimating the distance between a reader and a tag. The second is the use of a developed dynamic path-loss model which continually optimizes its parameters to cope with the dynamically changing construction environment using Particle Swarm Optimization (PSO) algorithm. The laboratory testing indicated the improvement in location estimation, where the produced location estimates using SC_WSN had an average error of 0.66m in comparison to 1.67m using the raw RSSI signal. Also the results indicated 60% accuracy improvement in estimating locations using the developed dynamic model. The developed prototypes are not only expected to reduce the risk of project cost and duration overruns by timely and early detection of deviations from project plan, but also enables project managers to observe and oversee their project's status in near real-time. It is expected that the accuracy of the developed hardware, can be achieved on large-scale real construction projects. This is attributed to the fact that the developed prototype does not require any scalable improvements on its hardware technology, nor does it require any additional computational changes to its developed algorithms and software.

ACKNOWLEDGMENT

All praise and thanks are due to Allah for giving me the patience and perseverance to successfully accomplish my Ph.D. study. This thesis is dedicated to my family and friends for their endless support and encouragement throughout my life. Their unconditional love has been a source of inspiration. I could never achieve this work without their endless encouragement and help.

I would like to express my sincere gratitude and thanks to my supervisor Dr. Osama Moselhi for his guidance throughout my studies. It was a great honor to work under his supervision. I would like also to thank Dr. Mohamed Marzouk, Dr. Tarek Zayed and Dr. Zhenhua Zhu for their support during my research. I would like also to thank all my colleagues in the Automation and Construction Laboratory, especially Dr. Ahmed Atef, Dr. Ali Montaser, Dr. Hani Alzraiee and Dr. Ibrahim Bakry for their encouragement and assistance.

Table of Contents

List of Figures

List of Tables

Chapter 1 : INTRODUCTION

1.1 General

Timely and accurate data acquisition is essential for any effective project tracking and control system. Based on current practice in construction projects, two methods are commonly used for data acquisition: manual and semi-automated methods. These methods can be untimely, subjective and expensive. Considerable research work has been conducted to develop methods utilizing automated site data acquisition for tracking and progress reporting. However, these developments are challenged by: the dynamic and noisy nature of construction jobsites; the indoor localization accuracy; and the data processing, including extraction of actionable Information. Limited research work attempted to study and develop customized design of wireless sensor networks to meet the above challenges and overcome limitations of related off-the-shelf technologies.

In 2014, over than 1000 construction management professionals and practitioners from the USA and Canada, participated in a survey of IT technologies used to support construction project collaboration (JBKnowledge, 2014). The results indicated that 75% of survey respondents use a manual or spreadsheet process to collect and transfer data from construction jobsites as shown in Figure 1-1. Only 19.3% used a dedicated IT solution, most of them use mobile apps for smartphones or tablets that track one aspect of field operations, such as time entry, plan viewing, crew alert and GPS location.

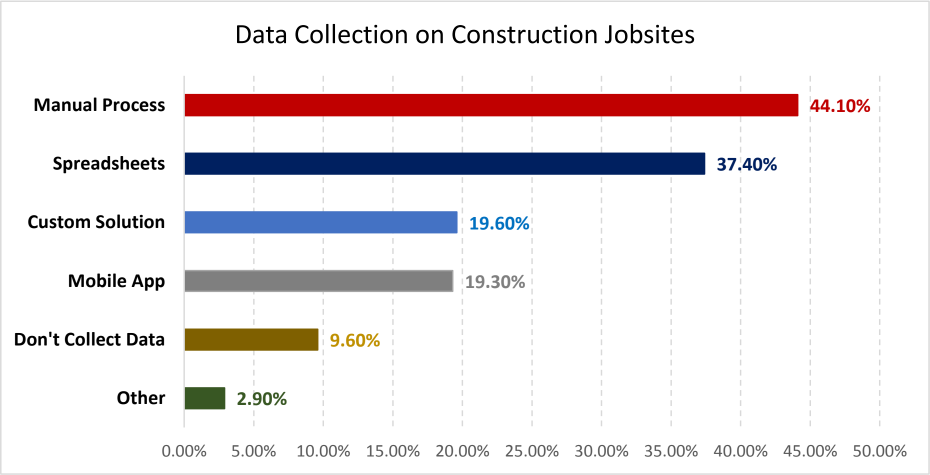


Figure 1-1: Data Collection on Construction Jobsites (JBKnowledge, 2014)

1.2 Problem Statement

Efficient management of construction operations relies on accurate and timely monitoring, tracking and reporting of onsite progress. There is considerable need to improve current practice and to advance research in automated site data acquisition to address this challenge. The limitations of the state of the art in automated site data acquisition articulate the problem in this field. These limitations in relation to this research are clustered in four main areas: outdoor tracking, indoor localization, data organization and processing, and hardware customizations.

For outdoor tracking applications, GPS and radio frequency identification (RFID) technologies had been utilized for tracking of earthmoving operations (Alshibani & Moselhi, 2010; Hildreth, Vorster, & Martinez, 2005; Montaser & Moselhi, 2012b; Navon & Shpatnitsky, 2005; Vahdatikhaki & Hammad, 2014). However, these methods suffer from the following limitations:

- Utilizing GPS data (location, speed, and distance) as the only source of information, might not be able to capture the big picture of the earthmoving operation nor accurately distinguish between productive, idle and out of service times.

- Utilization of fixed RFID readers at gates of loading and dumping sites is cumbersome in linear projects such as road and highway construction.
- GPS and RFID data cannot confirm whether the truck is actually fully loaded.
- The accuracy of these systems can be impacted with low GPS accuracy at urban construction jobsites.

For indoor localization applications, researchers experimented with multiple wireless technologies, such as radio frequency identification (RFID), ultra wideband (UWB) and wireless local area network (WLAN) (Ergen & Akinci, 2007; H. M. Khoury & Kamat, 2009; Montaser & Moselhi, 2014; Saiedeh Navabzadeh Razavi, 2010; Rueppel & Stuebbe, 2008; Soltani, Motamedi, & Hammad, 2013; Teizer, Lao, & Sofer, 2007). Several challenges with indoor localization had been identified:

- Passive RFID tags suffer from their short read range, which entails the deployment of a large number of tags and hence additional cost (N. Li & Becerik-Gerber, 2011).
- Ultra wideband (UWB) commercially available hardware is very costly and the deployment requires installation of fiber optic cables for timing synchronization (Aryan et al., 2011).
- Wireless local area network (WLAN) accuracy had been reported by researchers to be low; approximately 4–7 m with 97% confidence (Bahl & Padmanabhan, 2000; Deasy & Scanlon, 2004; Elnahrawy, Xiaoyan, & Martin, 2004; H. Khoury & Kamat, 2007; H. M. Khoury & Kamat, 2009; Woo et al., 2011).
- Utilization of static path-loss models to estimate distances between readers and tags, is not practical for dynamic and continually changing construction environment.

As to data organization and processing, various data sources are utilized to capture a complete picture of the on-going construction operations, and to extract actionable information. The utilization of multiple sources of data provides a large amount of data (Soibelman, Wu, Caldas, Brilakis, & Lin, 2008). However, processing and reducing data to actionable information and fusing the data from different sources remains as obstacles to achieving a practical automated progress tracking in near real-time.

For the hardware customization, most of previous research work focused on the use of off-the-shelf data acquisition technologies. Such use of one size fits all may not be efficient and incapable of addressing the targeted needs.

This research is motivated to utilize emerging advances in mobile computing, wireless communication and remote sensing technologies, to design and develop fully customized and cost effective automated site data acquisition tools addressing the above stated challenges.

1.3 Research Objectives

To address the challenges highlighted above, this research aims to study, design, configure and develop fully customized automated site data acquisition solutions, with a special focus on automated tracking and control of construction projects embracing cutting edge innovations in wireless and remote sensing technologies. This is to be achieved through the following sub-objectives:

1- Study previous research efforts made in the area of automated site data acquisition and identify gaps and limitations of these efforts, and the challenges in the use of related off-the-shelf technologies. Explore and experiment with wireless sensor networks (WSN) and internet of things (IoT) for possible use in this field.

2- Design and configure prototypes for outdoor construction progress tracking and control with a special focus on earthmoving operations, along with productivity assessment and analysis algorithms.

3- Select most suitable wireless technology for indoor localization and best RSSI smoothing algorithm, along with the development of a dynamic path-loss model to improve the indoor localization accuracy.

4- Develop self-adaptive algorithm for forecasting project cost at completion, and an algorithm for automated generation of as-built schedules.

5- Test and validate functions of the developed prototypes using MATLAB simulation and experiments in laboratory and outdoors environments.

1.4 Research Methodology

Figure 1-2 illustrates the methodology to achieve the objective of this research. The methodology is summarized in four stages: analysis stage, conceptual design stage, detailed design and experimental validation stage, conclusion and documentation stage. The analysis stage began with a problem statement and the definition of the objectives. Then, it focuses on performing a comprehensive state of the art review on the following domains:

- Current practices in project tracking and control.
- Automated data acquisition technologies and techniques.
- Emerging new technologies for automated site data acquisition.
- Data fusion models.

The conceptual design of the framework was developed embracing flexibility and adaptability. It encapsulates customized hardware prototypes and software algorithms for tracking and progress reporting of construction operations. Simulation and laboratory experiments are utilized to test, evaluate and re-design prototypes and algorithms using Rapid Prototyping approach.

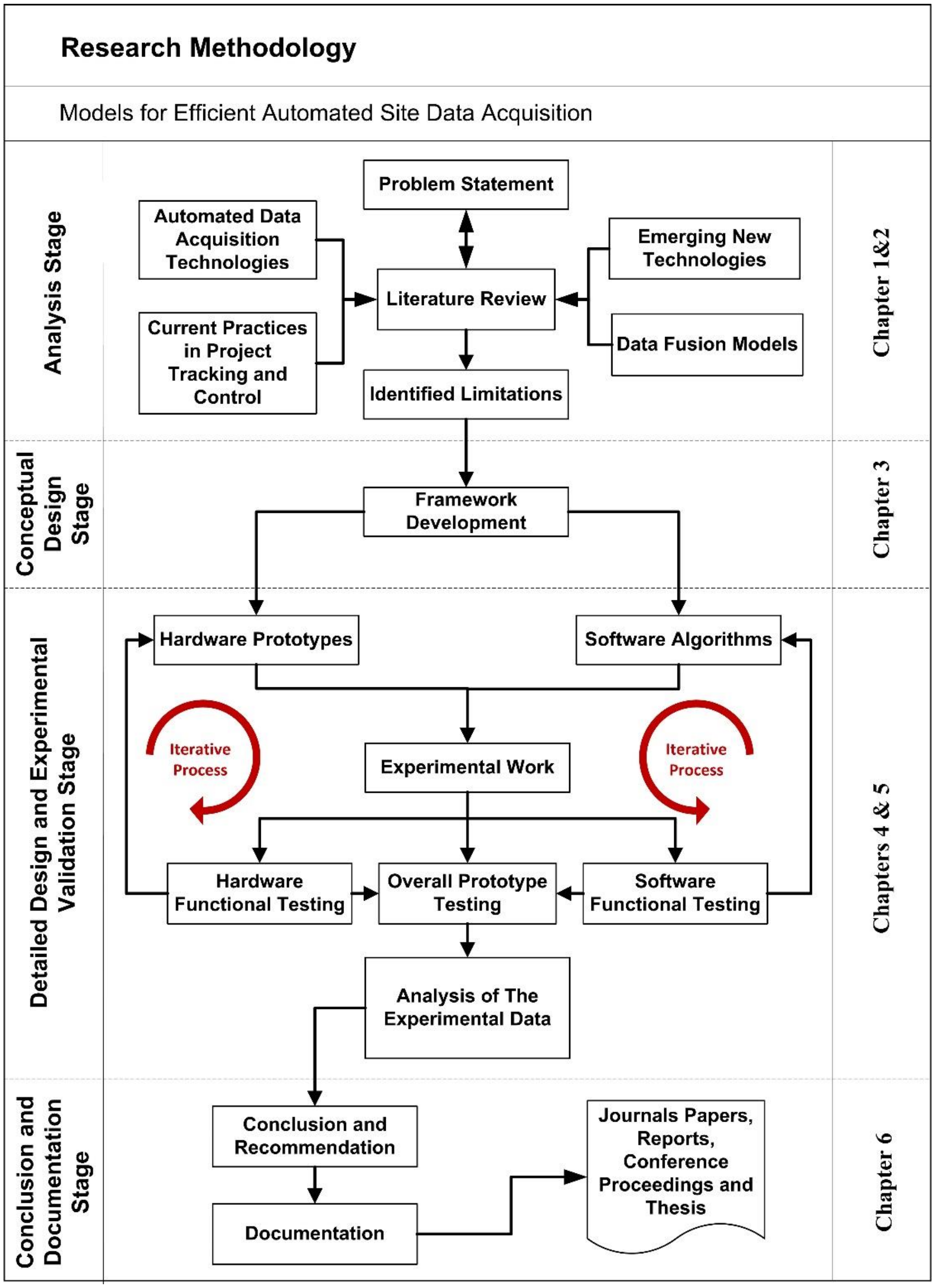

Research Methodology
Models for Efficient Automated Site Data Acquisition
Analysis Stage
Problem Statement
Automated Data Acquisition Technologies
Emerging New Technologies
Literature Review
Current Practices in Project Tracking and Control
Data Fusion Models
Identified Limitations
Chapter 1&2
Conceptual Design Stage
Framework Development
Chapter 3
Detailed Design and Experimental Validation Stage
Hardware Prototypes
Software Algorithms
Iterative Process
Experimental Work
Iterative Process
Hardware Functional Testing
Overall Prototype Testing
Software Functional Testing
Analysis of The Experimental Data
Chapters 4 & 5
Conclusion and Documentation Stage
Conclusion and Recommendation
Journals Papers, Reports, Conference Proceedings and Thesis
Documentation
Chapter 6

Figure 1-2: Research Methodology

1.5 Thesis Organization

This thesis consists of six chapters and seven appendices. Chapter 2 presents the literature review Summary of the limitations and gaps in previous research is presented at the end of that chapter. Chapter 3 presents the research vision, along with the conceptual design of the developed framework. An overview of developed framework is introduced, describing its main hardware prototypes and software algorithms. Chapter 4 presents a detailed design, testing and validation of the developed sensor aided GPS prototype (SA-GPS) for outdoor tracking of construction operations. Chapter 5 presents detailed design, testing and validation of Self-Calibrated Wireless Sensor Network (SC-WSN) for indoor localization. Chapter 6 highlights contributions and limitations of the developments made in the thesis along with suggested future research work.

Chapter 2 : LITERATURE REVIEW

2.1 General

This Chapter provides a literature review on current practices in data collection and analysis using EV tool for track and control construction projects. It also presents an overview of previous research efforts on automated data acquisition technologies and their implementation in the construction industry. A state of the art review for emerging new technologies such as wireless sensor networks (WSN) and data fusion models and their applications in construction management. Furthermore, indoor localization techniques are described and their applicability on construction jobsites. Finally, the identified gaps and limitations are outlined. Figure 2-1 illustrates the structure of this chapter.

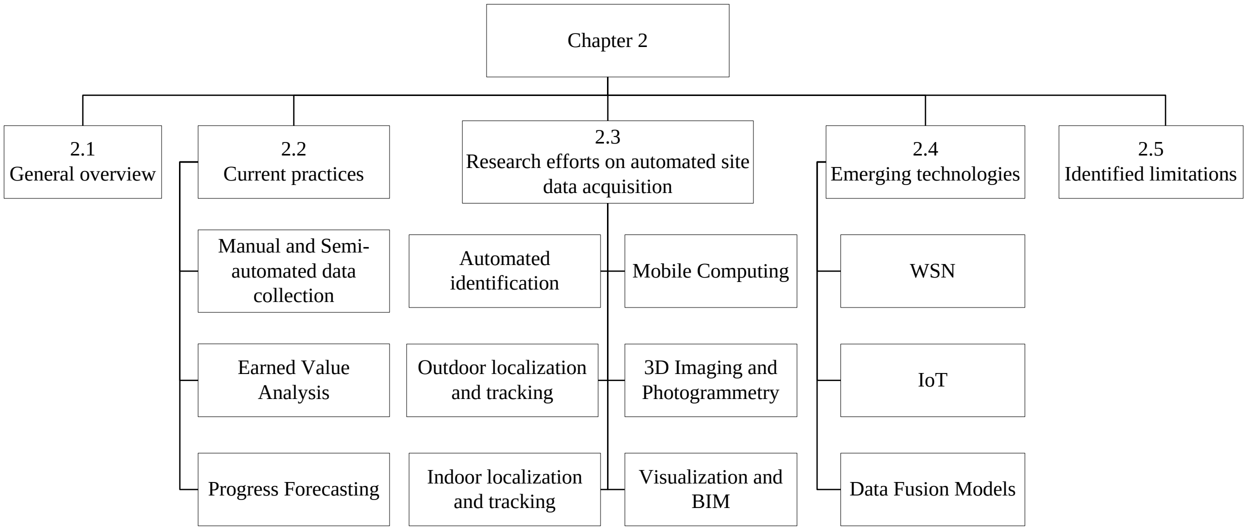

Chapter 2
2.1 General overview
2.2 Current practices
2.3 Research efforts on automated site data acquisition
2.4 Emerging technologies
2.5 Identified limitations
Manual and Semi-automated data collection
Earned Value Analysis
Progress Forecasting
Automated identification
Outdoor localization and tracking
Indoor localization and tracking
Mobile Computing
3D Imaging and Photogrammetry
Visualization and BIM
WSN
IoT
Data Fusion Models

Figure 2-1: Chapter 2 Structure

2.2 Current Practices in Project Tracking and Control

Traditionally, Earned Value Management (EVM) is conducted, integrating time and cost to measure the performance of a project by comparing its planned performance to the actual work performed on construction jobsites. EVM works by comparing three curves to display and evaluate the project performance. These curves typically have S shape, and they namely are: Budget Cost for Work Scheduled (BCWS), Actual Cost for Work Performed (ACWP), and Budget Cost for Work Performed (BCWP), as shown in Figure 2-2 (J. Li, 2004). The curve representing the base line planned project is the BCWS curve. The actual expenditure to date is represented through the ACWP curve. The third and final curve is the BCWP; it represents the budgeted cost of the work that is performed to date, which is the actual value earned for the project.

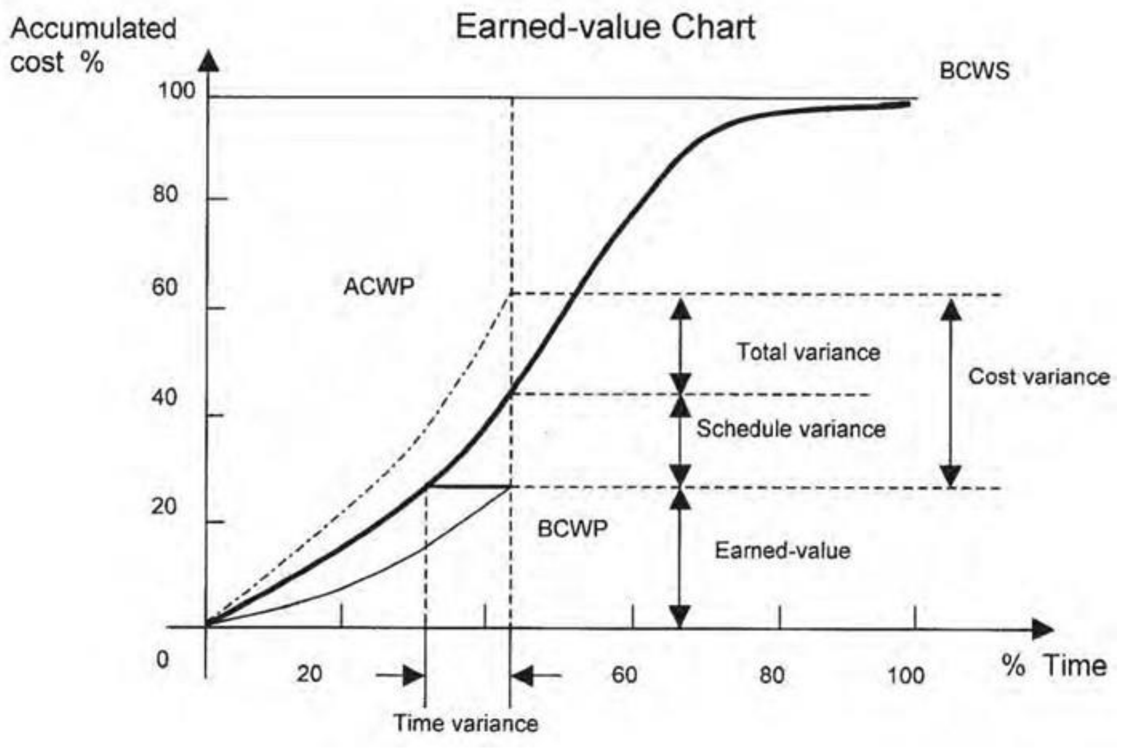


Figure 2-2: Earned Value Method (J. Li, 2004)

The main challenge in applying EV is the calculation of the Budget Cost for Work Performed (BCWP). Current practices utilize manual and semi-automated method for site data collection, then the data is analyzed to estimate the percentage of actual work completed. Project controllers also utilize templates to measure percentage complete of various construction tasks and/or process on jobsites. Templates are developed to represent a set of control points and earning percentages for each process as shown in Table 2-1 (Moselhi, 1993). However, manual and semi-automated data collection and analysis approaches are subjective, expensive and time consuming. Studies estimated that field supervisory personnel spend between 30-50% of their time on construction sites recording and analyzing field data (McCullouch, 1997) and 2% of the work on construction sites is devoted to manual tracking and recording of progress data (Cheok et al., 2000). Such practice leads to ineffective project management and creates the need for automated solutions that are accurate, efficient, timely and autonomous with minimal user intervention (Sacks et al., 2005).

Table 2-1: Earned Value Template (Moselhi, 1993)

Task	Work Content	Cumulative % Earned
Earthworks		
01	Excavation	50
02	Backfill	60
03	Compact	90
04	Fine grade	95
05	Hand over	100
Foundations		
01	Building	5
02	Formwork	50
03	Rebar and embedment	80
04	Pour	87
05	Strip cure and grout	95
06	Hand over	100

Forecasting is a frequent task performed by project managers from start to completion of their projects. Accurate forecasts enable project management teams to early estimate potential impacts on cost and/or schedule, and hence provide a room to take timely and necessary corrective actions. Several forecasting methods for project cost and duration at completion were introduced.

These methods can be categorized as detailed estimates of remaining work, index-based, stochastic, regression-based, and artificial intelligence methods. Detailed cost estimation method is used to estimate the cost of the remaining work, and hence forecast the project cost at completion. The accuracy of this method is very high, however, it requires substantial effort for detailed quantity takeoffs of remaining work items and estimating their costs.

As for the index-based forecasting methods, different equations working under different assumptions have been utilized to forecast project cost and duration at completion, as shown from Equations (2-1) till (2-5) (Hassanien, 2002; J. Li et al., 2006; Moselhi et sl., 2004). However, such method suffers from its general assumption that either the performance efficiency achieved up to the reporting date remains unchanged throughout the rest of the project, or that the performance will be as planned beyond the reporting date (Alshibani, 1999; Christensen, 1993; Fleming & Koppelman, 2000; Moselhi, 2011; Zwikael et al., 2000). Better forecasts can be achieved by removing specific time periods during which exceptional conditions are known to have prevailed and are not likely to be repeated beyond the reporting date (Hassanein & Moselhi, 2003).

$$\mathrm{EAC} = \frac{\mathrm{BAC}}{\mathrm{CPI}} \tag{2-1}$$

$$\mathrm{D} = \frac{\mathrm{PD}}{\mathrm{SPI}} \tag{2-2}$$

$$\mathrm{EAC} = \frac{\mathrm{BAC}}{\mathrm{CPI}\times\mathrm{SPI}} \tag{2-3}$$

$$\mathrm{D} = \frac{\mathrm{PD}}{\mathrm{CPI}\times\mathrm{SPI}} \tag{2-4}$$

$$\mathrm{EAC} = \mathrm{ACWP}\ (\text{to date}) + \frac{\mathrm{BAC}-\mathrm{BCWS}(\text{to date})}{\text{index}} \tag{2-5}$$

Where: BAC is Budget at Completion, D is duration at completion, and PD is planned duration.

For the stochastic techniques, the variability in the cost of individual activities within Work Breakdown Structure (WBS) is expressed in terms of stochastic S curves, which were utilized to provide probability distributions of forecasted cost at completion (Acebes et al., 2014; Alshibani & Moselhi, 2012; Barraza et al., 2004; Caron et al., 2013). However, these stochastic methods are more suitable when historical data of similar projects are available and when the duration of the project is relatively long to establish distributions of the activities and their respective remaining work.

As for regression based methods, the relationship between the actual cost and time is modelled to predict the project's Estimate-At-Completion (EAC) (Aliverdi et al., 2013; Narbaev & De Marco, 2014). However, for short term forecasting or with a limited number of observations, regression analysis is impractical. Also the performance of such models depends mainly on the availability of historical data of similar projects, and the correlation between the data used in the analysis.

For artificial intelligence methods, fuzzy logic and artificial neural networks (ANN) were utilized to forecast project cost and duration at completion under uncertainty (Iranmanesh & Mokhtari, 2008; J. Li et al., 2006; Naeni et al., 2011). However, the utilization of fuzzy logic requires an expert’s knowledge to express the relationship between the linguistic terms and fuzzy numbers. Also, the application of ANN requires a large historical data for training and testing of the network.

A recent study explored previous research works for forecasting cost at completion in both construction and other industries, revealed that the construction industry has difficulties in adequately importing advanced tools, and highlighted the need for future research emphasizing the adoption of methods from other experienced fields to reflect more reliable and consistent cost forecasting capabilities (Narbaev & Marco, 2011).

2.3 Automated Data Acquisition

Previous research work on automated data acquisition in construction operations can be grouped in five main categories: Automated identification technologies; Outdoor localization and tracking; Indoor localization and tracking; Mobile computing; 3D Imaging and Photogrammetry; and Visualization and Building Information Modeling. Each of these categories is explained in details in the following sections highlighting their advantage and limitations.

2.3.1 Automated Identification

Barcodes and Radio Frequency Identification (RFID) are used for tracking progress of structural steel erection (Cheng & Chen, 2002), on-site data collection and information sharing between project participants (Tserng et al., 2005), and tracking of material delivery (Akinci et al., 2002; Jaselskis et al., 1995; Lee et al., 2008; Montaser, 2013; Song et al., 2006). However, these technologies suffer from their limited read range, where barcodes read range is about few inches and passive RFID read range is (3-5) meters. Also the cost of RFID readers is quite high where the reader costs around $1500. Furthermore, the manual scanning and data analysis process is time consuming.

2.3.2 Outdoor Localization and Tracking

Global positioning system (GPS), radio frequency identification (RFID), ultra-wideband (UWB), attitude and heading reference system (AHRS), and vision based technologies have been utilized for progress tracking of outdoor construction operations.

Standalone GPS technology was utilized to track earthmoving operations (Alshibani & Moselhi, 2007; Hildreth et al., 2005; Montaser et al., 2012; Navon & Shpatnitsky, 2005), and tracking the position of pipe spools on a construction project (Caldas et al., 2006). However, there are a number of shortcomings associated with the usage of standalone GPS. The acquired data are limited to time and location, which sometimes makes it difficult to distinguish between productive and idle times. Also, these records do not provide enough to estimate the quantities of the excavated soil or confirm that the trucks are fully loaded.

As for Passive RFID, fixed readers installed at the entrance gates of loading and dumping sites were utilized for progress tracking of earthmoving operations, by recording the entrance and exit of RFID tags attached to dump trucks (Montaser & Moselhi, 2012b). However, passive RFID has a limited read range of 3 to 5 meters, which is not suitable for implementation in mega earthmoving operations or linear projects such as road and highway construction, where there is no entrance gates to the cut or fill locations, and equipment many entries the jobsite from several directions without detection. Furthermore, RFID alone cannot detect the truck load, and in that study, the dump truck was always assumed to be fully loaded.

For UWB, a combination of UWB integrated with AHRS was used to collect and process earthmoving equipment data, and update corresponding discrete event earthmoving simulation model (Akhavian & Behzadan, 2013). However, the utilization of UWB for location tracking specially on hauling routes is not applicable for real road construction projects due to the limited range of the UWB (100–200 m) (Teizer et al., 2007). Also, the results in this study shown several errors regarding the loader ideal time and the operation total cost (Ibrahim & Moselhi, 2014c). A novel framework for near real-time simulation of earthmoving operations based on the application of UWB and/or GPS tracking technologies was developed encompassing a rule set to capture details of truck and excavator operations (Vahdatikhaki & Hammad, 2014). However, their prototype required some manual data handling and processing for inputting real time location system (RTLS) files and averaging data over time.

As for computer vision–based methods, several studies have used video processing, including object tracking (Brilakis et al., 2011; Kim et al., 2011; Park et al., 2011; Golparvar-Fard et al., 2013) and object recognition (Chi & Caldas, 2011; Jog et al., 2011; Azar et al., 2013; Azar & McCabe, 2012). However, computer vision methods can fail under certain conditions, particularly in visually noisy images common to construction sites. Also, these methods are limited to the use of one camera, which is not adequate to cover large construction jobsites.

2.3.3 Indoor Tracking and Localization

In the construction management domain, several researchers have investigated indoor localization using a wide range of technologies, which can be divided into three main categories as shown in Figure 2-3: (1) wave propagation based; (2) image based; and (3) motion based.

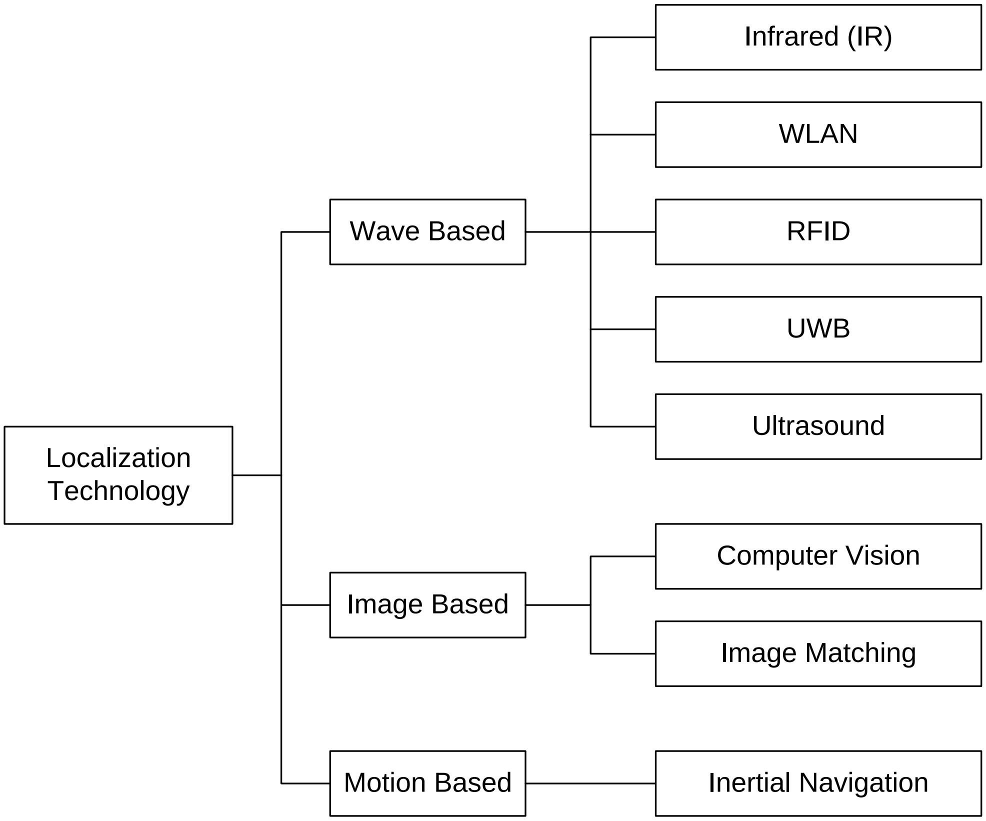


Figure 2-3: Indoor Localization Technologies

For wave propagation based technologies, propagation models are utilized to estimate the distance between a transmitter unit and a receiver unit. Researchers experimented with multiple wireless technologies specially radio frequency identification (RFID), ultra wideband (UWB), wireless local area network (WLAN). Each technology has its own inherited advantage and disadvantage with relative to accuracy, cost, coverage range, deployment requirements and scalability (Mahalik, 2007). RFID had been used for object tracking without localization (Goodrum, McLaren, & Durfee, 2006; Jaselskis et al., 1995) and tracking with localization (Ergen & Akinci, 2007; Montaser & Moselhi, 2014). N. Li & Becerik-Gerber (2011) reported that passive RFID tags are a cost effective solution for indoor localization, however, they suffer from their short read range, which entails the deployment of a large number of tags and hence requires additional cost. Research studies utilizing ultra wideband (UWB) had reported higher localization accuracy of approximately < 1m (H. M. Khoury & Kamat, 2009; Rueppel & Stuebbe, 2008; Teizer et al., 2007), however the measurement accuracy is highly dependent upon the line of sight of the point to be located (Aryan et al., 2011). Furthermore, cost of commercially available hardware is very high. WLAN is an attractive solution for indoor localization due to its existing universal infrastructure availability (Mazuelas et al., 2009). However, several researchers have reported its low accuracy to be approximately 4–7 m with 97% confidence (Bahl & Padmanabhan, 2000; Deasy & Scanlon, 2004; Elnahrawy et al., 2004; H. Khoury & Kamat, 2007; H. M. Khoury & Kamat, 2009; Woo et al., 2011). W. Jang & Skibniewski (2007) implemented combined radio frequency and ultrasound architecture using ZigBee wireless sensor modules for indoor position estimation. However, traditional ultrasound positioning is limited by line of sight, which is challenging in complicated construction environments (Shen et al., 2008). Combinations of RFID and ZigBee based sensor networks had been experimented by researchers for materials tracking and supply chain management (Cho, Kwon, Shin, Chin, & Kim, 2011; Shin, Park, & Kwon, 2007). In these studies, RFID tags were used for identification of construction materials, and ZigBee communication was used for wireless data transfer. While, wireless sensor network (WSN) was only used in these studies for data transfer, they confirmed the positive contribution of WSN on communication and network flexibility.

Shahi et al. (2013) developed a 3D marking method using UWB positioning system to track the progress of both structural and non-structural activities on construction projects. The developed method is able to quantify the progress of the activities that are not directly associated with the addition or removal of physical entities on a site, such as the welding or inspection of pipe-spools. This method also addresses another shortcoming of existing object-based models with respect to detecting progress for situations where the as-built location is different than the as-planned location. Therefore, the 3D marking approach can be used as a progress data source in progress tracking models, providing a unique dimension of site information that has not been incorporated in the previous attempts at automating construction progress tracking. However, the utilization of UWB has a main limitation due to wiring, calibration, and security, as explained in detail in (Aryan et al., 2011).

Montaser & Moselhi (2014) presented a detailed methodology on utilizing a low cost location identification and material tracking for indoor construction using a two-step algorithm. Their proposed method utilizes UHF passive RFID technology for capturing spatial data in an indoor environment. In this study, the work-active area is divided into exclusive zones, and each zone is spatially covered with a number of passive RFID tags. The user and material locations are estimated using two different RFID methods (triangulation and proximity) based on RSSI signal measurement. A specially designed relational database was used to store and organize RFID captured signals. The methodology was experimented on a construction facility in Montreal and a lab environment. The results were compared with 5 different test beds in different construction time intervals and 1 test bed in a lab environment. The results showed a mean error of 1.0 m and 1.9 m for user location identification and material tracking using the triangulation method, respectively. The results showed a mean error of 1.9 m and 2.6 m for user location identification and material tracking using the proximity method, respectively. The main limitations of the developed methodology are the need to generate a path-loss model for each type of tag used in case of using the triangulation method, the variability associated with deployment of tags, the uncontrolled influence of noisy signals and potential interference from equipment and/or vehicles located between the tags and between tags and the mobile reader.

Soltani et al. (2013) investigated the usage of active RFID technology for the localization of movable objects (e.g. components, equipment, and tools) equipped with RFID tags using handheld readers by extending a Cluster-based Movable Tag Localization (CMTL) technique which uses a k-Nearest Neighbor (k-NN) algorithm. CMTL uses a multidimensional clustering technique that considers the signal pattern similarity between the target and reference tags together with the spatial distribution of reference tags for detecting the region where the target tag is located. The proposed method uses artificial neural networks (ANNs) for positioning the target tag, as opposed to empirical weighted averaging formulas used in similar k-NN based methods. The proposed method adapts its performance to the environment and reduces the deployment cost of dense RFID-tagged environment, while achieving high accuracy by adding virtual tags (VRTs). The developed method was tested during the operational phase of a facility using 20 target tags, the localization results were compared to the LANDMARC and CMTL methods. The average errors considering only centered tags inside each room, which are surrounded by four tags were 1.55 m, 0.77 m, and 0.38 m for LANDMARC, CMTL, and CMTL+, respectively. However, the developed method showed an improvement of 38% compared to LANDMARC, and 7% compared to CMTL considering all the tags. The developed method utilizes irregular bilinear interpolation algorithm to simulate the RSSI for the grid of VRTs surrounded by real reference tags. The basic principle of the bilinear interpolation is that the 2-D interpolation is broken down into three 1-D interpolations. While the algorithm accounts for nonlinearity in 2D space, it assumes that the RSSI distribution is linear in 1D space. Moreover, ANN is applied as an alternative method for positioning, where RSSIs are processed and then summarized into dissimilarity indicators (β values). However, it is assumed that the target tags are stationary for the period of the training the ANN after each data collection step and localization.

Li et al. (2015) proposed the development and application of a real time locating system (RTLS) based on the chirp spread spectrum (CSS) technique, which is described in this paper for tracking the real time position of workers on construction sites. Experiments and tests were carried out both on- and off-site to verify the accuracy of static and dynamic targets by the system, indicating an average error of within 1m. Due to the limitations of the construction site involved, the developed system was only verified in one wing of a public residential construction project in Hong Kong for a short duration.

The second category of localization technology is image based localization, which utilizes image matching and computer vision techniques for determining user's location. Image matching detects distinguish visual patterns and characteristics in an indoor environment with images in a database (Ferdaus, Vardy, Mann, & Gosine, 2008). However, image matching approach has low accuracy such as room level accuracy, and it suffers from occlusions and changes in environments. Computer vision-based localization methods utilize sensors, such as laser scanners and video cameras for data acquisition and high processing power to process the data, and are mostly suitable for robot navigation (Taneja et al., 2012).

The third category of localization is Motion based technologies, which utilizes acceleration and heading measurements to determine an object's location relative to its last know location. This approach utilizes sensors, such as an accelerometer to measure acceleration in three dimensional spaces, a gyroscope to calculate the heading and a dead reckoning algorithm to fuse acceleration, and heading direction (Ibrahim & Moselhi, 2015b; Randell, Djiallis, & Muller, 2003). Unlike radio frequency localization, motion sensing technology is independent of any infrastructure (Gelb, 1974). However, the motion sensing does not provide high location accuracies, but the accuracy can be improved by smart algorithms, which able to correct drift errors (Glanzer, Bernoulli, Wiessflecker, & Walder, 2009).

A summary of localization technologies, advantages and disadvantages of the above mentioned technologies is presented in Table 2-2.

Table 2-2: Summary of Localization Technologies

Category	Technology	Advantages	Disadvantages	Accuracy
Wave Based	Infrared	Low power requirements Low circuitry costs Higher security Portable High noise immunity	Only works with line of sight Blocked by common materials Short range Light and weather sensitive Low data transmission rate	Down to a few centimeters
	WLAN	Usage of readily deployed infrastructure, reduced cost	Coarse localization Requires an offline phase Sensitive to interference, signal propagation effects, and dynamic environmental changes	Down to a few meters
	RFID	Low tag cost Active tags are more	Limited localization accuracy Limited range with passive	Down to a few meters

		expensive and require a battery	tags High reader cost	
	UWB	High data transmission rate Capability for expansion in the number of devices used simultaneously	Dedicated transmitter-receiver infrastructure Require time synchronization between nodes	Down to a few centimeters
	Ultrasound	Extremely high localization accuracies	Require external synchronization Affected by ambient noise. Accuracy affected by propagation issues and NLOS. Speed of sound variations, dependent on temperature and other environmental conditions	Down to a few centimeters
Image based	Computer vision	Hardware needed is becoming cheaper with off-the shelve components	Coarse accuracy Susceptible to occlusions and changes in indoor environments. Very computationally expensive and memory consuming. Cumulative error built-up	Down to a few meters
	Image matching	Data sources are already available in security System cameras	Complete prior knowledge of the fixed geometrical structure	Down to a few meters
Motion based	Inertial Navigation	Self-containment Resilience to environmental conditions. Continuous update of location estimates	Drift inherent to sensors. Relative localization Require initialization and calibration	Dependent on recalibration

Three main location estimation techniques had been utilized in literature to locate an object using radio frequency (RF): Trilateration, Scene Analysis and Proximity.

Trilateration is based on geometric properties, where the object's position is determined by measuring its distance from several reference points (Gonçalo & Helena, 2009). Trilateration method determines the position of a tag or reader using distance estimated at three reference points. Consider a tag positioning problem in a 2-D space as an example. Figure 2-4 shows how the tag position can be estimated using the trilateration method, where the range of the unknown tag to reference points (reader antennas) P$_1$(x$_1$, y$_1$), P$_2$(x$_2$, y$_2$), and P$_3$(x$_3$, y$_3$) are estimated as d$_1$, d$_2$ and d$_3$ respectively.

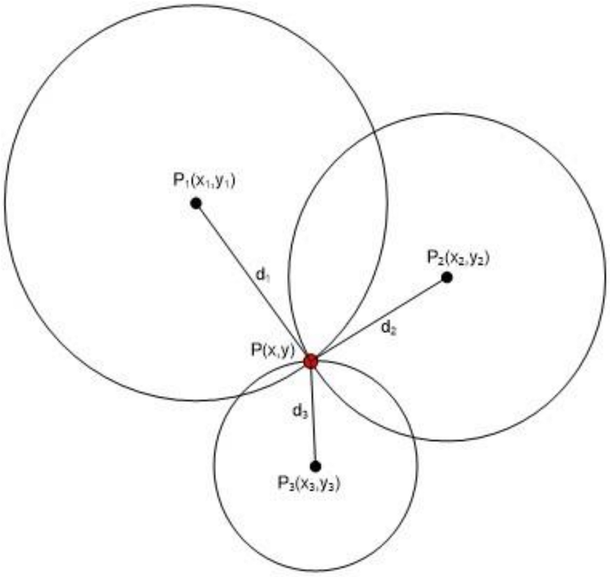


Figure 2-4: Tag Positioning using Trilateration

The location of the unknown tag, denoted as (x, y), can be determined by solving the following three equations

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2, \qquad i = 1,2,3. \tag{2-6}$$

As a result, the coordinate of the unknown tag is obtained as

$$\begin{cases} x = -\frac{1}{2}\frac{(d_1^2-y_1^2-x_1^2)(y_2-y_3)+(d_2^2-y_2^2-x_2^2)(y_3-y_1)+(d_3^2-y_3^2-x_3^2)(y_1-y_2)}{x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2)} \\ \\ y = -\frac{1}{2}\frac{(d_1^2-y_1^2-x_1^2)(x_2-x_3)+(d_2^2-y_2^2-x_2^2)(x_3-x_1)+(d_3^2-y_3^2-x_3^2)(x_1-x_2)}{y_1(x_2-x_3)+y_2(x_3-x_1)+y_3(x_1-x_2)} \end{cases} \tag{2-7}$$

As for scene analysis technique, which estimates the object's location using a pre-defined data set of observations about the surrounding scene. Such method requires offline training phase and data storage to maintain pre-defined observation data, which is not practical for dynamic environments such as construction jobsites (Fu & Retscher, 2009; Woo et al., 2011). In this technique, the location of the object is computed using features of a “scene” constituted of the electromagnetic signal characteristics map defined by the attenuation of a transmitted signal from multiple locations in the “field of view” for the scene. Therefore, there is an “RF signature” unique to a given location and combination of receivers (Bulusu, Heidemann, & Estrin, 2000; Hightower & Borriello, 2001). The major disadvantage of this technique is the extensive effort required to generate the signal signature database and reconstruct an entirely new database due to significant changes in the environment which typically occur on a large industrial construction project. Thus, this approach requires a fixed reader grid, a static signal transmission degradation map, and much recalibration when the transmission space changes.

For the proximity method, object’s distance to reference points is not actually measured, but only its presence within a certain range is determined (Aryan et al., 2011). Therefore, proximity algorithm is simple to apply, but has a coarse localization accuracy (N. Li & Becerik-Gerber, 2011). The proximity technique determines whether an object is near one or more known locations, by monitoring physical phenomena with limited range, such as physical contact and communication connectivity to the scanner or access points in a wireless cellular network. The method of constraints, accumulation arrays, Dempster-Shafer theory, and fuzzy logic are some of the approaches that can be used individually or in combination of proximity based localization models (Caron et al., 2007). A crude variation on this approach is the center of gravity (COG) analysis, where the COG of the RSSI readings of a tag is used to estimate its location. In LANDMARC, the concept of reference tags is introduced, which can provide reference locations. The known locations of the nearest neighboring reference tags and nearness to the tracking tag are used in computing the tracking tag’s location. The algorithm has the following advantages. First, the algorithm uses tags instead of more readers, which greatly reduces the cost of the system. Second, the reference tags and target tags are in the same environment, and the effect of environmental factors can be effectively off-settled. However, the algorithm has the following drawbacks. First, it does not work well in a closed area with severe radio signal multipath effects. Second, to further improve the localization accuracy, more reference tags are needed, which is costly and may cause RF interference phenomenon. The virtual reference tags were used to achieve a higher accuracy. These tags are virtually distributed linearly between real reference tags, which increase the density of the reference tag grids. The locations of the virtual reference tags are known and recorded, and their signal strength is estimated by linear interpolation of that of the real tags next to them. The sensing area is divided into small regions, and each reader maintains its own proximity map. As a result, the most probable location can be estimated.

2.3.4 Mobile Computing

Previous studies of mobile computing for construction demonstrated that mobile computing technologies have great potential to significantly improve various construction activities, including material tracking, safety management, defect management, and progress monitoring. The potential improvement is largely attributed to the enhanced mobility of computing devices, which allows users in any location to access and share important construction project information in an efficient manner. The recent advent of smart phones strengthens the trend of high mobility. The advantages of using mobile computing devices in the construction industry have been well described (Baldwin et al., 1994; Fayek et al., 1998; Kimoto et al., 2005; McCullouch, 1997; Saidi et al., 2002). Mobile computing devices have been used in the construction industry for a number of applications such as: (1) to develop a field inspection support system for civil systems inspections (Sunkpho & Garrett, 2003); (2) to develop a pen-based computer field application of an automated bridge inspection system (Elzarka & Bell, 1997); (3) to provide collaborative and information sharing platforms (Kim et al., 2013; Penã-Mora & Dwivedi, 2002); (4) to use mobile computers to capture data for piling works (Ward et al., 2003), and (5) to use PDAs in construction supply chain management systems (Tserng et al., 2005). Montaser & Moselhi (2012a) presented an automated methodology utilizing BIM 4D modeling and tablet PC for progress reporting in construction jobsites as shown Figure 2-5. The tablet PC was used to collect the as built progress data using RFID data, barcode data, images, notes, sounds and video clips. The collected data is then used to update project status on the 4D model, which is subsequently used for comparison with the as planned conditions.

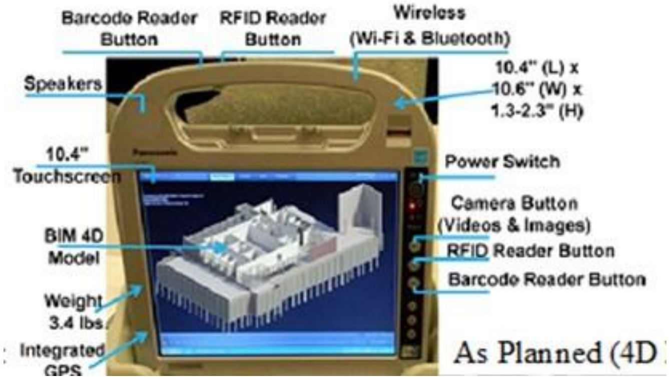


Figure 2-5: Tablet PC and BIM (Montaser & Moselhi, 2012a)

2.3.5 3D Imaging and Photogrammetry

3D imaging is used for a range of applications such as the creation of accurate as-built models, and the rapid surveying of highways and mines. Laser Detection and Ranging (LADAR) is a 3D laser scanner that is mainly used for spatial measurement. Other applications include surveying, earthmoving operations, monitoring the progress of concrete casting, highway alignment, paving operations and construction quality control (Lytle, 2011). Bosche et al. (2008) developed a method of inferring the presence of model objects in range images. Their approach focused on the comparison of a 3D image of a construction scene with a simulated scan of a 4D building model using similar scan parameters. The approach was successfully demonstrated in a steel construction project, as shown in Figure 2-6. Turkan et al. (2013) presented a system integrating 3D object recognition technology and schedule information into a combined 4D object-based construction progress tracking system. During the construction of a reinforced concrete structure, they performed and extensive field study to investigate the performance of the system.

Photogrammetry is the extraction of the geometrical properties of an object from photographic images (Styliadis, 2007). The value of photo images is that they can obtain information about texture and color, which is an advantage to Photogrammetry over laser scanning (Zhu, German, & Brilakis, 2010). Fard & Peña-Mora (2007) developed a methodology for construction progress monitoring that leverages the large number of photographs that are already taken on construction sites for production documentation. By analyzing imagery taken daily, a time-based visualization can be generated which compares the 4D as-built data with the 4D as-planned data within a common user interface.

Photogrammetry and 3D scanning was integrated to track changes for work accomplished. Integrating 3D imaging and Photogrammetry mitigates the limitations associated with each of them individually, such as the number of scans required and the time needed for each scan to produce satisfactory results during the 3D modeling process (El-Omari, 2008; Moselhi & El-Omari, 2007). Khosrowpoura et al. (2014) presented a new method for activity analysis of construction workers using inexpensive Microsoft Kinect RGB-D sensors. The developed method has an average accuracy of 76% and a maximum accuracy of 92% in activity analysis for interior operations. However, the sensor is affected by direct sunlight, and the sensor coverage is limited to 5 m, and hence multiple sensors are needed to cover several work areas but interferences need to be managed. Furthermore, occlusions and tool-interactions affect the accuracy of detecting body postures.

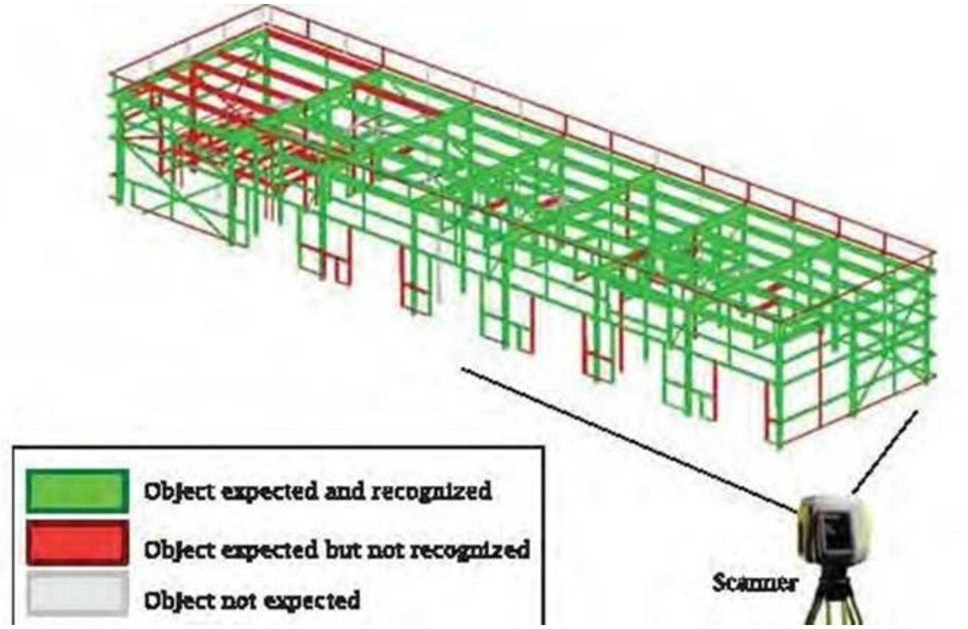

🟩 Object expected and recognized
🟥 Object expected but not recognized
⬜ Object not expected
Scanner

Figure 2-6: 3D Scan and 3D Model Object Recognition (Bosche et al., 2008)

2.3.6 Visualization and Building Information Modeling (BIM)

The National BIM Standard defines a Building Information Model (BIM) as a digital representation of the physical and functional characteristics of a facility (Eastman et al., 2008). BIM is a shared knowledge resource for information about a facility that forms a reliable basis for decisions during its life cycle, which is defined as existing from the earliest conception to demolition. A basic premise of a BIM is a collaboration among different stakeholders at different phases in the life cycle of a facility, which involves the insertion, extraction, updating or modification of information in the BIM to support and reflect the roles of that stakeholder. One of the advantages of a BIM over a 3D AutoCAD format is that the objects in the BIM are parametric, are linked to each other, and contain a variety of attributes. This long-term advantage of the BIM may justify the permanent attachment of sensors, such as RFID tags, to a number of key components (Motamedi & Hammad, 2009). The permanent RFID tags can be used for material tracking at the manufacturing and delivery stages (Razavi et al., 2008), for progress tracking during the construction stage, and for maintenance during the entire life cycle of the component.

4D BIM is a visual representation that combines an object oriented 3D BIM model with time. 4D BIM is information visualization that is easier to understand than traditional methods. 4D BIM models are a form of visual representation of a project that also takes into consideration the temporal aspect of how project teams plan to actually build a project, according to construction schedules (Hartmann et al., 2008). 4D BIM could be used strategically by on-site management for progress visualization and presentation, locating equipment such as material hoists, checking access/openings for equipment, storage visualization and the utilization and estimation of quantities. Moreover, 4D BIM can assist site personnel in brainstorming sessions and discussions about access, storage and sequencing of works (Chau et al., 2005). Better visualization facilitates team collaboration in removing logical errors in construction operations. Owners of the constructed facilities may have little experience in construction projects, and are often unable to truly participate in the construction plan development process unless a simple method of visualization and communication is made available to them (Kang et al., 2007). Montaser & Moselhi (2012a) presented an automated methodology utilizing BIM 4D modeling and tablet PC for progress reporting in construction jobsites as shown in Figure 2-7. The method integrates the project schedule and BIM, a 4D model is generated to simulate a planned construction sequence. A tablet PC is used to collect the as built progress data using RFID data, barcode data, images, notes, sounds and video clips. The collected data is then used to update project status on the 4D model, which is subsequently used for comparison with the as planned conditions. An example project was considered to apply to the proposed methodology on a construction jobsite of research laboratory building in the west end of Montreal.

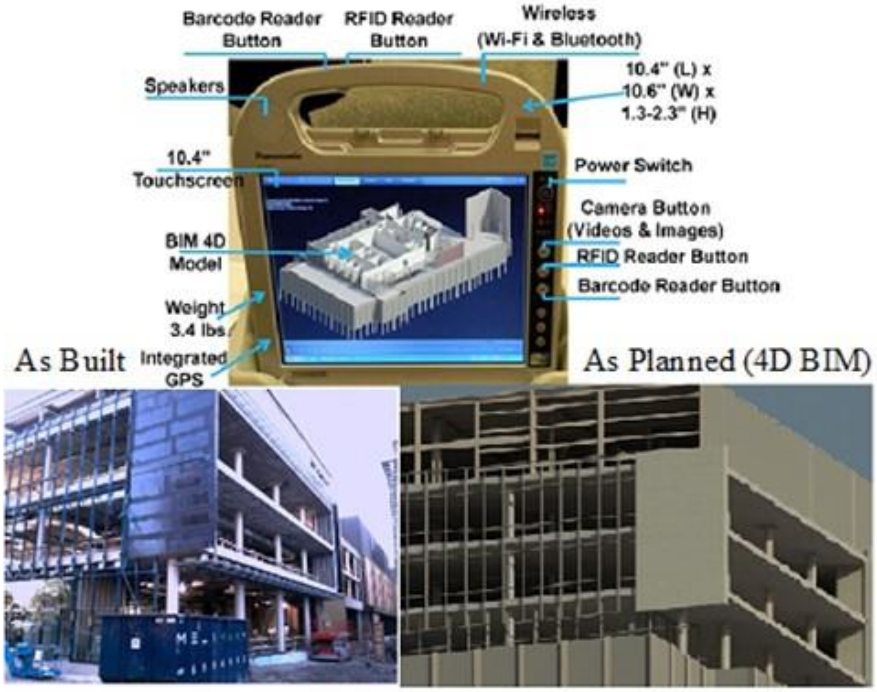


Figure 2-7: Four Dimensional Model (Montaser & Moselhi, 2012a)

2.4 Emerging Technologies

Recent advancement in computing and information technologies has presented tremendous opportunities for better automation in the construction industry. These technologies present advanced methodologies in data acquisition, capable of gathering on-site data wirelessly in near real-time. Wireless sensor networks (WSN) and Internet of Things (IoT) have many potential applications in construction project management, such as building automation, project tracking and control, job site safety, and civil infrastructure monitoring. WSN is equipped with a number of sensors and communication devices, which provide the capability to automate and integrate multiple data sources. WSN (hardware and software) configuration is highly dependable on their potential application objectives and required performance.

The idea of the internet of things (IoT) was developed in parallel to WSNs. The term internet of things was devised by Kevin Ashton in 1999 and refers to uniquely identify objects and their virtual representations in an “internet-like” structure. The Internet of Things (IoT) refers to machine-to-machine (M2M) technology enabled by network connectivity and cloud infrastructure, to reliably transform data into useful information. The value of IoT isn’t only gathering data, but also its ability to make better decisions. The potential for IoT applications in a wide range of industries has grown massively in the last couple of years due to the declining cost of sensors, connectivity, and data processing power, which is making the return on investment for IoT projects become more appealing.

2.4.1 Wireless Sensor Networks (WSN)

A WSN can generally be described as a network of nodes that cooperatively sense and may control the environment, enabling interaction between persons or computers and the surrounding environment. The development of WSNs was inspired by military applications, notably surveillance in conflict zones. Today, they consist of distributed independent devices that use sensors to monitor the physical conditions with their applications extended to industrial infrastructure, automation, health, traffic, and many consumer areas.

Research on WSNs dates back to the early 1980s, when the United States Defense Advanced Research Projects Agency (DARPA) carried out the distributed sensor networks (DSNs) program for the US military. Even though early researchers on sensor networks had the vision of a DSN in mind, the technology was not quite ready. More specifically, the sensors were rather large, and the number of potential applications was thus limited. Furthermore, the earliest DSNs were not tightly associated with wireless connectivity.

Recent advances in computing, communication and micro-electromechanical technology has resulted in a significant shift in WSN research and brought it closer to the original vision. The new wave of research on WSNs started around 1998 and has been attracting more and more attention and international involvement. The new wave of sensor network research puts its focus on networking technology and networked information processing suitable for highly dynamic ad hoc environments and resource-constrained sensor nodes. Furthermore, the sensor nodes have been much smaller in size and much cheaper in price, and thus many new civilian applications of sensor networks such as environmental monitoring, vehicular sensor network and body sensor networks have emerged.

WSNs nowadays usually include sensor nodes, actuator nodes, gateways and clients. A large number of sensor nodes deployed randomly inside of or near the monitoring area (sensor field), form networks through self-organization. Sensor nodes monitor the collected data to transmit along to other sensor nodes by hopping. During the process of transmission, monitored data may be handled by multiple nodes to get to the gateway node after multi-hop routing, and finally reach the management node through the internet or satellite as shown in Figure 2-8.

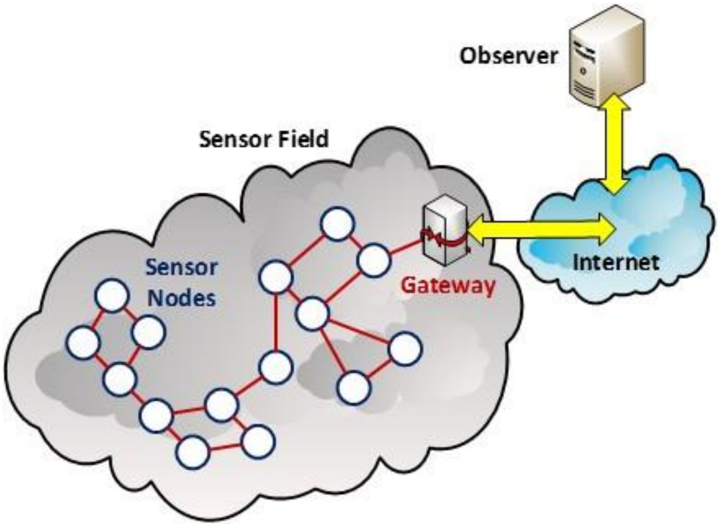


Figure 2-8: Wireless Sensor Networks

The sensor node is one of the main parts of a WSN. The hardware of a sensor node generally includes four parts: the power and power management module, a sensor, a microcontroller, and a wireless transceiver. The power module offers the reliable power needed for the system. The sensor is the bond of a WSN node which can obtain the environmental and equipment status. A sensor is in charge of collecting and transforming the signals, such as acceleration, vibration and position, into electrical signals and then transferring them to the microcontroller. The microcontroller receives the data from the sensor and processes the data accordingly. The Wireless Transceiver (RF module) then transfers the data, so that the physical realization of communication can be achieved.

Generally, a WSN consists of a number of sensor network nodes and a gateway for the connection to the internet as shown in Figure 2-8. The general deployment process of a WSN is as follows: firstly, the sensor network nodes broadcast their status to the surroundings and receive status from other nodes to detect each other. Secondly, the sensor network nodes are organized into a connected network, according to a certain topology (linear, star, tree, mesh, etc.). In order to expand the coverage of a network, the sensor network uses multi-hop transmission mode. That is to say the sensor network nodes are both transmitter and receiver. The first sensor network node, the source node, sends data to a nearby node for data transmission to the gateway. The nearby node forwards the data to one of its nearby nodes that are on the path towards the gateway. The forwarding is repeated until the data arrives at the gateway, the destination.

Data aggregation is the process of integrating multiple copies of information into one copy, which is effective and able to meet user needs in middle sensor nodes. The introduction of data aggregation benefits both saving energy and obtaining accurate information. The energy consumed in transmitting data is much greater than that in processing data in sensor networks. Therefore, with the node's local computing and storage capacity, data aggregating operations are made to remove large quantities of redundant information, so as to minimize the amount of transmission and save energy. In the complex network environment, it is difficult to ensure the accuracy of the information obtained only by collecting few samples of data from the distributed sensor nodes. As a result, monitoring the data of the same object requires the collaborative work of multiple sensors which effectively improves the accuracy and the reliability of the information obtained. The performance of the data aggregation protocol is closely related to the network topology.

A number of research efforts have investigated the use of WSNs for building management (Grindvoll et al., 2012; Huang et al., 2008; Jang et al., 2008; Kintner-Meyer, 2005; Osterlind et al., 2007). The application of WSNs has been extended to building automation (Feng et al., 2008; Huang et al., 2011; Huang et al., 2008; Malatras et al., 2008).

Also, some research efforts have investigated the use of WSNs for infrastructure monitoring. Sadeghioon et al. (2014) developed a methodology for leak detection in water pipelines using wireless smart sensor networks. Their developed method is able to monitor the condition, in particular the pressure and hence leaks, of buried water pipelines. This method allows easy installation of the sensor nodes on the pipes without jeopardizing the pipes' structural integrity. Their advantage over other commonly used leak detection methods is that they have a degree of redundancy as individual faulty nodes do not render the whole system obsolete and allow for continuous monitoring without operator intervention.

Jang & Skibniewski (2007) implemented a tracking architecture using wireless sensor modules by combining radio frequency signals and Ultrasound; the results showed accurate position estimations with enhanced network flexibility. However, traditional ultrasound positioning has some disadvantages, including line-of-sight transmission, multipath, high cost and power consumption, which may hinder the possible applications in complicated construction environments (Shen et al., 2008). Various combinations of RFID and Zigbee-based sensor networks have also been applied for materials tracking and supply chain management (Cho et al., 2011; Shin et al., 2007). These studies confirmed that WSN can improve the wireless communication and network flexibility, but their primary use was only data transmission, and not positioning. Shen & Lu (2012) investigated the application of ZigBee-based WSN for indoor positioning and tracking of construction resources. Real-time positions of the mobile nodes were determined by applying the RSSI method and the trilateration algorithm. However the RSSI values were susceptible to application environments, which presented difficulties and challenges to implementation of such positioning methodologies in complicated and dynamic settings in construction.

2.4.2 Data Fusion Models in Construction

Data fusion refers to the combining of information from multiple sources in order to improve the quality of information obtained separately from each source. It is utilized to make inference decisions about the state of a construction project based on data from different sources. For informed decisions and objective assessments of progress at a construction site, data from a number of sources must be combined because not all of the necessary information can be captured using a single data source.

In recent years, a number of studies have considered multi-sensor data fusion models in order to capture a more complete picture of the progress of a project by using information acquired from GPS, RFID, and other sources of information for tracking and locating construction materials (Cheng & Chen, 2002; Ergen & Akinci, 2007; Moon & Yang, 2009; Razavi, 2010; Song et al., 2006). With the development of all of these technologies, a large amount of data can be collected from construction sites both semi-automatic and semi continuous. However, challenges related to the processing and reduction of data to produce meaningful conclusions and the fusing of data from a variety of sources remain obstacles to the achievement of a practical and comprehensive automated progress-tracking solution for construction sites.

Studies of data fusion for the particular application of automated progress tracking in construction projects have focused primarily on automated object-recognition models (Cheng et al., 2012; Golparvar-Fard et al., 2013; Y. Turkan et al., 2013) and on automated object-tracking models (Khaleghi et al., 2013; Shahandashti et al., 2011). These object-based models have shown promising results for projects in which progress is tracked in terms of a bulk quantity of materials or objects, such as steel-framed building construction, where the progress of the building project is reported in tons of steel installed; in such a context, recognition of the number of objects that have been installed provides an adequate level of detail for that type of progress tracking. However, many activities on construction projects, entail specific elements (welding, inspection, etc.) that are not associated directly with the movement or addition of a physical entity at the site and therefore cannot be tracked effectively using object based models.

Shahi et al. (2014) and Shahi (2012) presented an activity-based data fusion model, which incorporated an Ultra Wide Band (UWB) positioning system to track activities in a construction project. A field experimentation, study on an industrial-type building construction project was conducted to validate the model presented in this research. The scope of the experimental program was limited to ductwork, HVAC, and piping activities on the project. It was noted that the number of changes occurring during construction may be significantly higher for piping and industrial projects in comparison to steel or concrete building construction. The significance of the design change variability is that although automated object recognition and material or asset tracking algorithms that use the 3D CAD or BIM model as a-priori information may be accurate for concrete or steel structures, they may be ineffective in tracking the progress of piping and many other mechanical and electrical services found throughout most projects.

2.5 Summary and Identified Limitations

The literature was reviewed in areas pertinent to automated data acquisition technologies, indoor localization and data visualization. The literature review was conducted with prime focus on the impact of those areas on the development of efficient automated site data acquisition models. The following gaps and limitations are identified accordingly:

- While a significant number of researchers have investigated the field of automated site data acquisition, in all cases the focus has been on using off-the-shelf technologies. Off-the-shelf technologies are produced to meet the perceived needs of a particular market or application. Where their generic features are designed as a one size fits all, which may not form a perfect fit for complex applications such as construction operations.

- Although WSN technologies promise a great potential for applications in construction, little research has been pursued to develop customized and flexible automated site data acquisition models for tracking and progress reporting on construction operations.
- Outdoor automated progress tracking and reporting methods which are limited to only one source of data such as GPS or RFID. Might not be able to capture a complete picture of the on-going construction operations.
- Standalone GPS and RFID based earthmoving tracking methods have a main underlying assumption that the hauling units are loaded to their full capacity, which is not a valid assumption in many cases, especially in a situation where a carry back is building up on the truck bed as shown in Figure 2-9, decreasing the carrying capacity of the truck and wasting a considerable amount of fuel.

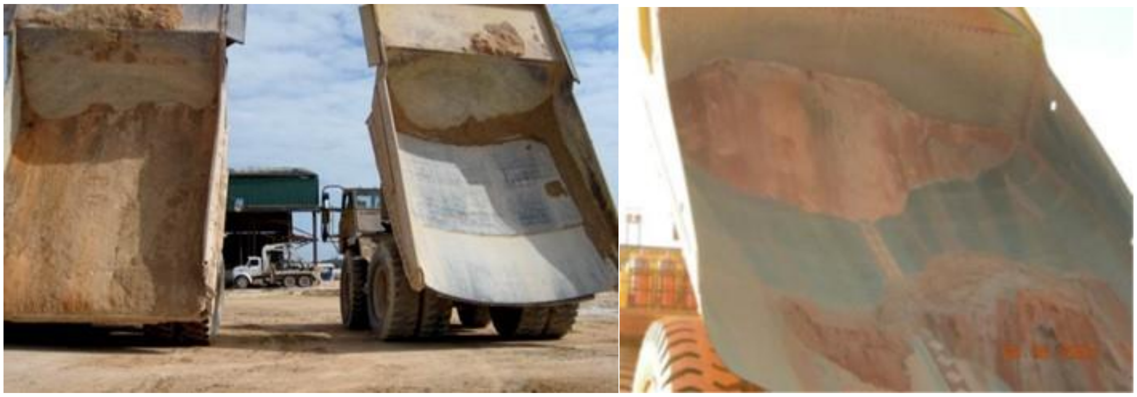


Figure 2-9: Truck Carry back Buildup

- Standalone GPS tracking methods accuracy might be impacted with the limited GPS accuracy in urban construction jobsites due to limited satellite reception as shown Figure 2-10, or even complete failure to report the progress in case of GPS hardware malfunction.

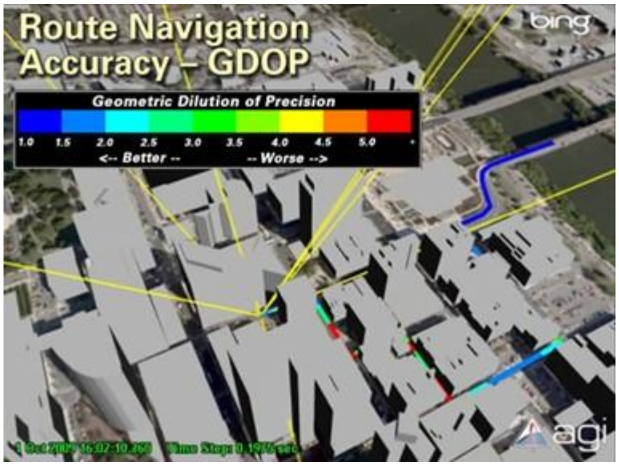


Figure 2-10: GPS Accuracy in Urban Locations (Courtesy of Agi.org)

- Standalone GPS data alone is not enough to distinguish between different modes of operations of construction equipment, for example, it might not be possible to distinguish between trucks queuing for loading or the loading activity itself in a situation similar to the setting shown in Figure 2-11.

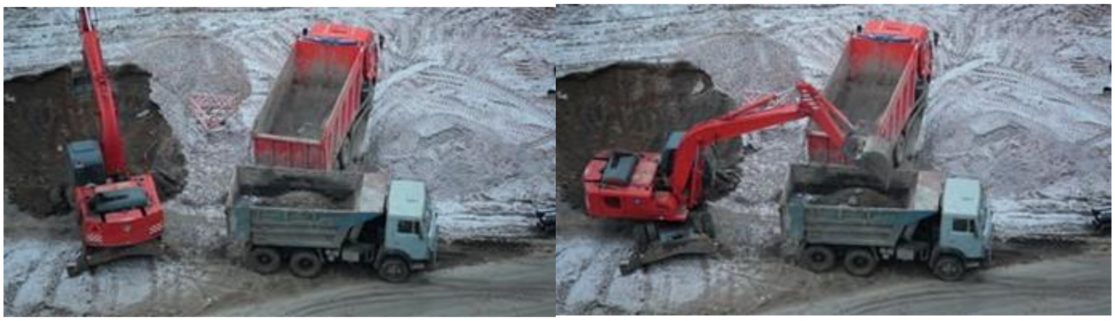


Figure 2-11: Truck Loading Queue

- Several heavy construction equipment such as excavators, stay stationary for long durations while executing their work. Location sensing alone cannot give a direct indication whether they are operational or ideal. Detection of equipment interaction and poses is essential to identify the various modes of operations of such equipment. Real-time measurement of durations spent by each equipment with respect to their mode of operation enables contractors to optimize their operation and fleet configuration in real-time to maximize their profit.

- Expensive technology such as On Board Instrumentation System (OBIS) and its black box format, prevents users from accessing its respective algorithms and modifying it as they see fit. Also, these systems store data in propriety data formats, and the stored data is often difficult to access without using the vendor specific software.
- In the literature, there is a lack of data fusion algorithms for near real-time productivity analysis.
- Automated progress tracking and reporting described in literature lacked the collection of contextual data such as weather conditions. This type of data is important for realistic productivity analysis and it enables better estimates on future jobs.
- Several researchers studied the utilization of various technologies for indoor localization, especially radio frequency and wireless protocols. While each technology has its own advantage and disadvantage, there is no experimental analysis studies to guide the selection of best performing wireless protocols indoor.
- Received signal strength (RSSI) is affected by the several interferences such as multi path and shadow fading interference, which causes fluctuations in the measured RSSI signals. In literature moving average filter is commonly used to remove noise from measured RSSI, however, it has several limitations such as time lag and the removal of important features of

the original signals. There are no experimental studies to compare the performance of different filtering techniques for RSSI, and their effect on the localization accuracy.

- The fundamental key for reliable and accurate indoor location estimation is path-loss models, which are used to convert measured received signal strength (RSSI) into the corresponding distances. Researchers commonly conduct laboratory experiments in an offline phase to construct static path-loss models. Using a static path loss model might be valid for a constant environment, where minimal changes in the physical layout of the building. However, in the presence of moving resources, metallic objects and structural barriers, static path-loss models produce poor distance estimates. In order to alleviate such impact, smart and adaptive path-loss models are required to cope with the fast-changing construction jobsites environment.
- Indoor location estimation in two dimensional spaces is not practical for multiple floor jobsites. There is a need for three dimensional indoor localization systems for accurate tracking of resources on high rise building projects.

Chapter 3 : PROPOSED FRAMEWORK

3.1 General

The aim of this chapter is to provide a comprehensive overview of the developed framework with a special focus on automated tracking and progress reporting of construction operations. The research vision and approach are presented; highlighting the main concepts of rapid prototyping, objective driven design and modular configuration. Figure 3-1 depicts the main sections of this chapter.

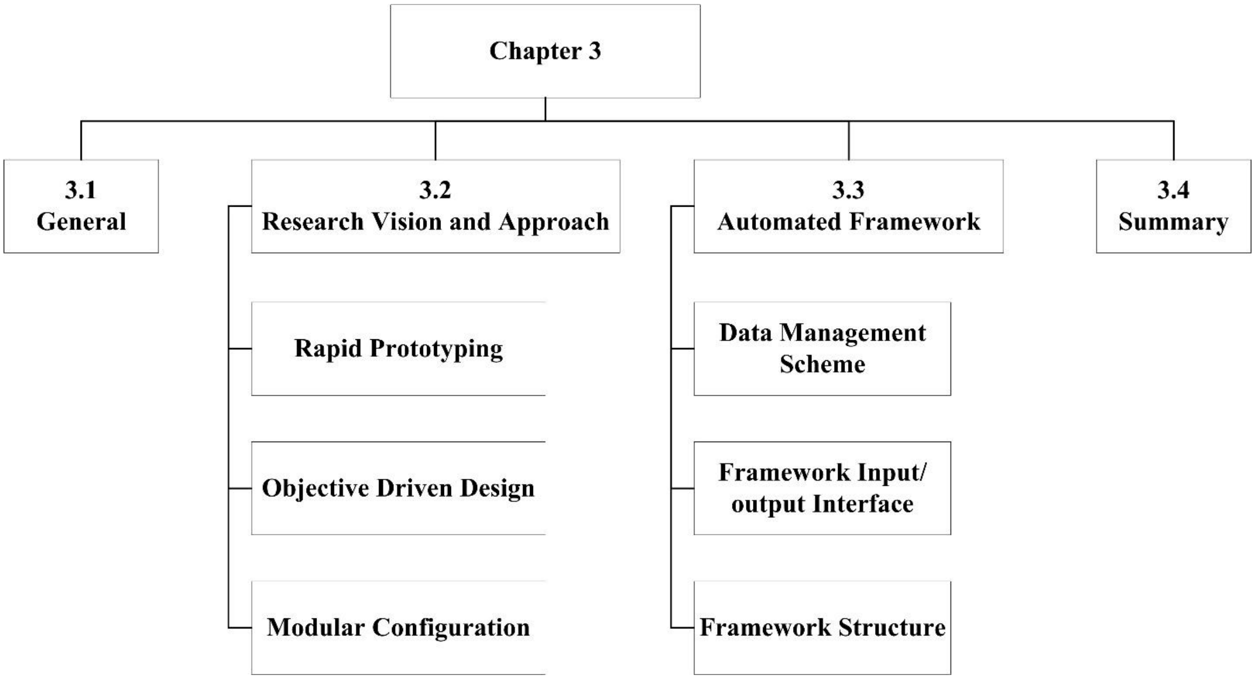


Figure 3-1: Chapter 3 Overview

3.2 Research Vision and Approach

While a number of researchers have conducted in-depth studies in the field of automated site data acquisition, in most cases, the focus has been on utilizing off-the-shelf systems. The vision for this research was formed based on the identified knowledge gaps and current practice limitations described in section 2.5. This vision can be summarized as follows:

- Fundamental experimental studies with various sensors and wireless protocols in laboratory and outdoor environments to test, validate and assist in the design of automated site data acquisition prototypes.
- Design of adaptable, flexible and cost effective automated site data acquisition prototypes integrating various sensor technologies to address limitations of off-the-shelf technologies.

- Study and experiment with various RSSI filtering techniques to assist in the selection of the best RSSI filter to improve indoor localization accuracy.
- Design and validate a self-calibrating and dynamic path-loss model to cope with continuous changes in the construction environment.

This research vision was guided by three main principles:

- Rapid prototyping to speed up the development and enable continuous improvement in the design.
- Objective driven design to ensure a structured mechanism for setting the design objectives and the performance measures of the prototype.
- Modular configuration to provide the flexibility for customized project configuration and enables re-usability of hardware components in order to cut the development cost.

3.2.1 Rapid Prototyping Approach

The use of the rapid prototyping technique in mobile wireless systems differs from the traditional technique used in mechanical engineering, in this context, rapid prototyping is mainly associated with the experimental implementations and development of hardware and software prototypes. In this research, rapid prototyping is utilized to allow development of prototypes in lab and/or simulation virtual environment. The hardware functions are tested and validated using simulation and lab experiments. In this way, an economy of time and material are obtained. It is fundamental that the design of the hardware prototype architecture to be flexible for necessary modifications required for system optimization. The choice of the Arduino UNO microcontroller board for developing prototypes in this research is based on two main factors. First, its low cost (about $25) combined with its great processing capacity, operating at 16 MHz clock frequency and executing up to 16 million instructions per second (MIPS). Which makes the prototype sufficiently efficient for implementations of complex software algorithms required for on-site data acquisition. An overview of the Arduino UNO characterizes and size is shown in Figure 3-2 and Table 3-1.

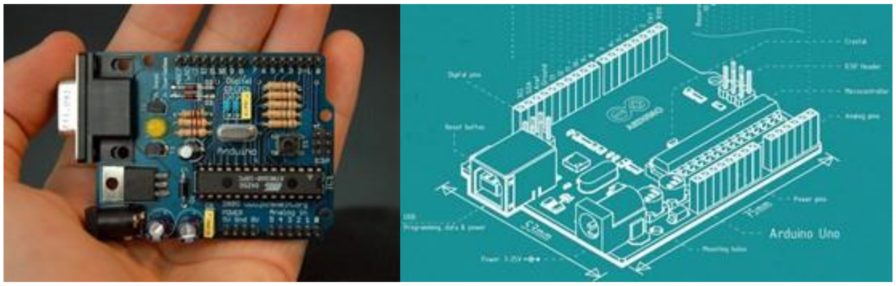


Figure 3-2: Arduino UNO Microcontroller Board

Table 3-1: Arduino UNO Microcontroller Characteristics

Microcontroller	**ATmega328**
Operating Voltage	5V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
Flash Memory	32 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Second, Virtual Breadboard Emulation Package for Arduino hardware enables virtual utilization and designing of Arduino UNO microcontroller in real time with Hardware-In-Loop (HIL) techniques. The HIL technique is commonly used by the aerospace industry for real time simulation and development of embedded mobile robotic controllers (Ledin, 2001). The Virtual Breadboard package is able to emulate program runs on the board, which listens to commands arriving via serial port, executes the commands, and, if needed, returns a result. Also, SIMULINK® and MATLAB® support running program algorithms on Arduino for control system and robotics applications and then simulate to verify that your algorithms work during simulation as shown in Figure 3-3.

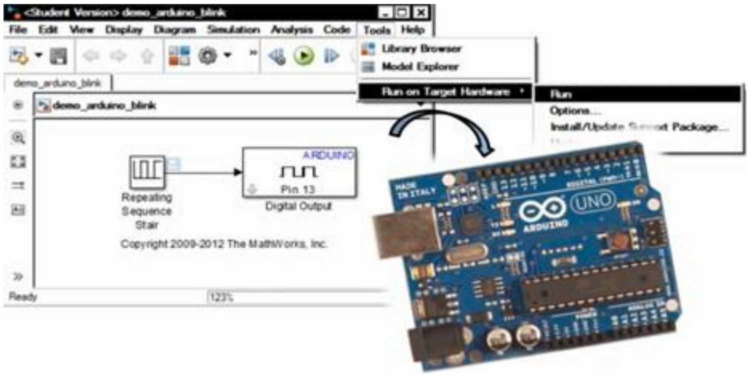


Figure 3-3: Simulink® Support for Arduino

3.2.2 Objective Driven Design

The initial step of developing a prototype is to identify the requirements and objectives governing the system operation. An objective driven development approach forms system goals based on initial functional directives, and then elaborates and refines these objectives until they have been broken down into functions that can be achieved by single modules, forming the requirements specifications. By ensuring that the requirements specifications achieve all high-level objectives, this approach provides a precise criterion for sufficient completeness of requirements specification (Lamsweerde, 2001).

The initial list of objectives is generally a high-level abstract description of the prototype target application. The developer then identifies sub-objectives, considering how each objective can be achieved. This results in a hierarchy of goals, which are then regrouped by similarity to produce system's functionalities. Sequence of steps is then formed to further refine the objectives, each step is either an action or an event. Every step lists data inputs and outputs, as well as data processing algorithms involved, and interfaces between the prototype and surrounding environment. When designing wireless sensor network prototypes for applications in automated site data acquisition, two major classes of design objectives must be considered; network architecture and application requirements (Ibrahim & Moselhi, 2014e).

The network architecture requirements contain the physical and logical organization of the network as well as the density of the sensor nodes. In general, the objective of sensor networks is to efficiently cover the deployment area. The logical and hierarchical organization of the network also impacts energy consumption and the selection of communication protocols. In addition, based on topology requirements, sensor networks can have a distributed organization or a clustered organization, where selected nodes can handle data forwarding. The network architecture requirements for construction applications can be determined by answering the following questions:

- What type of network topology best fits the application? (One-to-one, one-to-many, many-to-one or many-to-many?)
- How will the monitoring network work? (Master–slave, point-to-point, point-to-multipoint or peer-to-peer?)
- What are the worst case ambient conditions in the coverage area?
- Are there any known potential interference problems due to physical obstructions, RF interference?

The application requirements identify the information to be collected and processed by the sensor network. This data should be classified and quantified based on the type of the data and the required sensors for collecting such data. These classifications can be achieved by a comprehensive analysis of the targeted application. Based on the application requirements, configurations of individual sensor nodes can be identified which enables the selection of the sensors and communication protocols. The following questions can help to determine these configurations:

- What is the Quality of Service requirements of the application? (Does it require real-time monitoring or delay tolerant monitoring?)
- Does the system continuously poll for the information (periodic monitoring) or is it generated by exception (event-based monitoring)?
- What is the type of the data, i.e., location, proximity, tilt, height, etc.?

As a result of answering the above mentioned questions, the design objectives can be determined by the network topology, architecture and application requirements comprehensively. Full consideration of different sensor network options and how they will fit the targeted application is critical for a successful development.

3.2.3 Modular Configuration

Designing and configuring a WSN prototype to work seamlessly for various applications with little user intervention is not a simple task. In fact, it is difficult to design a platform that can satisfy a wide range of applications without sacrificing performance. The user has to reprogram the microcontroller to read a specific digital or analog port connected to a sensing element. Reprogramming the sensor platform's microcontroller to use a given sensor element could involve a very daunting sequence of tasks including learning the embedded operating system, MCU architecture, middleware, application support layer, etc. (Kouche, Hassanein, & Obaia, 2014).

The target users of the prototypes developed in this research are the construction project management community, and therefore we cannot assume that the users of such systems have a strong background in electrical engineering. Therefore, it is desirable to limit the input sensors to a predefined set of simple "plug and play" modules. The software interfaces should also share this characteristic of simplicity while maintaining flexibility. The user should not be required to learn a new programming language in order to use the system. The cost of the platform must also be sufficiently cost-effective to justify its use.

The vision of this research is to develop a modular hardware design to make the rapid prototyping, easier, as well as to allow for quicker redesign and the ability to reuse some of the hardware modules. Such approach enables redesign the prototype to adapt to different applications with relative ease. The main features of these modules are low cost, small size, and easy adaptation to different applications. The modular platform used in this research consists of three-layer as shown in Figure 3-4. This modular platform is divided into three functional layers: processing, communication, and sensing.

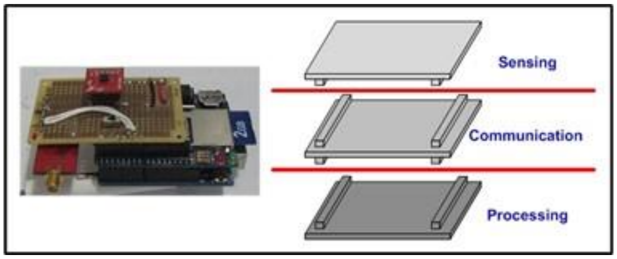


Figure 3-4: Modular Configuration

This modular design enables its users to simply plug a sensing module and the on-board software auto detect the hardware module and enables its software configuration and algorithms.

3.3 Automated Framework

The developed framework for automated project tracking and progress reporting encompasses two types of data acquisition prototypes as shown in Figure 3-5. The first prototype is Sensor Aided GPS (SA-GPS), which is designed for tracking outdoor activities such as earthmoving operations. This prototype consists of a microcontroller equipped with GPS module as well as a number of sensors such as accelerometer, barometric pressure sensor, Bluetooth proximity and strain gauges. This configuration is able to overcome standalone GPS limitations through data fusion of sensor data with the GPS data, which in turn enhances progress assessment and productivity analysis. The detailed description of the SA-GPS prototype developments for this prototype is presented in chapter 4. The second prototype is Self-Calibrated Wireless Sensor Network (SC-WSN), which is designed for indoor localization and tracking of construction resources (labor, materials and equipment). This prototype is able to enhance the indoor localization accuracy by consistently adapting its parameters to cope with the changing construction environment. The detailed description of the SC-WSN prototype developments is presented in chapter 5. Contextual data are collected with respect to weather conditions using a cluster of sensors integrated into a weather station. The details of the weather station components are presented in Appendix A.

The data acquisition prototypes collect the on-site progress data and send it to the progress measurement and productivity analysis algorithms. The captured data is utilized to estimate the actual progress, which is then compared to the as-planned baseline progress, using earned value analysis (EVA) to measure the project’s performance. The productivity analysis is responsible for analyzing the measured productivity and the contextual data to identify any bottlenecks.

Then, the progress reports are generated and the project cost and duration at completion are forecasted.

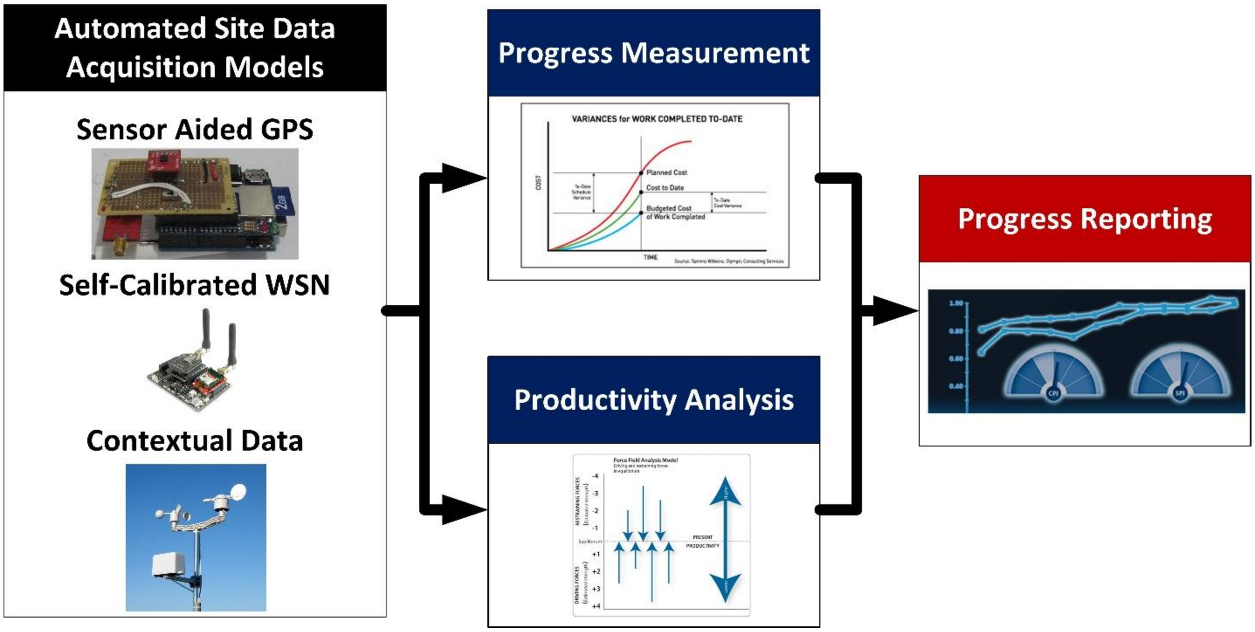

Automated Site Data Acquisition Models
Sensor Aided GPS
Self-Calibrated WSN
Contextual Data
Progress Measurement
VARIANCES for WORK COMPLETED TO-DATE
COST
To-Date Schedule Variance
Planned Cost
Cost to Date
Budgeted Cost of Work Completed
To-Date Cost Variance
TIME
Source: Tammo Wilkens, Olympic Consulting Services
Productivity Analysis
Force Field Analysis Model
Driving and restraining forces in equilibrium
RESTRAINING FORCES (Estimated strength)
-4
-3
-2
-1
Equilibrium
+1
+2
+3
+4
DRIVING FORCES (Estimated strength)
PRESENT PRODUCTIVITY
Higher
Lower
Progress Reporting
1.00
0.80
0.60
0.40
CPI
SPI

Figure 3-5: Proposed Framework Conceptual Design

3.3.1 Data Management Scheme

In automated site data acquisition methods such as those cited in literature, data collection schemes commonly use sensors or readers to relay raw data to a mobile computing unit. Such raw data have little value in themselves and need to be processed to extract meaningful information. The data collection scheme adopted in these methods suffers from a high volume of data traffic toward the sink node, which creates a bottleneck and results in long processing times (Figure 3-6).

We propose a scheme that supports localized cooperation of sensor nodes to perform complicated tasks, and in-network data processing to transform raw data into high level useful and actionable information. Toward this direction, data aggregation and processing, rendering its value for near real-time application. In the developed framework, sensory data are collected, aggregated, and interpreted at the sensor node level. This decreases post-processing time and user intervention required for data analysis. The proposed scheme is composed of a gateway node and multiple sensor nodes. The gateway node, which acts as an interface to the system, has sufficient computing power, essential energy and memory. The sensor nodes, on the other hand, are resource constrained devices, running on batteries and performing actual data acquisition. The sensor nodes are organized into a tree that routes data directly to the gateway node, as shown in Figure 3-6. Such a tree configuration facilitates data aggregation and reduces data routing.

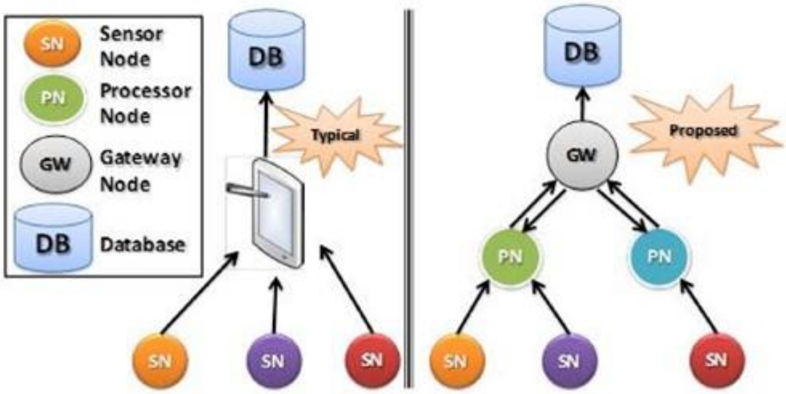


Figure 3-6: Data Management Schemes

3.3.2 Framework Input/output Interface

The developed framework had two main inputs: 3D Building information model, and resource loaded schedule as shown in Figure 3-7. In this research, Autodesk Navisworks with Timeliner is utilized to create the 4D model. The Autodesk process of creating the 4D model consists of three steps:

- Add the project 3D model to the Navisworks project file.
- Add the project schedule using the Navisworks Timeline module.

- Link the 3D objects to the schedule activities.

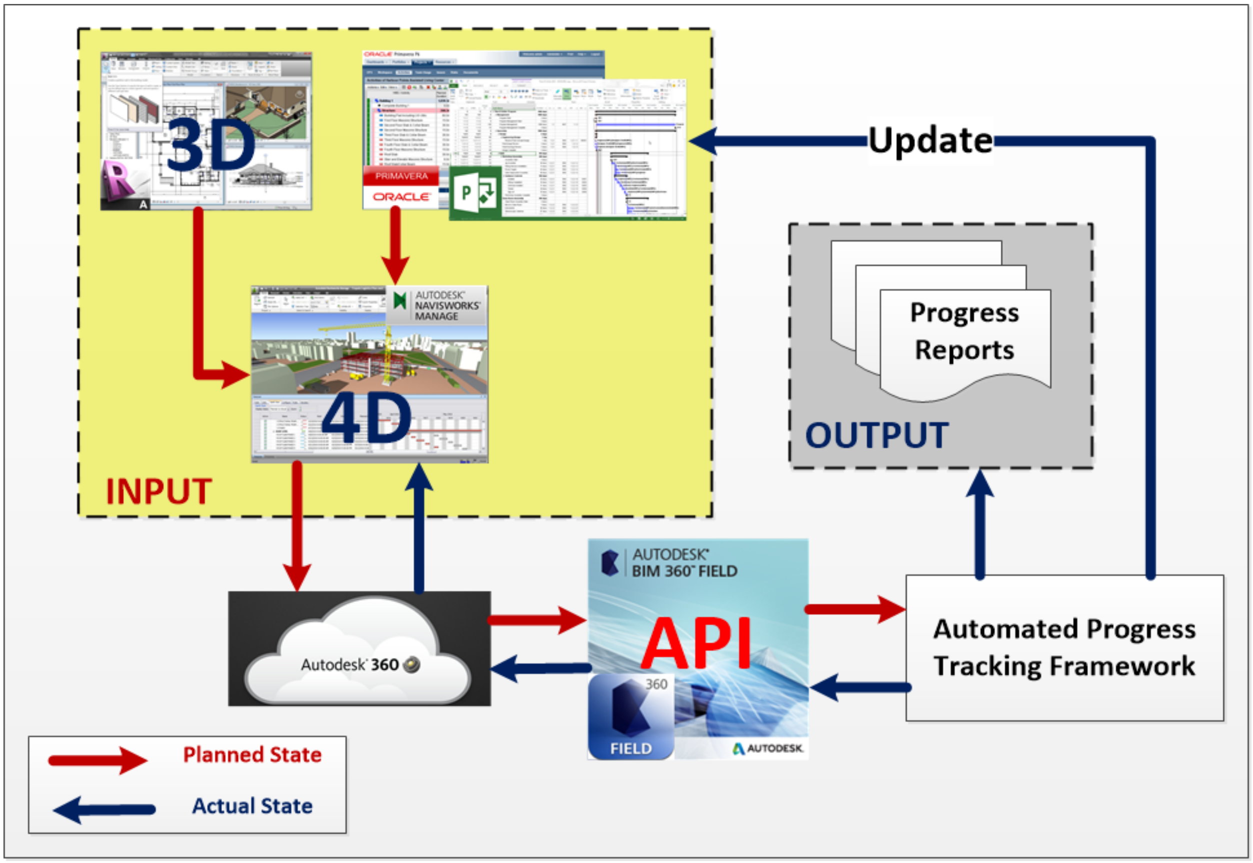


Figure 3-7: Developed Framework In/Out Interface

Autodesk BIM 360™ Field is a field management software for 2D and 3D environments that combines mobile technologies (iPad) at the construction site with cloud-based collaboration and reporting. BIM 360 Field™ enables remote access to project information by field personnel, helping to improve quality, safety, and commissioning for construction projects. Although BIM 360 Field™ offers some integrated reporting capabilities, they do not fully address the needs of automated progress reporting. It requires field personnel to manually enter information regarding the on-going construction operations. To overcome this limitation, this research utilizes BIM 360 Field API to provide inputs for the proposed automated tracking and progress reporting framework. In this context, BIM 360 Field API is used to exchange 4D model and construction site data through the Autodesk 360 cloud services.

3.3.3 Framework Structure

The developed tracking and progress reporting framework consists of six main modules as shown in Figure 3-8:

- Automated site data acquisition module, which is responsible for site data collection utilizing the two developed automated data acquisition prototypes. These prototypes are especially designed for tracking construction activities using low cost open source microcontrollers and a number of sensors (Ibrahim & Moselhi, 2013a). A detailed description of the developed prototypes is presented in chapters 4 and 5.
- Input data module, which is responsible for extracting the planned activities and their planned information such as: work zones, start dates, end dates, durations, planned quantities, man-hours, resources, and productivity. Work zones are created based on layout points created at the design and planning phases, where fixed points are marked on objects in 3D model using Autodesk Revit®. Then contractors at the execution phase can also create points on almost any object and export point locations to automated site data acquisition using Autodesk Points Layout® software. Project plan data along with quantities are extracted from the 4D model using the BIM 360 Field API and fed automatically to the database. A brief description of C# code utilized for connecting to the BIM 360 Field API and extracting the data is presented in Appendix B.
- Database module, which is responsible for storing processed data in a relational SQL® database. The upper level of the database represents a project entity connected to activity entity through one to many relationship. An activity entity represents the project scope of

work on time phase. Then a many to one relationship is used to link an object entity to an activity entity. The object entity includes geometrical properties of this object such as length, surface area and volume based on its activity units of measurement. Also an activity entity, is linked with a one to many relationship to resources entity, where each resource has a unique id that is used for tracking of this resource during the execution of the project. This id is the MAC address of the attached tag on the resource. The details of the database are presented in Appendix C.

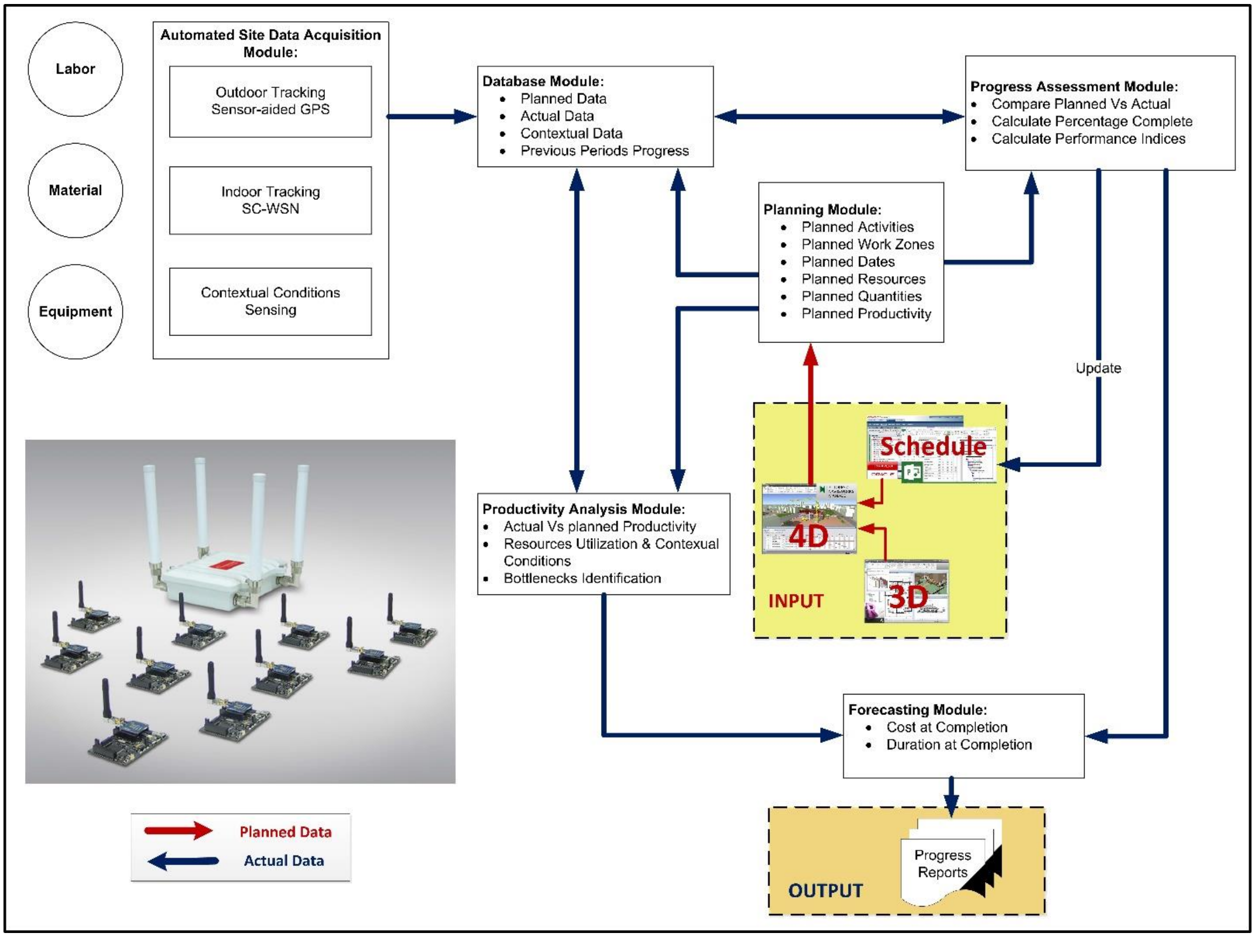

Labor
Material
Equipment
Automated Site Data Acquisition Module:
Outdoor Tracking Sensor-aided GPS
Indoor Tracking SC-WSN
Contextual Conditions Sensing
Database Module:
Planned Data
Actual Data
Contextual Data
Previous Periods Progress
Progress Assessment Module:
Compare Planned Vs Actual
Calculate Percentage Complete
Calculate Performance Indices
Planning Module:
Planned Activities
Planned Work Zones
Planned Dates
Planned Resources
Planned Quantities
Planned Productivity
Update
Schedule
4D
3D
INPUT
Productivity Analysis Module:
Actual Vs planned Productivity
Resources Utilization & Contexual Conditions
Bottlenecks Identification
Forecasting Module:
Cost at Completion
Duration at Completion
Progress Reports
OUTPUT
Planned Data
Actual Data

Figure 3-8: Proposed Automated Progress Tracking Framework

- Progress assessment module, which is responsible for comparing the actual progress gathered from the on-site data after applying the data fusion algorithms with the planned baseline progress. The module utilizes the earned value management (EVM) and progress templates for measuring the project performance. The module updates the baseline schedule with the actual progress to generate automatically as-built schedules (Ibrahim, Germain, Guevremont, Forcier, & Moselhi, 2013). The details of the developed as-built schedule generation algorithm is presented in Appendix D.

- Productivity analysis module, which is responsible for comparing actual and planned productivity. It also associates any deviation in productivity with the weather and operating condition existed on the site and captured using the on-site data acquisition. The productivity deviation indices can be used for identifying bottlenecks and provides better forecasting of project duration or for future estimates. The productivity analysis module was designed for analyzing only earthmoving operation in this research. The details of the developed productivity analysis algorithm is presented in chapter 4.
- Forecasting module, which is responsible for forecasting the at-completion project cost and duration using the newly developed self-adaptive forecasting technique (Ibrahim & Moselhi, 2013b). It takes into account any productivity deviation encountered on site and measured by the productivity analysis module. The module produces periodical progress reports according to the period specified by the user. The self-adaptive forecasting is applied to earthmoving productivity estimates in chapter 4 and the developed algorithm is described in details in Appendix E.

3.4 Summary

This chapter presented the research vision and an overview of the developed framework. The framework embraced integration and automation; utilizing wireless sensor network prototypes. The rapid prototyping approach is used for the development of the WSN prototypes. The developed framework input/output interface described in details showing the main inputs and outputs of the developed framework, along with its six modules.

Chapter 4 : SENSOR-AIDED GPS PROTOTYPE

4.1 General

The aim of this chapter is to describe the detailed development, testing and validation of sensor aided GPS prototypes for automated tracking and progress reporting of construction operations in outdoor environment. Figure 4-1 depicts the main sections of this chapter.

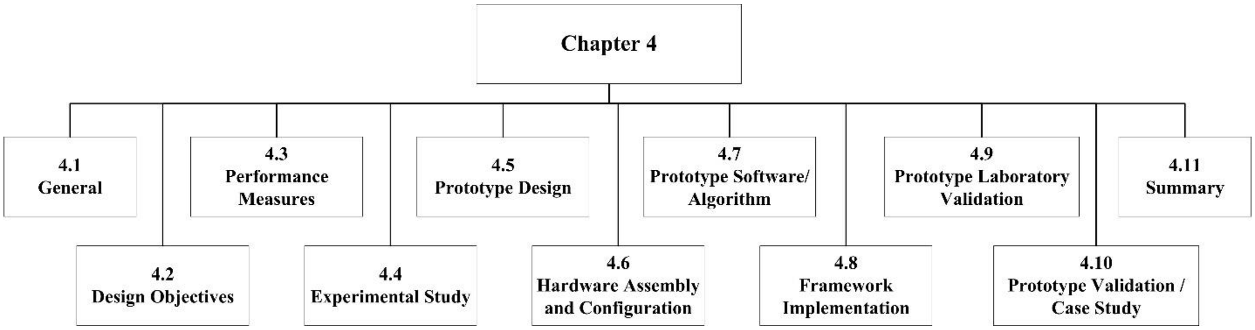

Chapter 4
4.1 General
4.2 Design Objectives
4.3 Performance Measures
4.4 Experimental Study
4.5 Prototype Design
4.6 Hardware Assembly and Configuration
4.7 Prototype Software/ Algorithm
4.8 Framework Implementation
4.9 Prototype Laboratory Validation
4.10 Prototype Validation / Case Study
4.11 Summary

Figure 4-1: Chapter 4 Overview

The developed prototype is designed with a special focus on earthmoving operation, however, it can be easily reconfigured for other outdoor construction operations such as concreting, steel structure, building envelopes, and landscape …etc.

In order to follow the rapid prototyping and objective driven design principles described in chapter 3, prototypes design objectives and performance measures are defined in the following sections.

4.2 Design Objectives

The main design objectives for the SA-GPS prototype are:

- To design prototypes for tracking earthmoving operation with a focus on improve the accuracy and robustness.
- To implement effective algorithms for data aggregation and processing to support near real-time progress tracking.
- The prototype must be able to work autonomously with minimal user intervention.
- Extraction of timely and accurate actionable information and maximizing the useful utilization of collected data, to support near real-time productivity analysis.
- The prototype must be scalable for application on real construction jobsites.

4.3 Performance Measures

The prototype development process requires a methodical and iterative approach through all levels of data collection and analysis. The five main performance measurement are explained as following:

- Accuracy: The main performance measure in comparing the developed prototype to traditional methods is the productivity measurement accuracy. The higher the accuracy, the better the system; however, there is often a trade-off between accuracy and other characteristics such as cost. The accuracy is measured as the average error in the productivity estimates.
- Latency: The prototype latency is attributed to hardware, computing, and human intervention/efforts during data transfer and processing. The proposed prototype is designed to measure the productivity in near real-time, which requires fast and efficient data processing with minimal human intervention.

- Scalability: The proposed prototype is required to be deployed to any project size without any need for further adjustment or development.
- Robustness: The prototype robustness is defined by its ability to function normally even when some signals are not available. The proposed prototype is designed to have multiple data sources from different sensors and be able to function even if some sensor data is missing or corrupted.
- Cost: The proposed prototype must be cost effective with respect to traditional methods.

4.4 Experimental Study

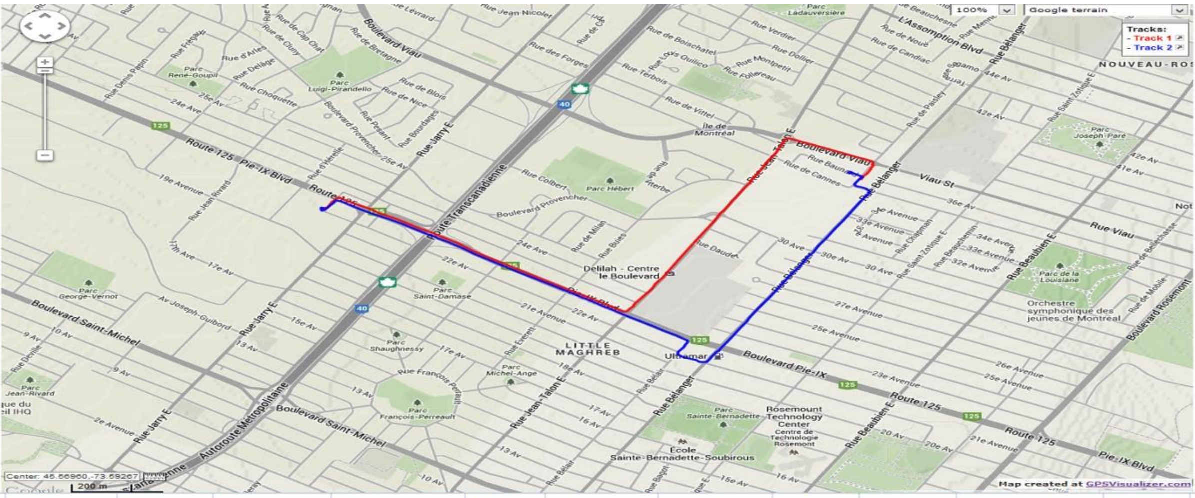
To design prototypes for the data acquisition models, an experimental study was conducted in laboratory and outdoor environments, to explore and test the suitability of various sensors, and wireless protocols for the intended application. A university lab kit was purchased from Libelium® (Spain). This kit included 7 microcontrollers, 18 sensors and 7 wireless communication modules. This experimental study investigated the functionality, accuracy, robustness and range of various components of the kit.

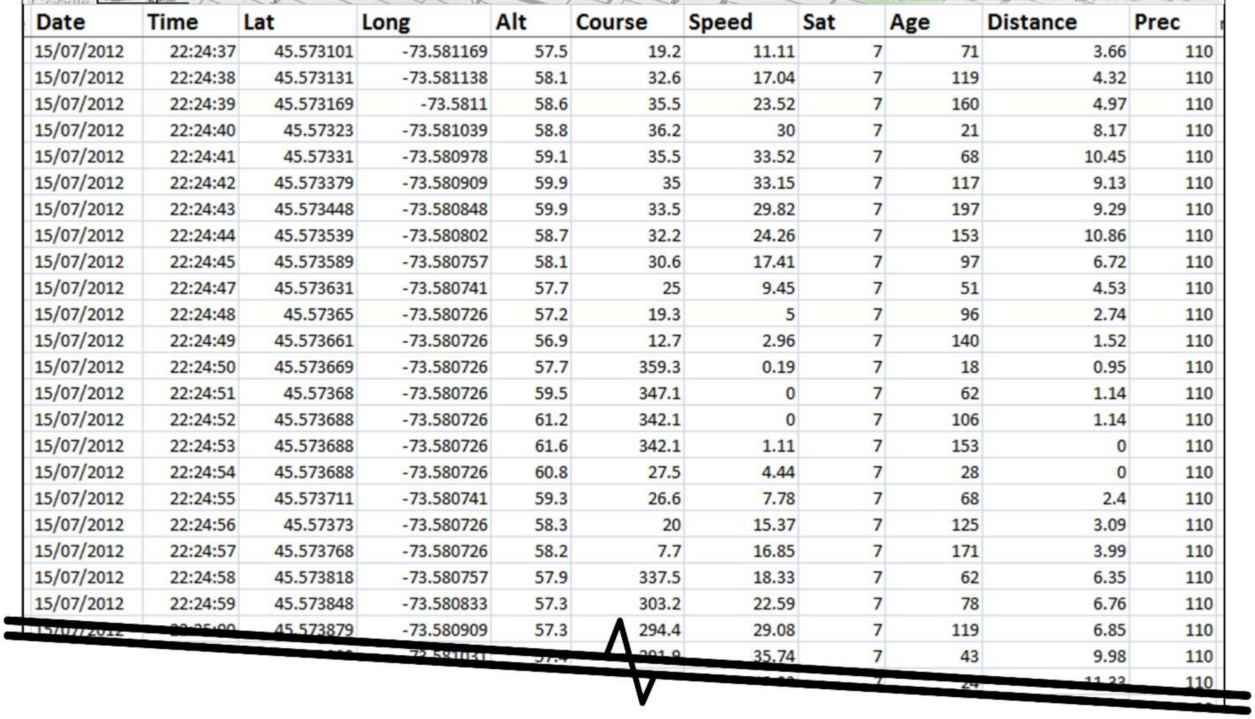
4.4.1 GPS Sensor

The GPS functionality is tested for the logging and reporting capabilities of accurate location for the outdoor environment. The prototype was programmed to report several parameters as shown in Figure 4-2:

- Date and Time
- Latitude, longitude in millionths of a degree and elevation in meters
- Course in degrees and speed in Km/hr.
- Number of locked satellites and age of fix in milliseconds.

- Distance between last logged coordinates in meters.
- Horizontal dilution of precision in 100ths.





Date	Time	Lat	Long	Alt	Course	Speed	Sat	Age	Distance	Prec
15/07/2012	22:24:37	45.573101	-73.581169	57.5	19.2	11.11	7	71	3.66	110
15/07/2012	22:24:38	45.573131	-73.581138	58.1	32.6	17.04	7	119	4.32	110
15/07/2012	22:24:39	45.573169	-73.5811	58.6	35.5	23.52	7	160	4.97	110
15/07/2012	22:24:40	45.57323	-73.581039	58.8	36.2	30	7	21	8.17	110
15/07/2012	22:24:41	45.57331	-73.580978	59.1	35.5	33.52	7	68	10.45	110
15/07/2012	22:24:42	45.573379	-73.580909	59.9	35	33.15	7	117	9.13	110
15/07/2012	22:24:43	45.573448	-73.580848	59.9	33.5	29.82	7	197	9.29	110
15/07/2012	22:24:44	45.573539	-73.580802	58.7	32.2	24.26	7	153	10.86	110
15/07/2012	22:24:45	45.573589	-73.580757	58.1	30.6	17.41	7	97	6.72	110
15/07/2012	22:24:47	45.573631	-73.580741	57.7	25	9.45	7	51	4.53	110
15/07/2012	22:24:48	45.57365	-73.580726	57.2	19.3	5	7	96	2.74	110
15/07/2012	22:24:49	45.573661	-73.580726	56.9	12.7	2.96	7	140	1.52	110
15/07/2012	22:24:50	45.573669	-73.580726	57.7	359.3	0.19	7	18	0.95	110
15/07/2012	22:24:51	45.57368	-73.580726	59.5	347.1	0	7	62	1.14	110
15/07/2012	22:24:52	45.573688	-73.580726	61.2	342.1	0	7	106	1.14	110
15/07/2012	22:24:53	45.573688	-73.580726	61.6	342.1	1.11	7	153	0	110
15/07/2012	22:24:54	45.573688	-73.580726	60.8	27.5	4.44	7	28	0	110
15/07/2012	22:24:55	45.573711	-73.580741	59.3	26.6	7.78	7	68	2.4	110
15/07/2012	22:24:56	45.57373	-73.580726	58.3	20	15.37	7	125	3.09	110
15/07/2012	22:24:57	45.573768	-73.580726	58.2	7.7	16.85	7	171	3.99	110
15/07/2012	22:24:58	45.573818	-73.580757	57.9	337.5	18.33	7	62	6.35	110
15/07/2012	22:24:59	45.573848	-73.580833	57.3	303.2	22.59	7	78	6.76	110
15/07/2012	22:25:00	45.573879	-73.580909	57.3	294.4	29.08	7	119	6.85	110
			-73.581031	57.4	291.9	35.74	7	43	9.98	110
							7	24	11.33	110

Figure 4-2: GPS Tracking and Logging Field Test

DOP is an indicator of three dimensional positioning accuracy as consequence of relative position of GPS satellites with respect to a GPS receiver. Table 4-1 shows the range of values for the dilution of precision for the GPS and their meaning in respect to the location measurement. The DOP of the tested GPS module ranged from 2.4 to 0.7, where better DOP was achieved after locking on 7 to 10 satellites as shown in Figure 4-3.

Table 4-1: Dilution of Precision Values

DOP Value	Rating	Description
Less than 1	Ideal	This is the highest confidence level to be used for applications demanding the highest possible precision at all times.
1-2	Excellent	Positional measurements are considered accurate enough to meet all but the most sensitive applications.
2-5	Good	Positional measurements could be used to make reliable in-route navigation.
5-10	Moderate	Positional measurements could be used for calculations, but more open view of the sky is required.
10-20	Fair	Represents a low confidence level. Very rough estimate of the current location.
Greater than 20	Poor	At this level, measurements are inaccurate by as much as 300 meters.

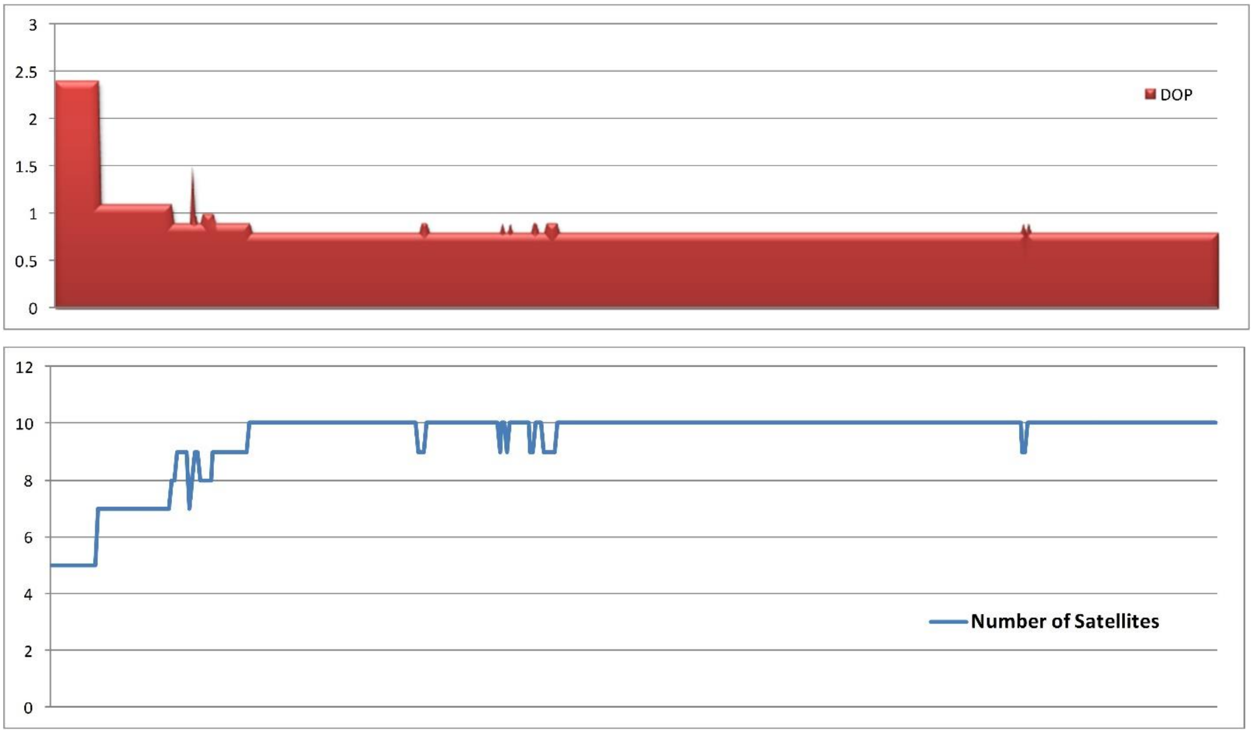


Figure 4-3: Measured DOP and Number of Satellites

4.4.2 3 Axel Accelerometer Sensor

The 3D accelerometer had been tested for the detection of aggressive driver behavior, and the detection of dump action of hauling trucks. The aggressive driver behavior was tested on a truck during a road test. 50 accelerations and braking runs were performed, of which 12 harsh accelerations and 12 harsh brakes. The algorithm was able to detect 10 of the harsh acceleration and 11 of the harsh brakes, with a detection percentage of 83.33% and 91.67% respectively. The higher detection accuracy in braking is due to the surge measured in x axis acceleration because the accelerometer experiences a force pushing it to the front. This surge is significant even when the brake is applied at low speed. Figure 4-4 illustrates a sample of the level of accelerations measured during the test.

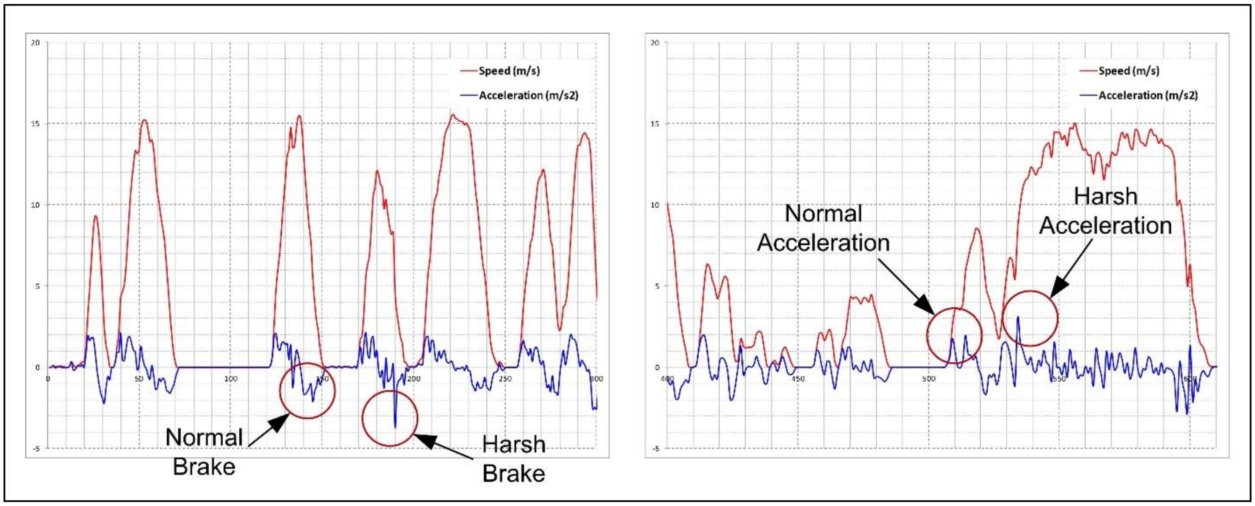


Figure 4-4: Truck Operator Behavior Monitoring Field Test using SA-GPS

The dump action of hauling truck was simulated to calculate the tilt angle of the truck bed as shown in Figure 4-5. The accelerometer measures the projection of the gravity vector on the sensing axis. The amplitude of the sensed acceleration changes as the sin of the angle α between the sensitive axis and the horizontal plane.

$$A = g \times \sin(\alpha) \tag{4-1}$$

The raw accelerometer signal was noisy and needed to be filtered. A moving average filter was applied to the signal, where the window for the filter was varied from 0.1 to 2 seconds. It was found out that a 1 Second filter removed the noise while keeping acceptable lag time.

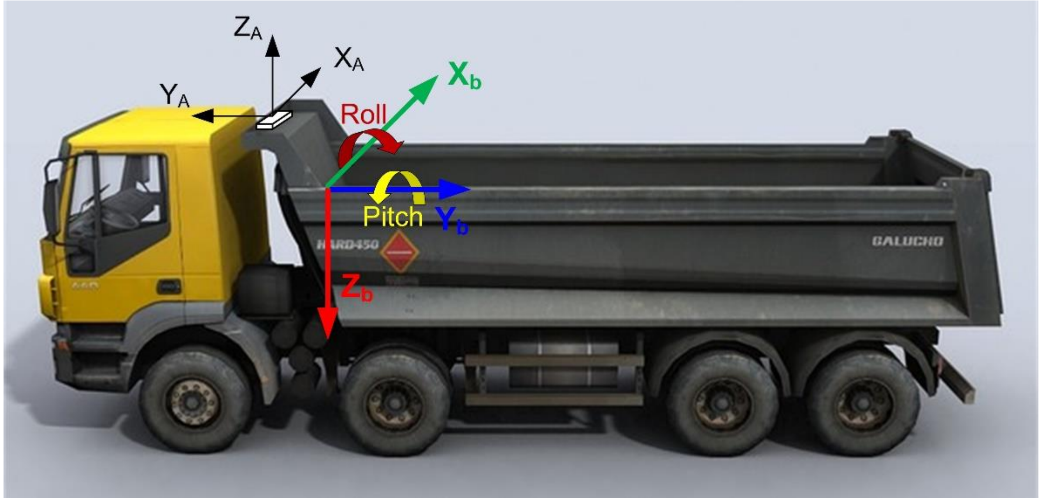


Figure 4-5: Accelerometer Installed on Hauling Truck

The test was conducted using the setup indicated in Figure 4-6. The sensor was attached to a flat piece of cardboard, and placed flat on the floor. The test starts by recording the readings of the accelerometer in the flat position, and then the cardboard is lifted from one end for a tilt angle of approximately 45^0 degrees.

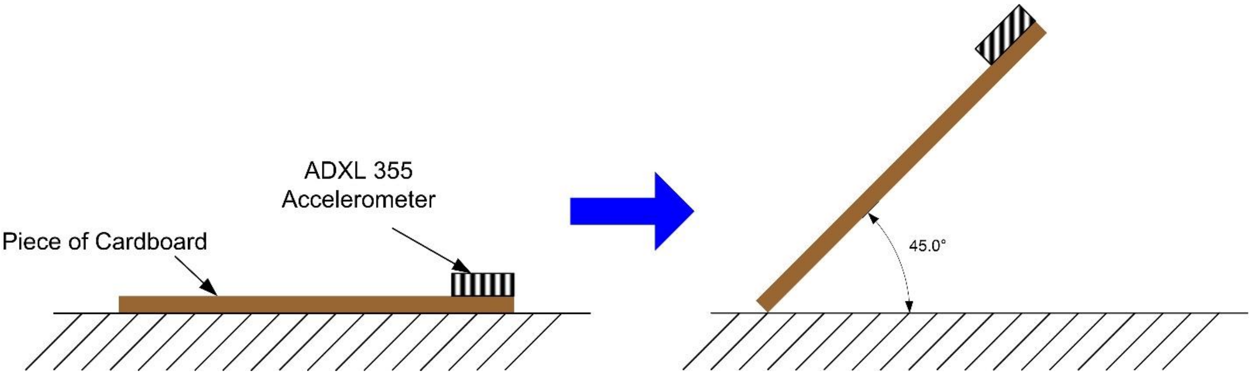


Figure 4-6: ADXL355 Accelerometer Test Setup

The measured acceleration by the ADXL355 accelerometer in the X-axis is plotted against time as indicated in

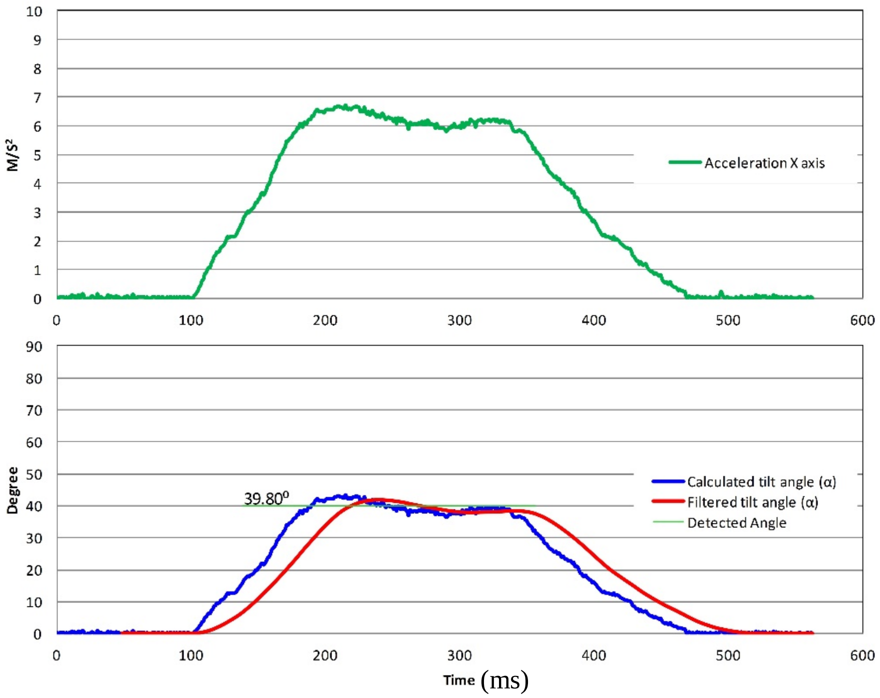


Figure 4-7. The blue curve is the calculated tilt angle. A 1 second filter was applied to the calculated angle value to smooth the readings and enable true detection as shown by the red curve. The results indicate that the average measured angle is 39.80^0 degrees.

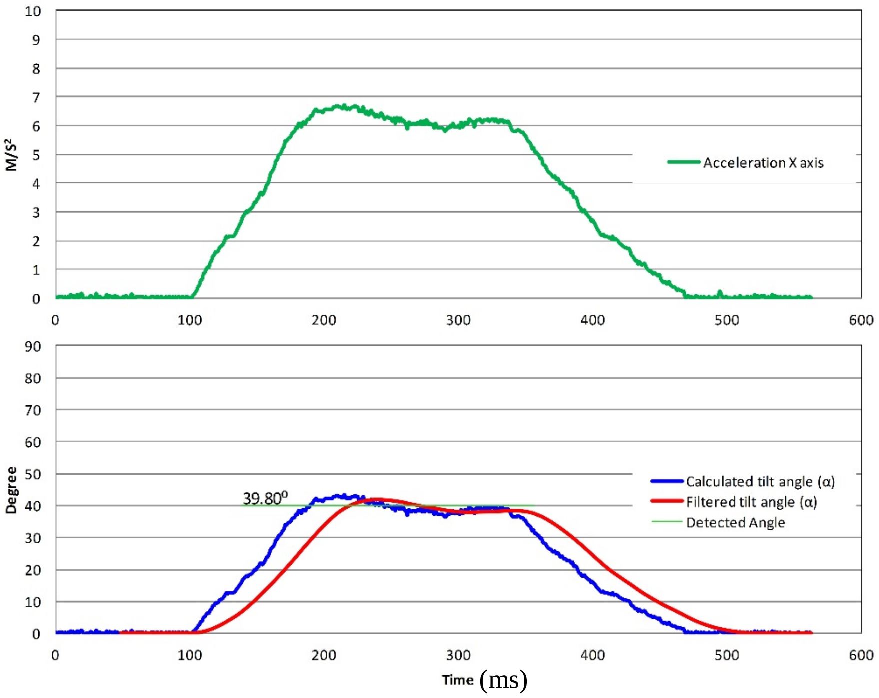


Figure 4-7: Accelerometer Test Results

4.4.3 Bluetooth Proximity

Bluetooth is used to detect equipment proximity to each other, therefore a test was conducted to measure the average proximity detection time. Where, 860 data sets were collected in outdoor environments with line of sight and non-line of sight, Figure 4-8 illustrates the histogram for the proximity detection time in seconds. The average detection time was found to be 10.52 seconds.

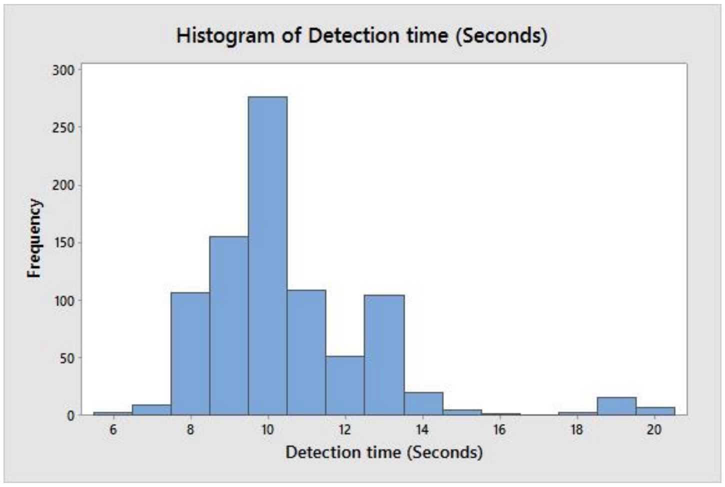


Figure 4-8: Bluetooth Proximity Detection Time

The cumulative distribution function of the Bluetooth detection time illustrated in Figure 4-9 shows that with a 95% confidence level, the detection time is 14.16 seconds.

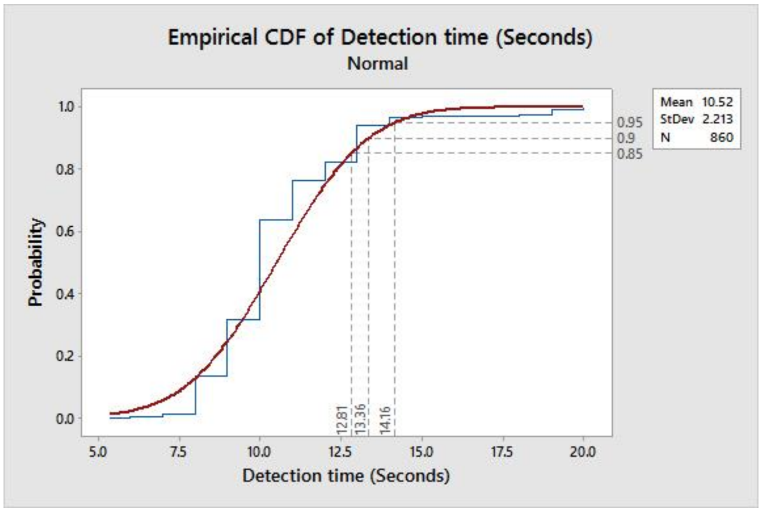


Figure 4-9: Bluetooth Proximity Detection Time CDF

In the context of detecting moving objects, it is important to calculate the maximum object speed. Figure 4-10 illustrates the moving object speed verses the detection time window, given a range of 100 m for the detection. It is possible to detect moving with an average speed up to 25 Km/hr within a 14.16 seconds window.

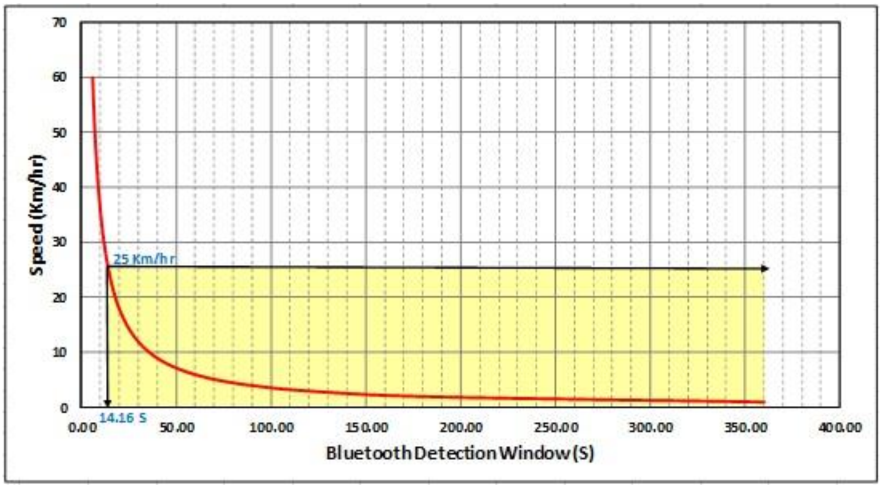


Figure 4-10: Moving Object Speed Vs Bluetooth Detection Time

4.4.4 Barometric Pressure Sensor

The BMP180 barometric pressure sensor was tested to evaluate its performance for detecting change in elevation. The test setup shown in Figure 4-11 was utilized by placing the barometric pressure sensor in position 1 for 30 seconds, and then placing it in position 2 for another 30 seconds.

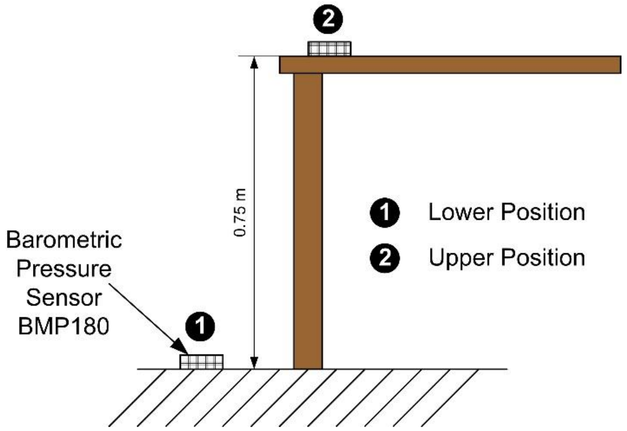


Figure 4-11: Barometric Pressure Sensor Test Setup

A 1 second moving average filter was utilized to smooth and remove noise from the sensor raw measurement as shown in Figure 4-12. While the actual elevation difference between position 1 and 2 is 0.75 m, the calculated elevation from measured barometric pressure values, was 0.60 m, which yield a 20% error as shown in Figure 4-12. However, in the context of the proposed application, this accuracy is adequate enough for the designed purposes.

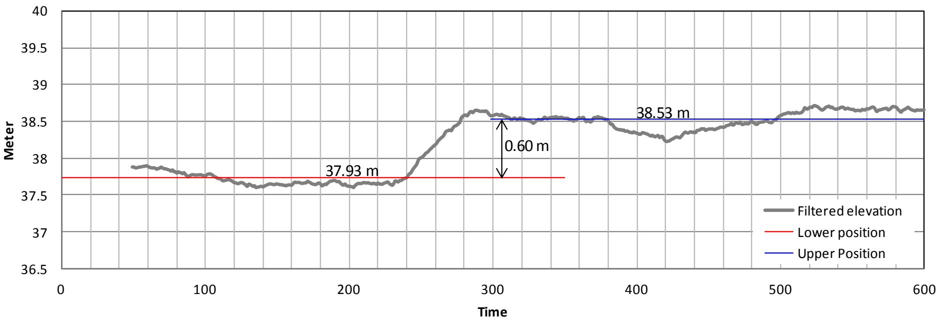


Figure 4-12: Barometric Pressure Sensor Test Results

The barometric pressure sensor was also tested for elevation estimation in the EV-building at Concordia University. The sensor was carried during travelling in the elevator as indicated in Figure 4-13. The elevation was calculated from the barometric pressure data using the following equation:

$$Elevation\ (m) = 44330 * \left(1 - \left(\frac{P}{P_0}\right)^{\frac{1}{5.255}}\right) \qquad (4\text{-}2)$$

Where P is the measured pressure in hPa and P_0 is the atmospheric pressure at sea level

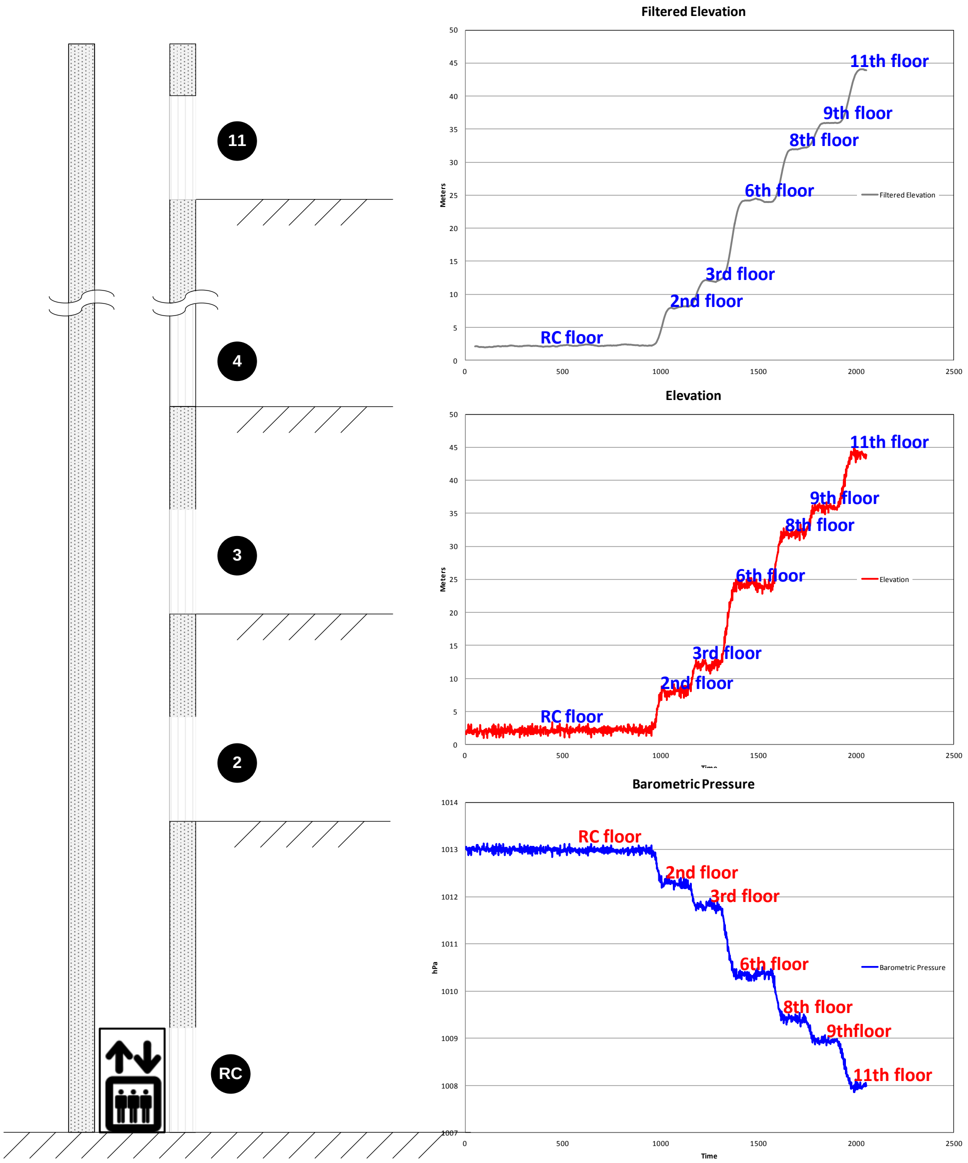


Figure 4-13: Barometric Pressure Test in Elevator

4.4.5 Wireless Protocols

In order to experiment and investigate indoor propagation of different wireless networks, 21 experiments are conducted and 1752 data sets are recorded for more than 876 minutes (grand total of all experiments). All these experiments are performed in different scenarios either in terms of number of nodes, distance between the nodes, line of sight and finally, in terms of topology i.e. straight-line/grid as shown in Figure 4-14.

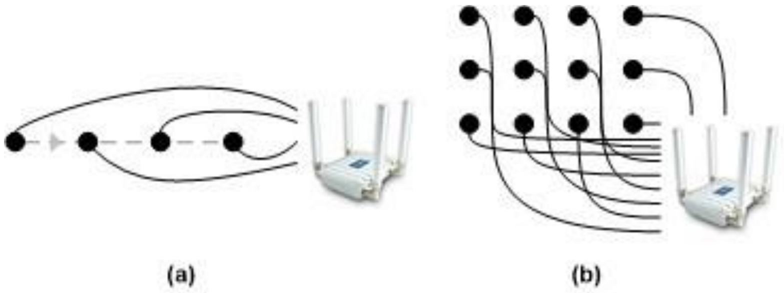


Figure 4-14: Straight Line (a) and Grid (b) deployment

A Waspmote platform is used to build the mobile nodes for the experimentations, which includes a microcontroller operating at 14MHz, 128K of memory, a wireless transceiver interface socket, and a USB interface for device programming and logging. Each device operates on rechargeable batteries. Its wireless interface socket is compatible with different communication protocols (WLAN, Bluetooth, Zigbee and Synapse SNAP) and frequencies (2.4GHz, 868MHz, 900MHz) as shown in Figure 4-15.

Figure 4-15: Waspmote platform mobile nodes

Four wireless technologies are used in the experiments, in particular, Wireless Local Area Networks (WLAN), Bluetooth, Zigbee and Synapse SNAP. Their technical details with respect to frequency, output power, range, sensitivity and cost are summarized in Table 4-2. All experiments are summarized in Table 4-3. The setup for straight line experiments is shown in Figure 4. The path is 20 m long, straight track with 20 waypoints with a distance of 1 m between two consecutive waypoints. Two stationary sensor nodes are placed next to the track at 0 m and 21 m. Each experiment is repeated for each of the four wireless networks (WLAN, Bluetooth, Zigbee and Synapse SNAP). The setup for grid setting experiments is shown in figure 5. The grid size changes from 3m x 3m to 6m x 6m, with a distance of 1 m between two consecutive nodes. One stationary sensor node is placed next to the grid at 1 m and center of the grid. Each experiment is repeated for each of the four wireless protocols (WLAN, Bluetooth, Zigbee and Synapse SNAP).

Table 4-2: Wireless protocols hardware

Wireless Network	**Bluetooth**	**Zigbee**	**WLAN**	**Synapse**
Hardware Module	Roving Network (RN-41)	Xbee 802.15.4	Roving Network (RN-171)	RF300
Frequency	2.4 GHz	2.4 GHz	2.4 GHz	915MHz
Data Rate Kbps	3x1024	250	921	150
Power dBm	15	0	10	20
Range m	100	90	75	250
Sensitivity dBm	-80	-92	-83	-99
Tx current mA	65	35	120	85
Rx current mA	35	50	38	18.5
Cost	$24.95	$31.95	$39.95	$28.95

RSSI of data packets received by the mobile node is measured using a program written using C++ and installed on the fixed node. In this program, a stationary node sends a PING packet to the mobile node and the mobile node responds with another data packet that contains the RSSI value with which the PING packet was received. The two stationary nodes send PING packets in a strict round-robin fashion to avoid packet collisions. Each node sends 3 packets per second, which results in 3 RSSI samples per second per link.

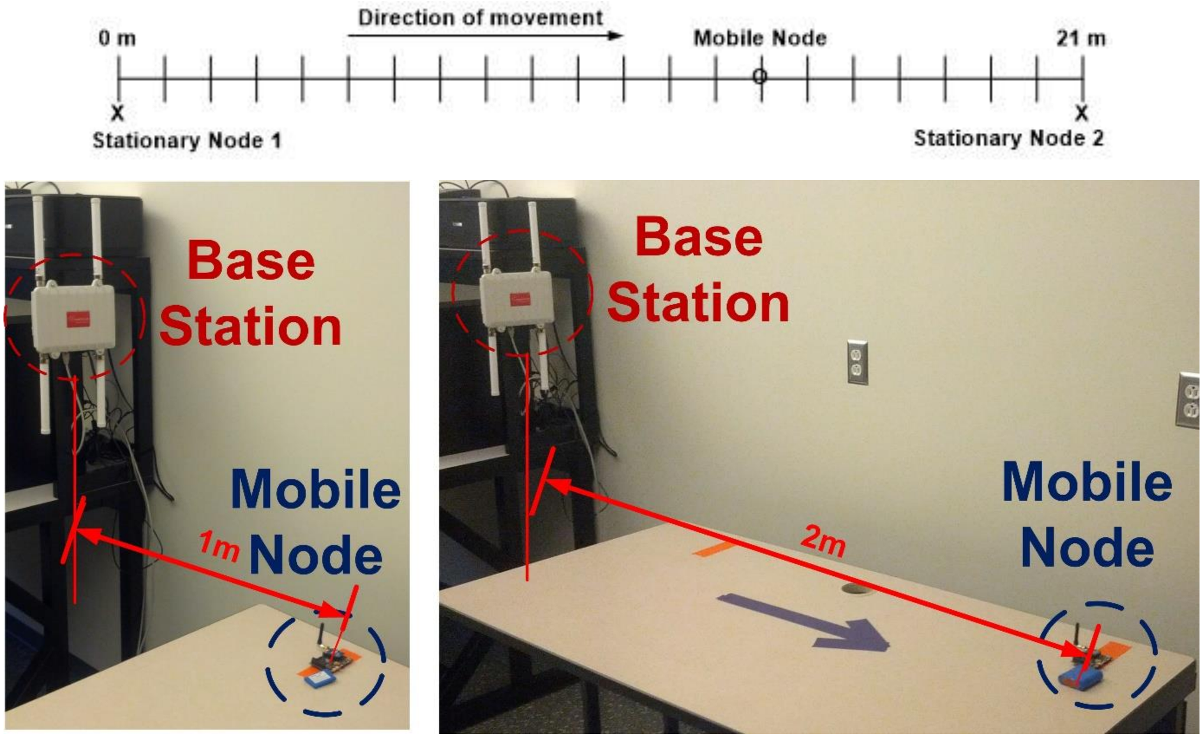


Figure 4-16: Experimental setup for evaluation of wireless networks in a straight line setting

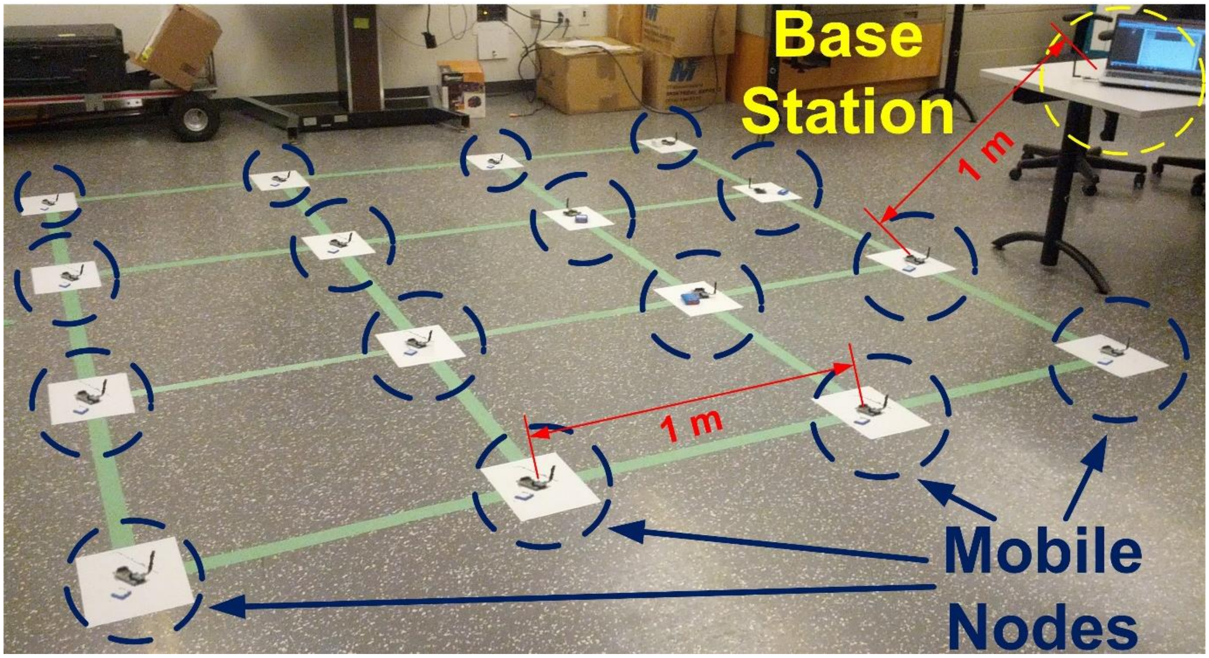
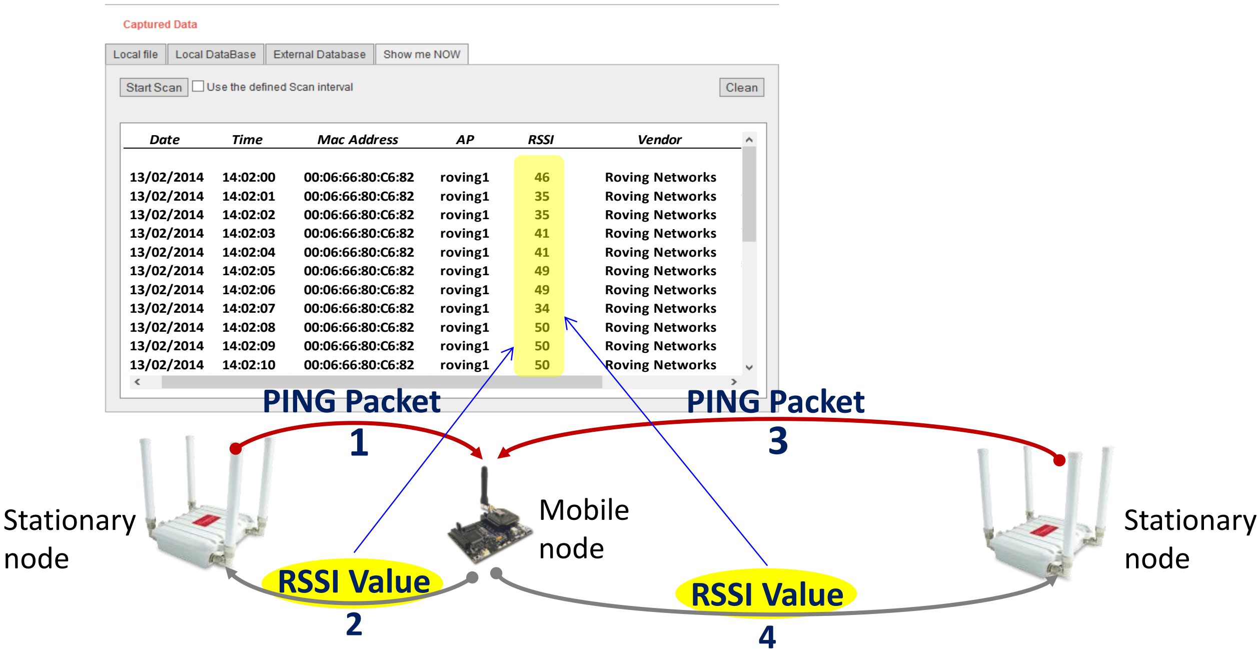


Figure 4-17: Experimental setup for grid setting

Table 4-3: Experiments Scenarios

Exp. #	Nodes #	Dist. (m)	TX Level	LOS	Topology
1	3	1=>20	0	Y	S.L
2	3	1=>20	0	N	S.L
3	3	1=>20	1	Y	S.L
4	3	1=>20	1	N	S.L
5	3	1=>20	2	Y	S.L
6	3	1=>20	2	N	S.L
7	3	1=>20	3	Y	S.L
8	3	1=>20	3	N	S.L
9	3	1=>20	4	Y	S.L
10	3	1=>20	4	N	S.L
11	3	1=>20	5	Y	S.L
12	3	1=>20	5	N	S.L
13	3	1=>20	6	Y	S.L
14	3	1=>20	6	N	S.L
15	4x4	3	6	Y	Grid
16	4x4	6	6	Y	Grid
17	6x6	3	6	Y	Grid
18	6x6	6	6	Y	Grid
19	3X4	3	6	Y	Grid
20	3X6	3	6	Y	Grid
21	4X6	3	6	Y	Grid



Date	Time	Mac Address	AP	RSSI	Vendor
13/02/2014	14:02:00	00:06:66:80:C6:82	roving1	46	Roving Networks
13/02/2014	14:02:01	00:06:66:80:C6:82	roving1	35	Roving Networks
13/02/2014	14:02:02	00:06:66:80:C6:82	roving1	35	Roving Networks
13/02/2014	14:02:03	00:06:66:80:C6:82	roving1	41	Roving Networks
13/02/2014	14:02:04	00:06:66:80:C6:82	roving1	41	Roving Networks
13/02/2014	14:02:05	00:06:66:80:C6:82	roving1	49	Roving Networks
13/02/2014	14:02:06	00:06:66:80:C6:82	roving1	49	Roving Networks
13/02/2014	14:02:07	00:06:66:80:C6:82	roving1	34	Roving Networks
13/02/2014	14:02:08	00:06:66:80:C6:82	roving1	50	Roving Networks
13/02/2014	14:02:09	00:06:66:80:C6:82	roving1	50	Roving Networks
13/02/2014	14:02:10	00:06:66:80:C6:82	roving1	50	Roving Networks

Figure 4-18: RSSI measurement program interface

The collected data was averaged and presented in Figure 4-19, which clearly show that the received signal strength at each point is declining as expected. Even though, the declining rate is inconsistent with all wireless technologies. For example the node at 4 meters receives a weaker signal on the incoming packets with Bluetooth and WLAN than the node at 5 meters. Also the node at 7 meters receives a weaker signal on the incoming packets with Zigbee than the node at 8 meters. In all measurements Synapse hardware showed more consistency in returning RSSI values in declining order.

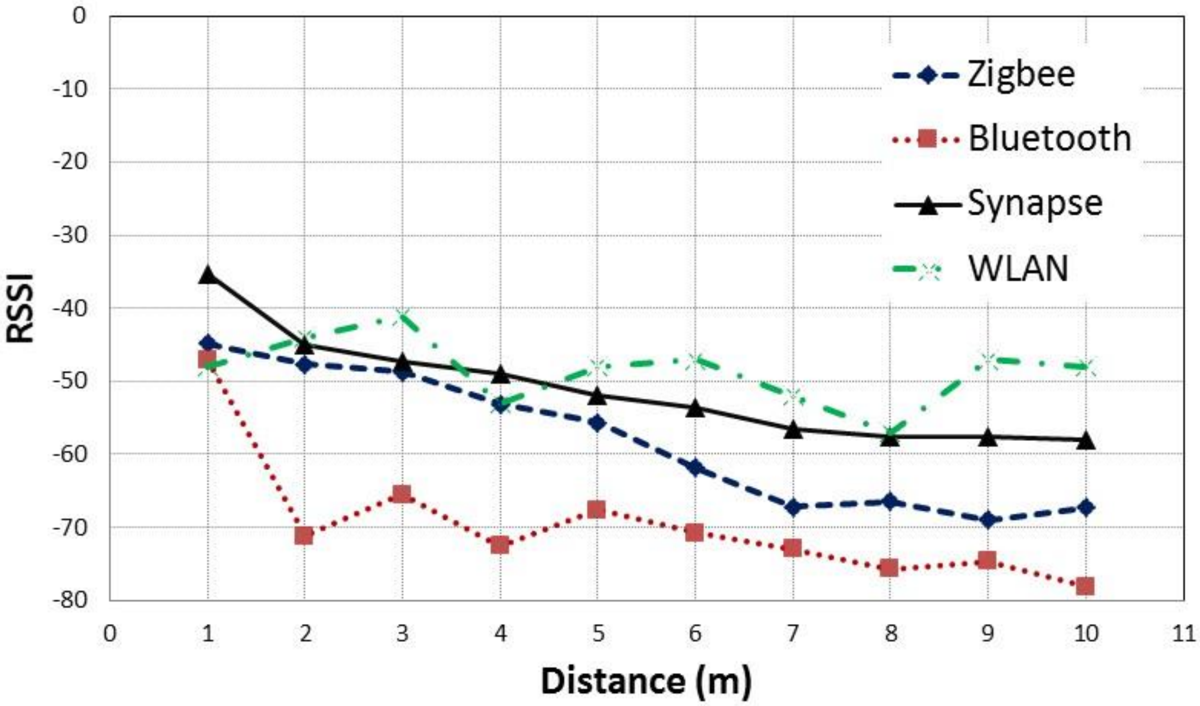


Figure 4-19: RSSI measurement straight line formation

A major source of error when measuring RSSI is due to multipath effects caused by objects in the environment. In the office environment, where the tests were performed, the radio environment is likely to change between every measurement point as the room contains quite many things that could cause multipath effects. Multipath in indoor environment is caused by multiple signal reflections from walls, ceilings and other objects as shown in Figure 4-20. This directly affects the measured RSSI. Any type of reflected signal that can be additive or destructive to the original signal is identified as multipath interference.

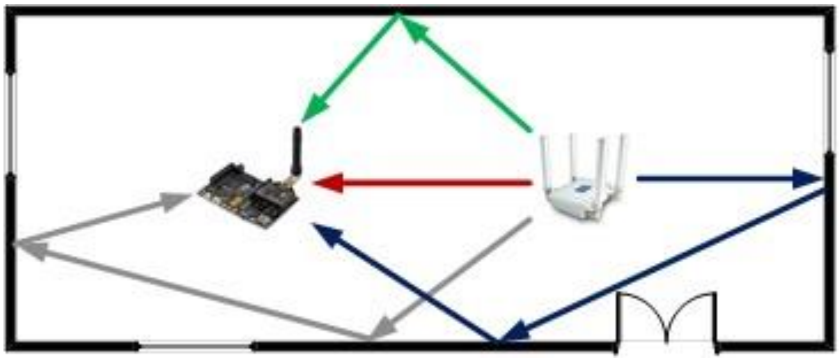


Figure 4-20: Multipath interference

As the signal strikes an objective, it can react in several ways, creating a reflection, scattering, refraction, diffraction or all of the above. Reflection is simply when the signal is reflected back towards the transmitter. Scattering occurs when the signal is scattered back towards the transmitter into multiple new signals. Refraction occurs when the signal is bent as it passes through an object and Diffraction happens when the signal changes direction as it passes around an object. RF signal strength is reduced as it passes through various materials. This effect is referred to as Attenuation. As more Attenuation is applied to a signal, its effective range will be reduced. The amount of Attenuation will vary greatly based on the composition of the material the RF signal is passing through.

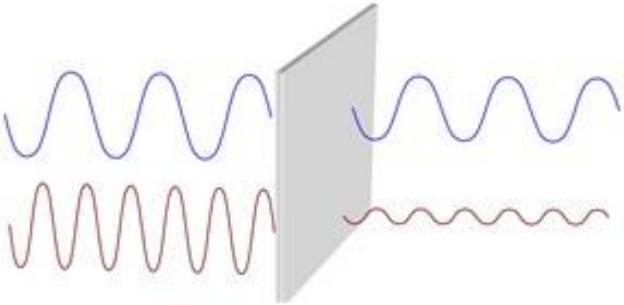


Figure 4-21: Attenuation interference

Using frequencies below 900 MHz can significantly improve connectivity in an indoor environment as shown in Figure 4-21. Low frequencies also exhibit more uniform signal propagation, which simplifies the network design and deployment problems.

Each antenna has its own radiation pattern that is not uniform. The measured RSSI value is impacted by the antenna orientation of the transmitting and receiving nodes. To quantify the antenna orientation impact, the average RSSI is measured at 24 different degrees with a fixed receiver node and a rotating transmitter node at a 1m distance in a relatively obstacle-free environment (No obstacles making reflections within three meters). As shown in Figure 4-22(a), the radiation pattern of the antenna is asymmetrical and suffers from distortion with difference in the measured RSSI. One of the main factors that cause the antenna orientation phenomenon is the magnetic field of the antenna, which is distorted by the interference from nearby devices. Figure 4-22(b) illustrates the coverage range, which is calculated based on the radiation pattern. Synapse has shown a wider and higher coverage range than the other three wireless technologies.

The results of the above experiments were scored on a level from 1 to 4, where 4 is the best and 1 is the worst performance in each criteria as shown in Table 4-4. The Synapse protocol has the highest total score.

Table 4-4: Wireless Protocols Test Summary

	Bluetooth	**Zigbee**	**WLAN**	**Synapse**
RSSI Vs Distance	2	4	1	4
Range	2	3	2	4
Attenuation	2	4	2	4
Cost	4	2	1	3
Total Score	10	13	6	15

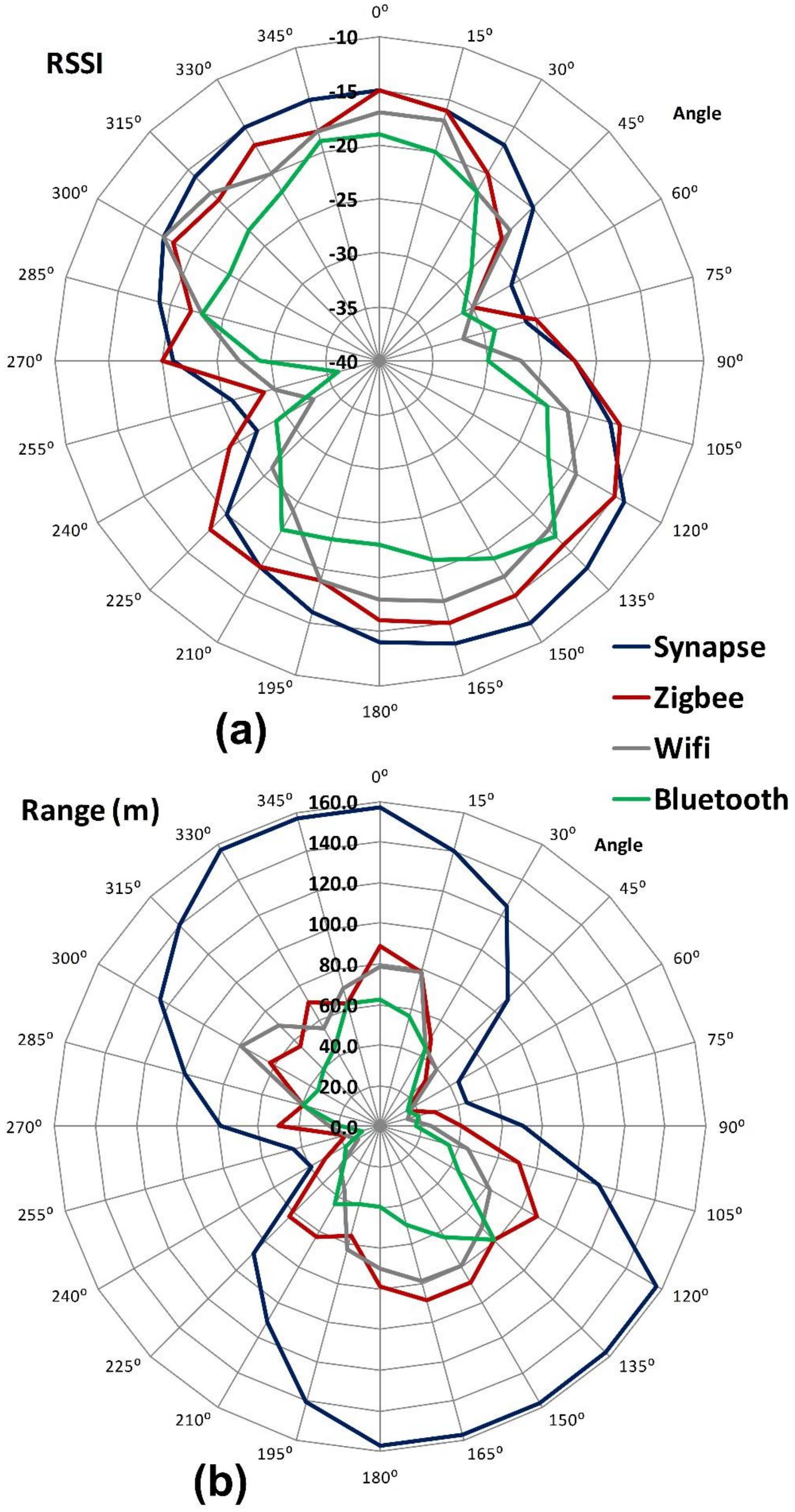


Figure 4-22: Antenna orientation and range

4.5 Prototype Design

The SA-GPS prototype consists of mobile units installed on equipment and gateway fixed unit installed on construction jobsite, the conceptual deployment of the developed prototype is shown in Figure 4-23. The mobile units collect data from each piece of equipment, and then data is sent to the gateway fixed unit. The gateway fixed unit performs data pre-processing and fusion then upload the data to a remote database server, where the final data processing and storage takes place.

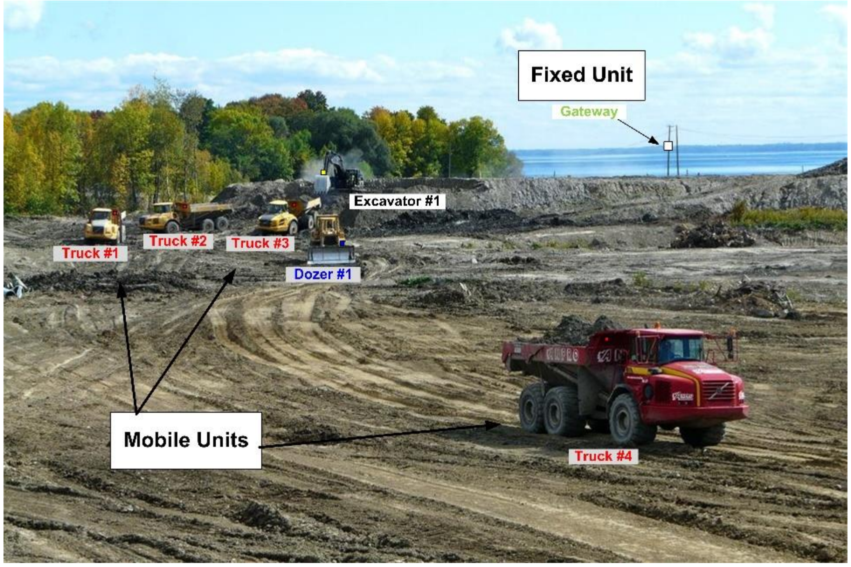


Figure 4-23: Conceptual deployment of Sensor Aided GPS Prototype

The rapid prototyping approach was utilized to realize the design for the SA-GPS, where a basic version of the prototype was configured, assembled and tested. From a data acquisition perspective, the SA-GPS prototype consists of three main entities, namely; data sources, network infrastructure, and data sinks. Data Acquisition Sources are sensors, which generates data regarding the construction operations. These sensors are directly integrated to collect various types of information about construction equipment utilization, and interactions, which requires several levels of configuration and automation. Wireless network infrastructure to provide end-to-end data routing and transfer. The function and performance of the network infrastructure are a crucial key for real-time application. Therefore, it was very important to efficiently configure and test its operation in lab experiments and outdoor environment. Data Acquisition Sinks are entities for data aggregation, processing and fusing. They interact with the network infrastructure to extract actionable information from raw data for later processing and analysis.

The required data acquisition entities for the developed prototype are illustrated in Figure 4-24. The developed prototype integrates redundant data sources to enhance its overall robustness.

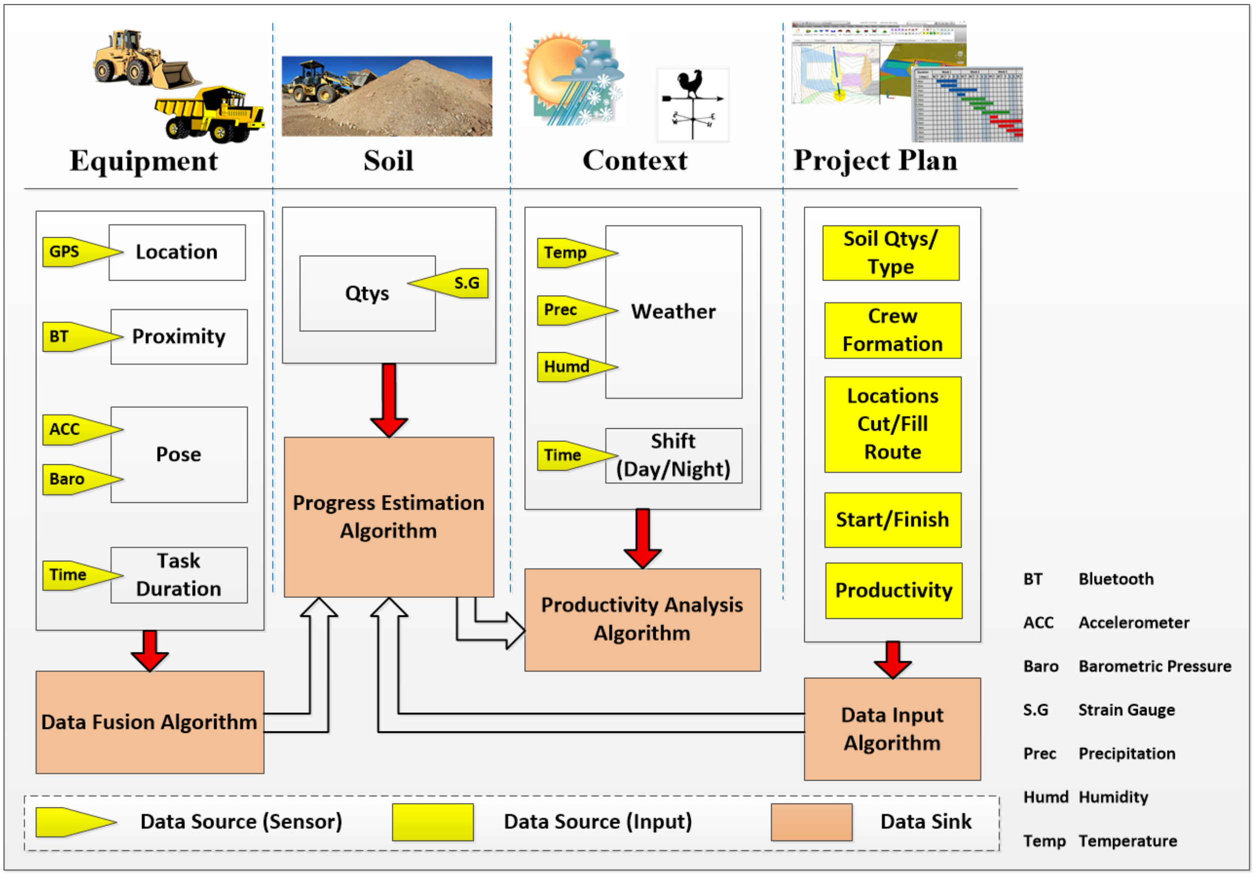


Figure 4-24: Proposed SA-GPS Entities

Selecting the sensor type depends on the hosting earthmoving equipment, data to be collected and the physical pose or movement of equipment that need to be captured. Table 4-5, illustrates sensor configurations for some of the most common heavy equipment used in earthmoving operations. Equipment positions are captured by the GPS modules, excavator and loader boom swing angle is sensed by accelerometer, excavator and loader bucket height is sensed by the barometric pressure sensor, truck bed tilt angle is sensed by accelerometer, and truck load weight is measured by strain gauges.

Table 4-5: Sensors Configuration for Earthmoving Equipment

Equipment Type	**Sensor**	**Measured data**
Hauling Truck	Three-axis accelerometer	Tilt-sensing of truck bed. Dynamic acceleration resulting from motion, shock, or vibration
	Strain gauges	Truck load weight
	Piezoelectric sensor	Truck bed vibration
Loader / Excavator	Three-axis accelerometer	Bucket tilt /boom swing
	Barometric pressure sensor	Bucket vertical movement
Dozer	Three-axis accelerometer (Two Sensors)	Blade tilt Ripper tilt
Scraper	Limit Switches (Three)	Status of apron, bucket and gate (open/close)

The fixed gateway unit is a Meshlium, which is a Linux router which contain 5 different radio interfaces: Wi-Fi, Synapse SNAP, 3G/GPRS, Bluetooth and ZigBee. The gateway has a 500MHz (x86) processor with 256MB RAM, and it has three data storage options including 160GB internal storage as shown in Figure 4-25. It can be powered by solar and/or battery. The gateway is enclosed in an aluminum IP65 casing, which allows placing in harsh outdoor construction environments.

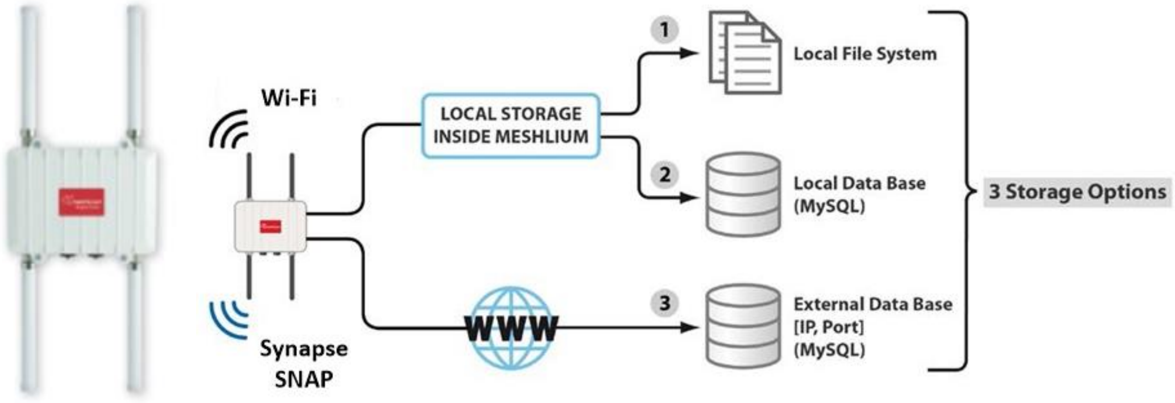


Figure 4-25: Meshlium Gateway

The proposed communication scheme for the SA-GPS prototype is depicted in Figure 4-26, where Synapse SNAP communication protocol is used to transfer the data between the mobile nodes and the fixed gateway node after pre-data fusion and processing. In order to get the big picture of the construction operation, data is post processed on the gateway node, by aggregating data from various mobile nodes and perform data fusion. Finally information gained from the data fusion is transfer using wireless LAN to server based database through the internet cloud.

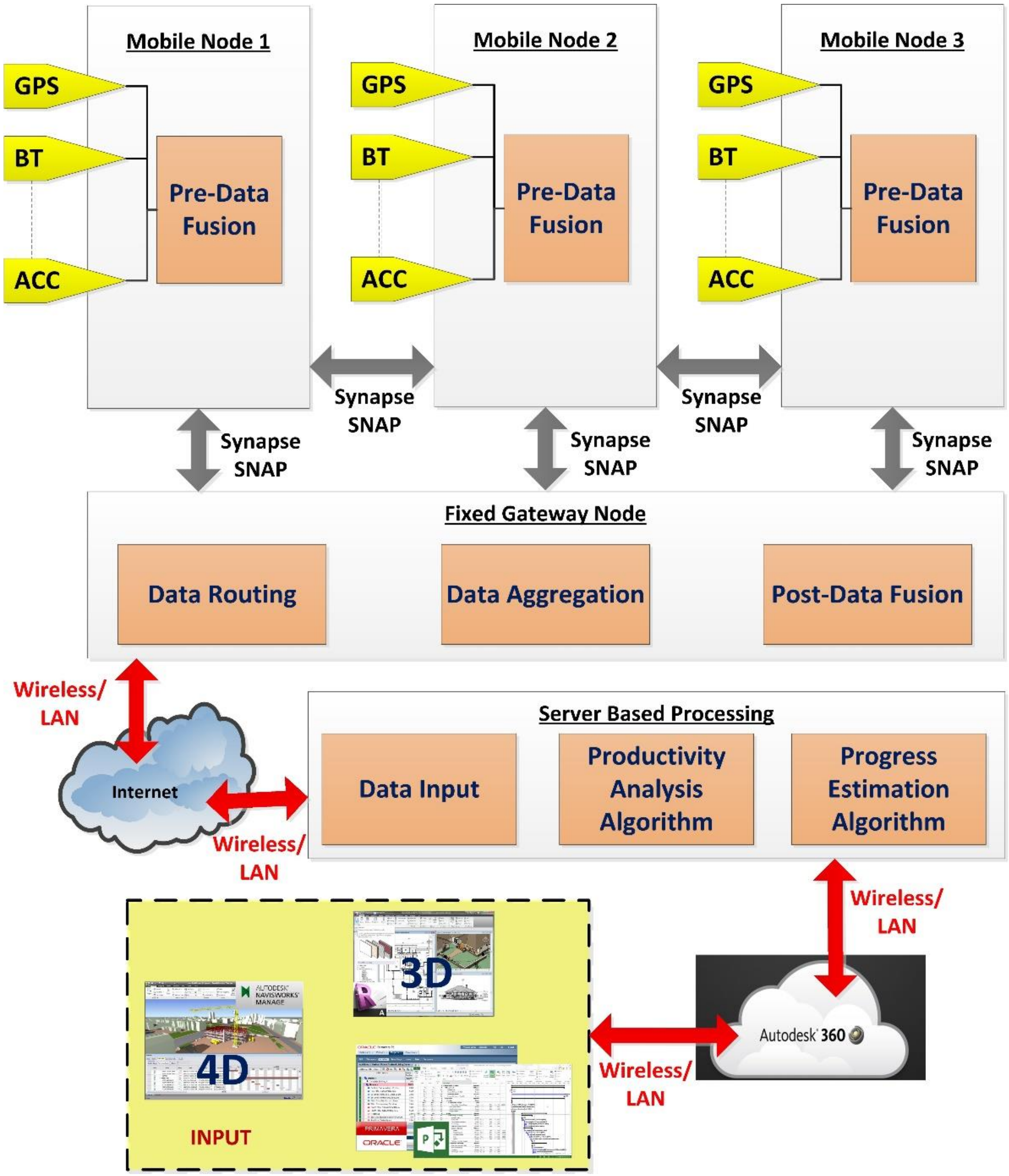


Figure 4-26: Prototype Communication Configuration

4.6 Hardware Assembly and Configuration

Three prototypes were developed: a loader mobile unit, an excavator mobile unit, and a truck mobile unit. The developed mobile unit prototype consists of a microcontroller with data logger, Bluetooth module, GPS module, Synapse RF module and a power supply. The block diagram of the mobile unit is shown in Figure 4-27 and Figure 4-28. The description of each hardware component is explained as follows:

- Arduino UNO microcontroller, which is based on the ATmega328, it has 14 digital input/outputs and 6 analog inputs. It is reasonably priced (about $25.00), and the development software is open source.
- Roving Networks (RN-41) module is a Class 1 Bluetooth Module with a range of up to 100 m.
- SkyTraq (Venus638FLPx) GPS module is a high performance, low cost, single chip GPS receiver with low power consumption, high sensitivity, and a low time-to-first-fix.
- Adafruit Industries (DS1307) data logger, which has a real-time clock with backup battery for up to seven years of timekeeping. It can fit any SD/MMC storage up to 32 GB.

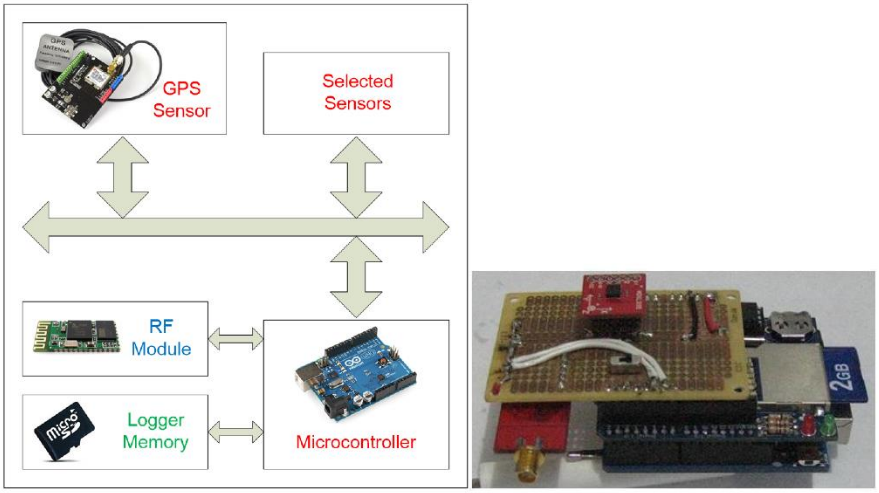


Figure 4-27: Mobile Unit Block Diagram and Hardware Prototype

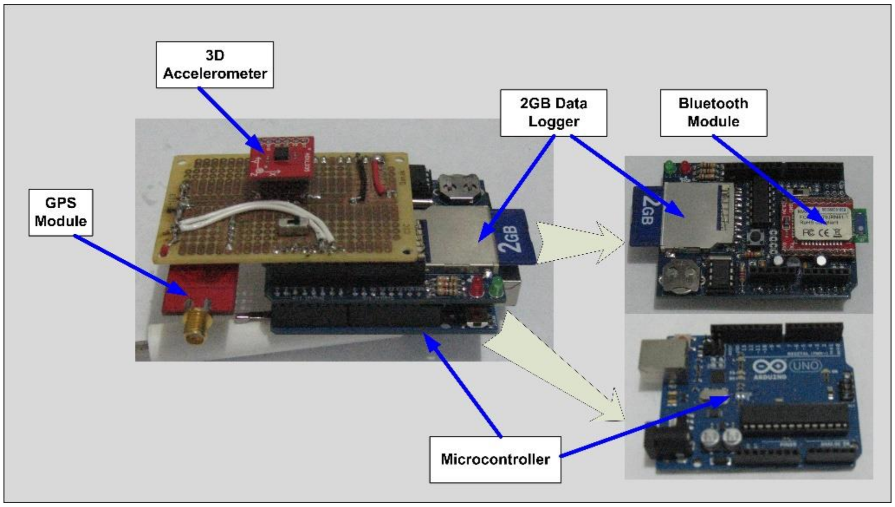


Figure 4-28: SA-GPS Basic Configuration Prototype

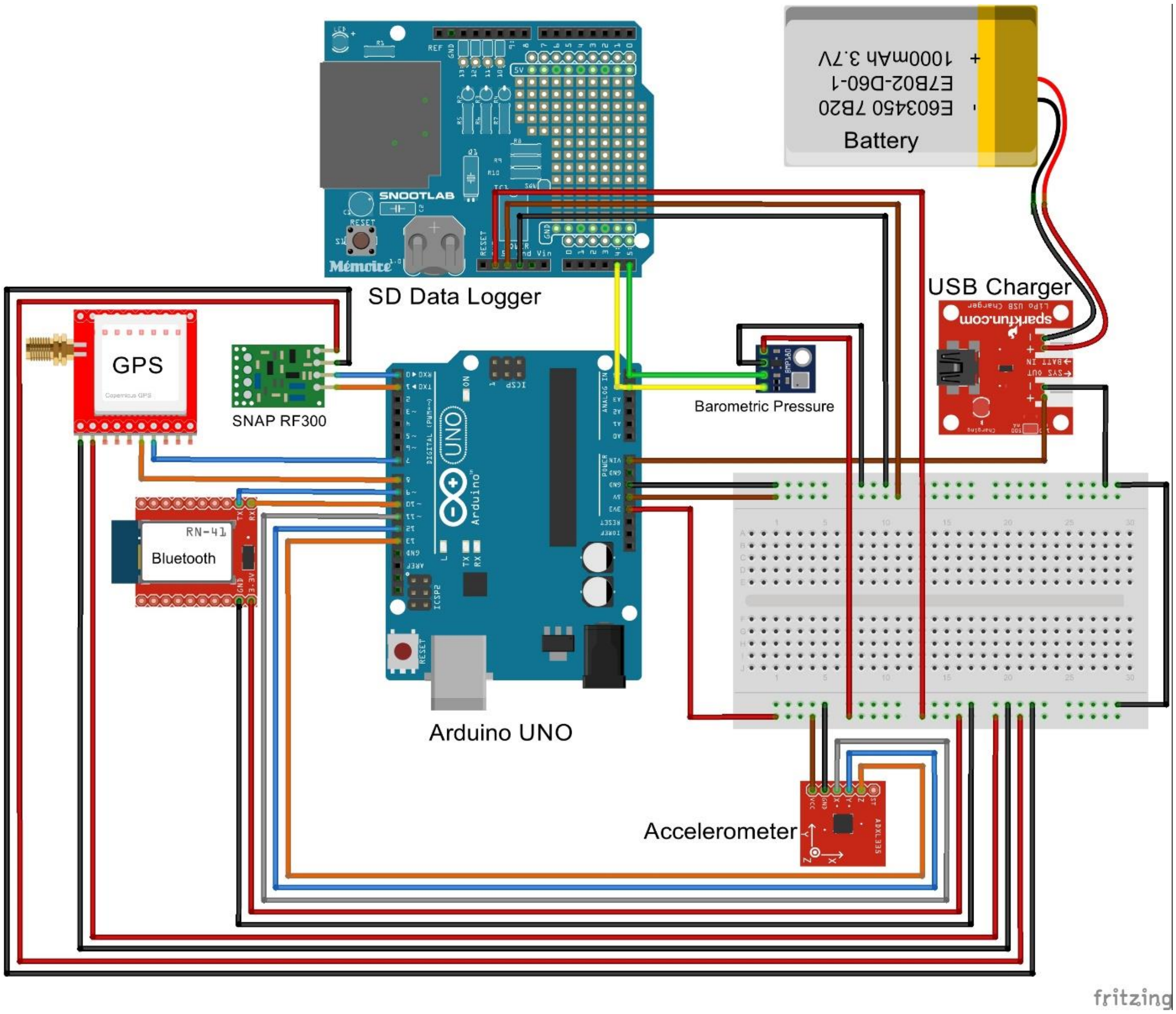


Figure 4-29: SA-GPS Loader/Excavator Prototype Wiring Diagram

The wiring diagram presented in Figure 4-29 depicts the assembly of the SA-GPS prototype for loader / excavator type of equipment. In the Loaded/Excavator prototype three-axis accelerometer from Analog Devices (ADXL335) is integrated to measure tilt and swing of the loader bucket and the excavator arm. Also, another prototype for hauling trucks was developed as shown in Figure 4-30

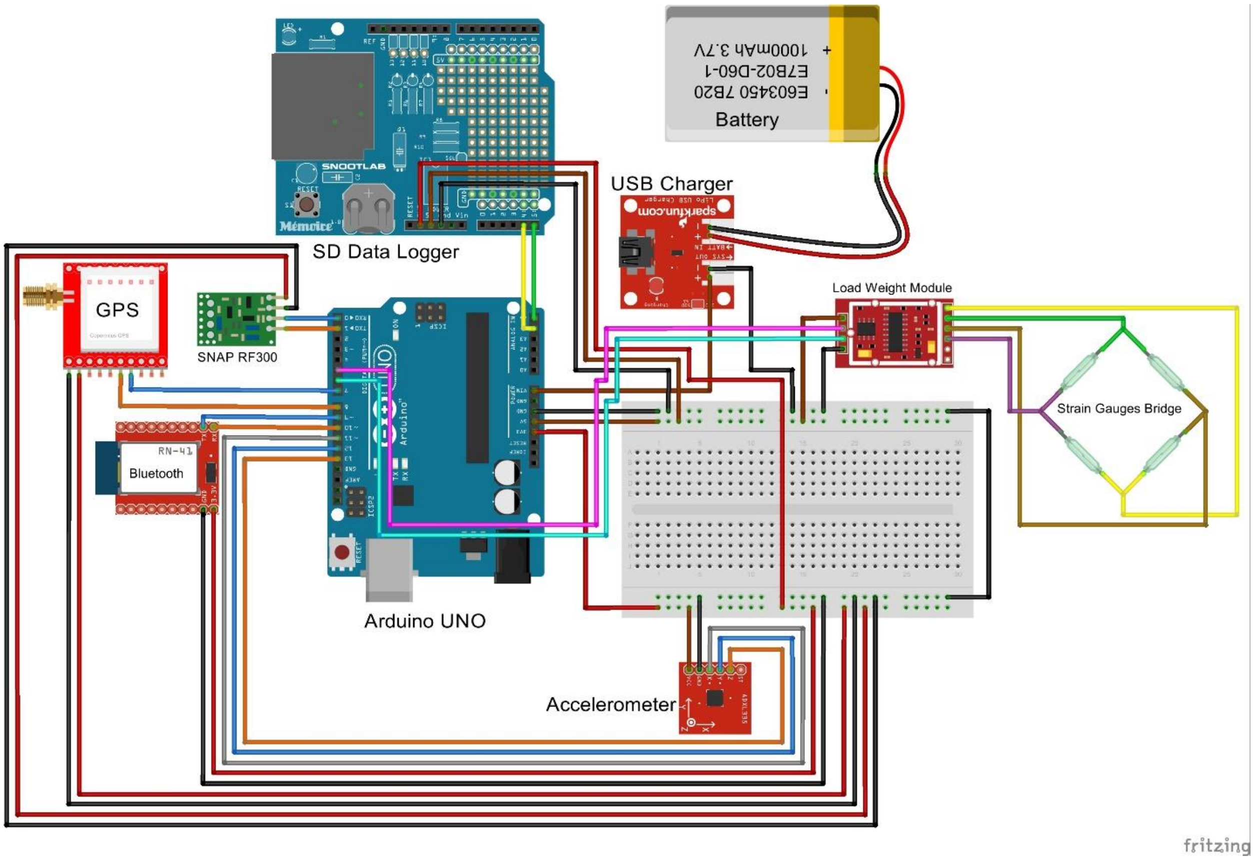


Figure 4-30: SA-GPS Hauling Truck Prototype Wiring Diagram

4.7 Prototype Software/ Algorithm

Four software algorithms are developed on the mobile nodes microcontrollers, namely: localization algorithm, proximity detection, payload volume estimation and data fusion algorithm.

4.7.1 Localization Algorithm

The localization algorithm is responsible for identifying the work zone for construction equipment based on the GPS data. Two localization algorithms are developed in this research: Finely algorithm, and fuzzy c-mean (FCM) algorithm. If the boundaries for the cut and fill locations are known, Finley algorithm is utilized. The steps of Finley (2007) algorithm are illustrated in Figure 4-31 and Figure 4-32.

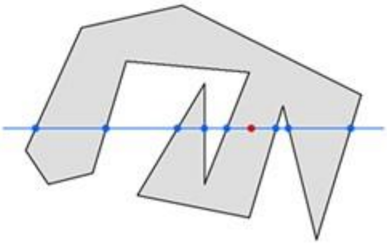


Figure 4-31: Finely (2007) Method for Solving Point in Polygon Problem

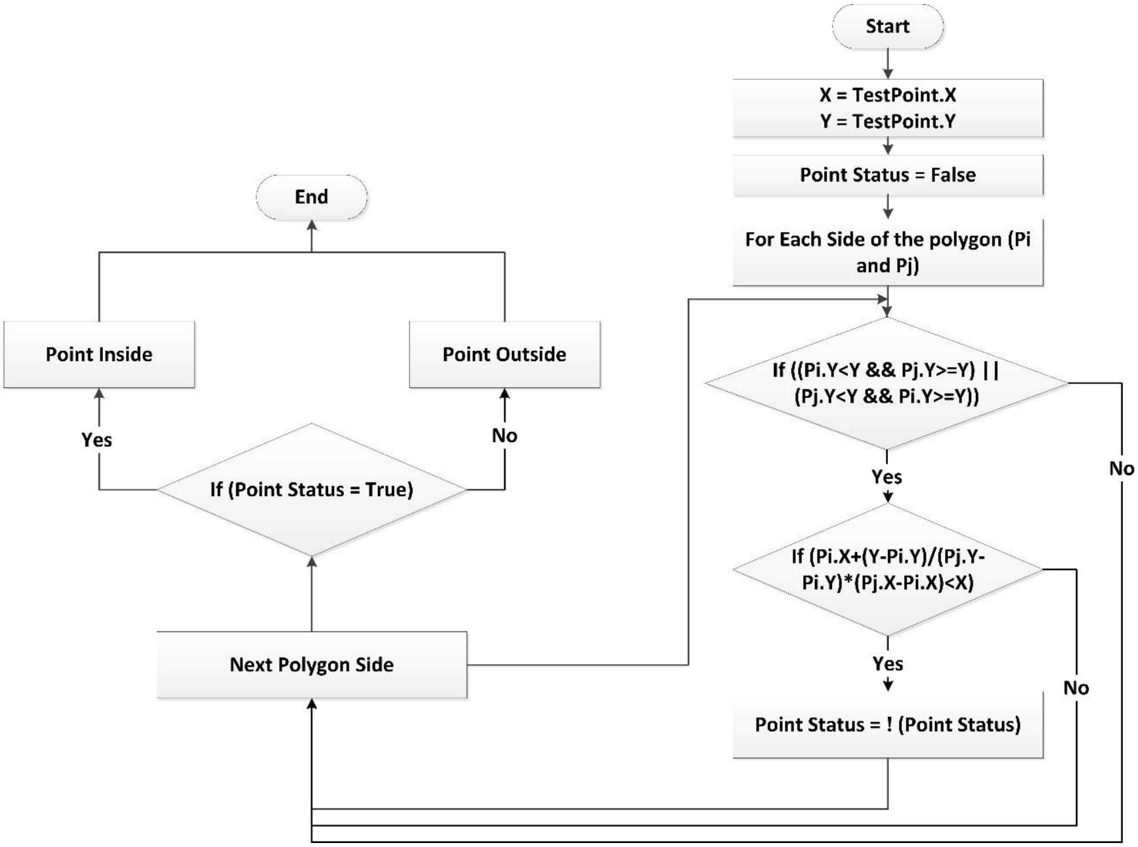

Start
X = TestPoint.X
Y = TestPoint.Y
Point Status = False
For Each Side of the polygon (Pi and Pj)
If ((Pi.Y<Y && Pj.Y>=Y) || (Pj.Y<Y && Pi.Y>=Y))
Yes
No
If (Pi.X+(Y-Pi.Y)/(Pj.Y-Pi.Y)*(Pj.X-Pi.X)<X)
Yes
No
Point Status = ! (Point Status)
Next Polygon Side
If (Point Status = True)
Yes
No
Point Inside
Point Outside
End

Figure 4-32: Point in Polygon Algorithm Flowchart

If the boundaries of the cut and fill locations are dynamically changing as the case in highway construction, fuzzy C-mean clustering algorithm is utilized to cluster GPS data to identify locations where loading equipment and spreading equipment spend more time, which in-turn dynamically identifies the cut and fill locations. In case of malfunctioning sensor on loading equipment, truck data are also clustered with respect to their location and speed to extract the correct locations. A sample of GPS data for a dump truck was clustered using FCM algorithm and two clusters were identified as shown in Figure 4-33.

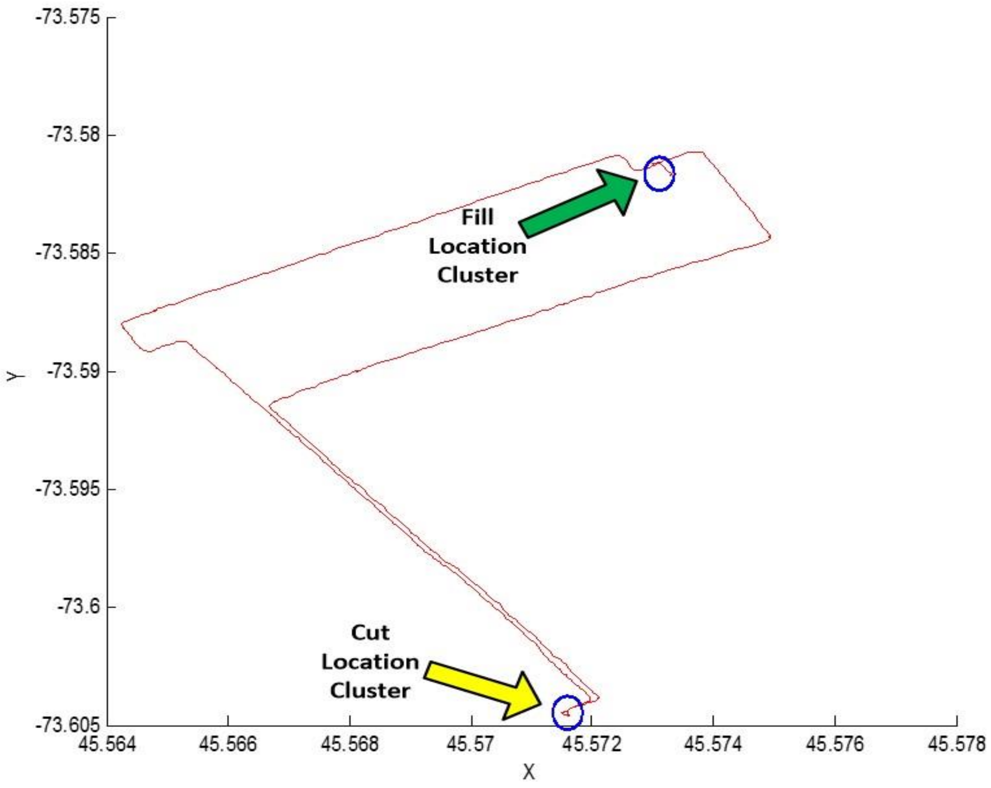


Figure 4-33: FCM Localization Algorithm

4.7.2 Proximity Identification Algorithm

Equipment proximity identification algorithm is responsible to discover equipment and reference areas such as entrances and exits of jobsites. Continues Bluetooth radio scanning is utilized for the proximity detection as shown in Figure 4-34. The discovered equipment type is identified by cross matching their media access control (MAC) address against a pre-defined list stored on the microcontroller's SD memory card. The pre-defined list is generated by the planning algorithm from the planned resources. The maximum proximity detection range is 100m, however, it can be adjusted using software parameters, which in turn control the power of RF transmitter on seven levels from - 20 to 12 dBm. This range is adjusted based on the size of the jobsite; a small size will require a lower setting for proximity detection.

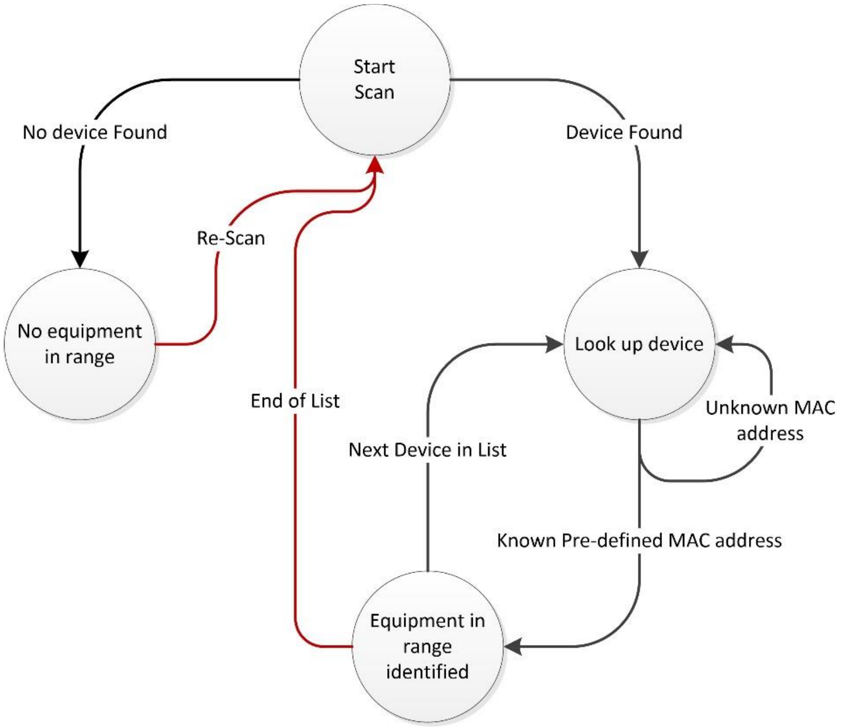


Figure 4-34: Equipment Proximity State Diagram

4.7.3 Hauling Volume Algorithm

Hauling volume calculation algorithm utilizes the methodology presented by (Yang et al., 2008), which estimates truck's payload weight from readings of four strain gages mounted on truck's suspension leaf springs. This feature enables improved progress tracking accuracy using the estimated quantities of soil excavated, also it enables alarming equipment's operator for any overloading conditions, and hence protects the contractor from possible extra costs for fuel, tires and mechanical failures. The weight at each suspension is calculated based on voltage signals of strain gages. The total payload of the vehicle is obtained by summing load readings in all suspensions; but resultant error can be very high. To reduce that error, a Kalman filter is used to account for nonlinearity in measurement. Once the payload is measured the volume of the payload can be estimated based on the excavated soil properties. The truck suspensions are denoted as Left-Front (LF), Left-Rear (LR), Right-Front (RF), and Right-Rear (RR), respectively. In Figure 4-35, the cross mark denotes the center of gravity of the payload with a value represented by W.

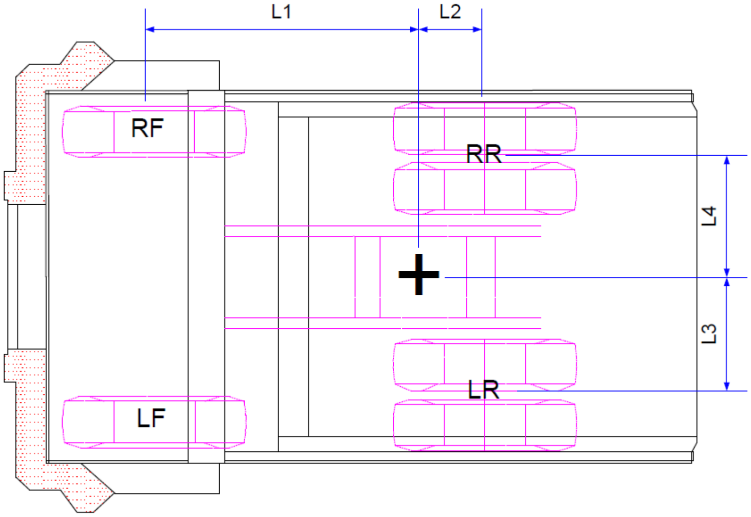


Figure 4-35: Schematics Diagram for Truck Payload

The measured strain gauge voltage is converted to weight, and hence the load applied to each suspension is calculated, then the gross payload W is calculated using Eq. (4-3).

$$P_{LF} + P_{RF} + P_{LR} + P_{RR} = W \qquad (4\text{-}3)$$

Where: W is the gross payload weight in Kg, P_{LF} , P_{RF}, P_{LR}, P_{RR} are the measured weight in Kg at the left front, right front, left rear and right rear truck suspensions respectively.

The Payload volume is calculated from the measured load weight using Eq. (4-4).

$$V = \frac{(W_{gross} - W_{truck\,bed})}{\rho_{soil}} \times Swell\,factor \tag{4-4}$$

Where:

W_{gross} is the gross payload weight in Kg.,

$W_{truck\ bed}$ is the weight of the truck bed in Kg., and

ρ_{soil} is the soil density in Kg/m3.

After calculating the load volume, a Kalman filter is applied to the measurement to reduce the error in the measurement. The Kalman filter is modeled as following:

$$\hat{V}_k = K_k Z_k + (1 - K_k)\hat{V}_{k-1} \tag{4-5}$$

Where:

$\hat{V}_k$is the current volume measurement,

K_k is Kalman gain, and

$\hat{V}_{k-1}$is the previous volume measurement.

4.7.4 Data Fusion Algorithm

Accurate identification of equipment modes of operations is crucial for realistic productivity estimates of ongoing earthmoving operations. Observing trends and patterns in the sensor data enables better identification of equipment modes of operations. However, these data are collected from multiple sources, which are heterogeneous and diverse in nature, content and format. Therefore, data fusion algorithms are required to stitch these data together and extract information pertinent to equipment modes of operation, and hence enable better estimation of activities start, finish, durations and resource utilization.

Typical earthmoving hauling equipment (dump truck) has seven modes of operation: queuing for loading, loading, travelling, queuing for dumping, dumping, returning and out-of-service. These modes are repeated regularly by a truck in earthmoving operation. The developed data fusion algorithm provides joint assessment of the data captured by the seven sensors integrated in the SA-GPS prototype as shown in Figure 4-36 (Ibrahim & Moselhi, 2014b). This is carried out by the 14 if-then rules shown in Figure 4-37 and listed in Appendix G. The sensor raw data is aggregated, filtered and transformed into logical levels. The logical representation of collected sensors data is passed onto the reasoning engine which performs logical reasoning against the pre-set modes of operations.

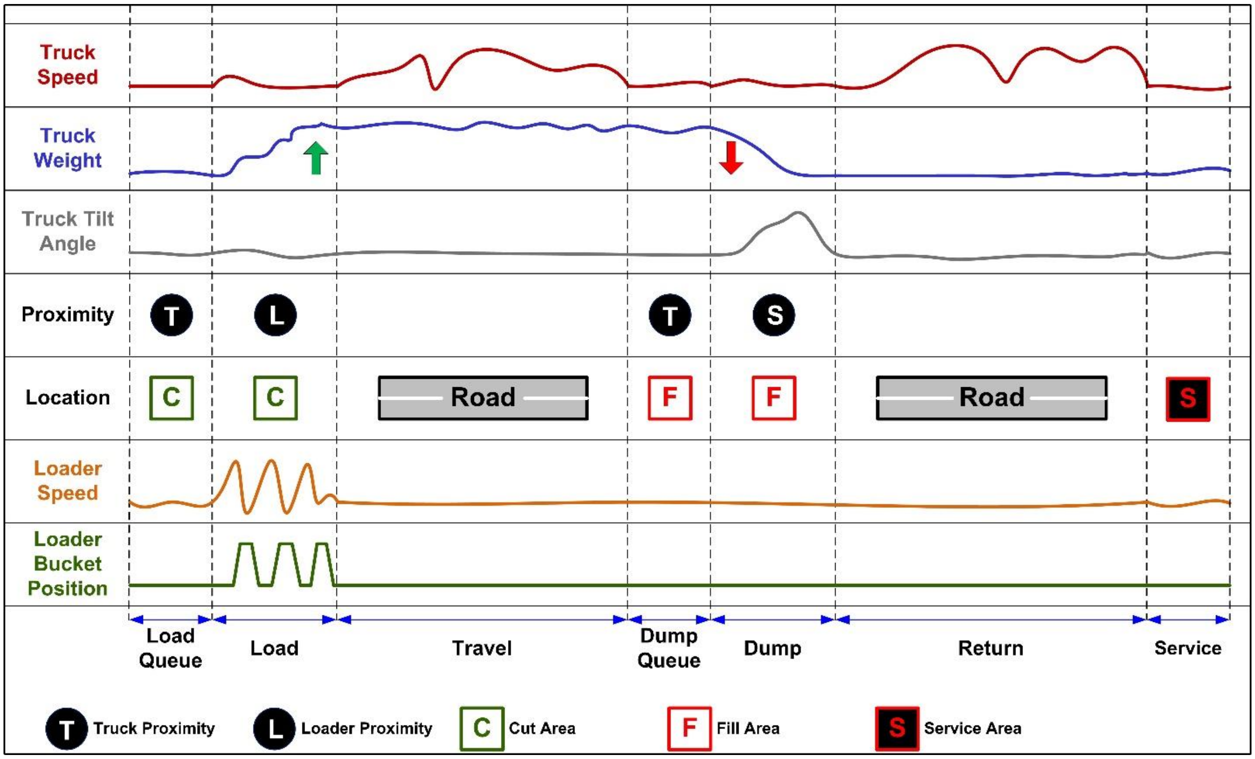


Figure 4-36: Hauling Truck Modes of Operations (Ibrahim & Moselhi, 2014b)

For example, a dump truck is identified as being loaded if it is in the cut area, its speed is below zero speed, it is in proximity to a loader, the loader is busy by detecting its boom angle and bucket height, and its load weight is increasing. Similarly, if the truck is travelling with speed higher than zero speed and its load weight is near maximum, then it can be identified as hauling task.

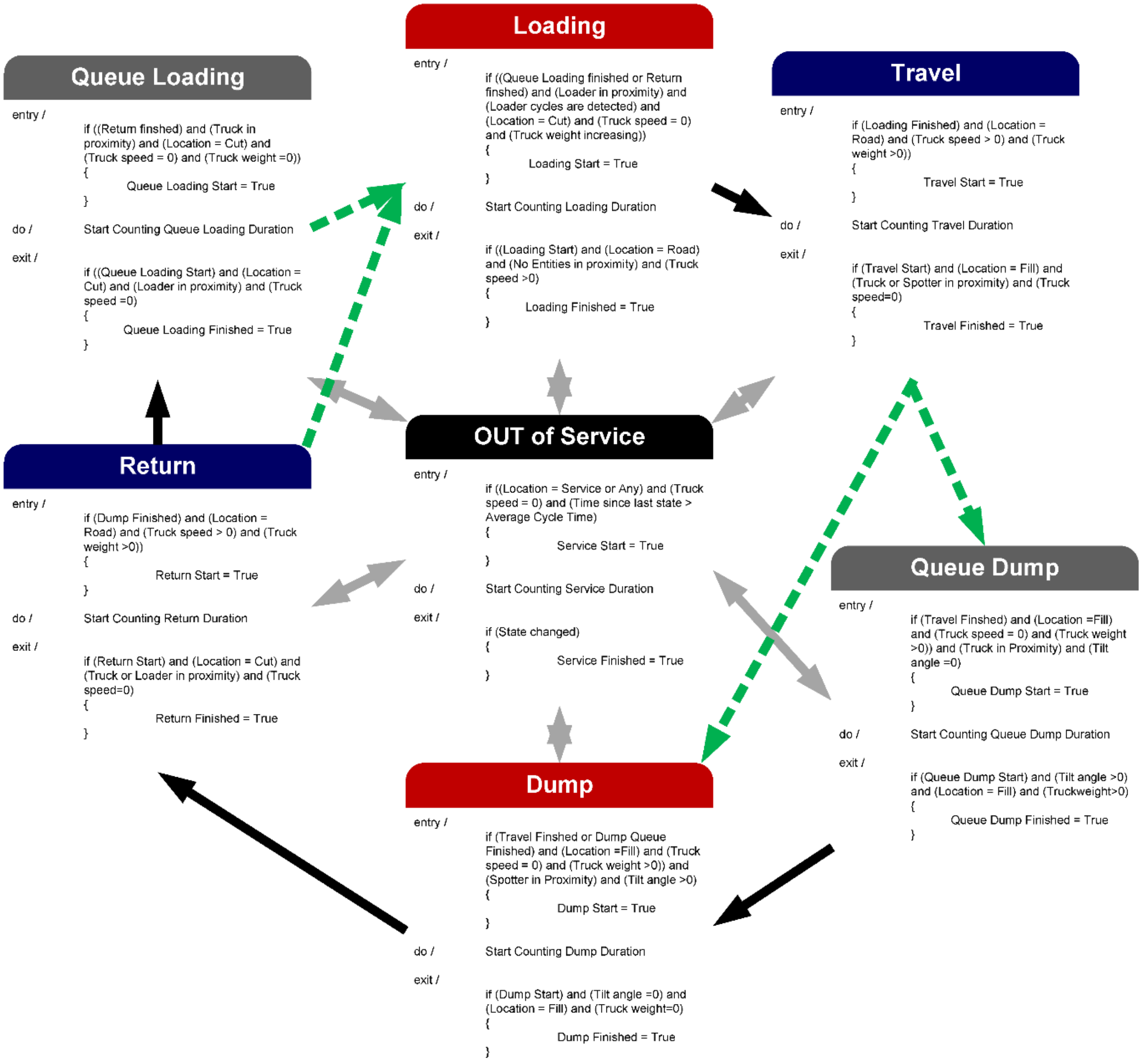

Queue Loading
entry /
if ((Return finshed) and (Truck in proximity) and (Location = Cut) and (Truck speed = 0) and (Truck weight =0))
{
Queue Loading Start = True
}
do /
Start Counting Queue Loading Duration
exit /
if ((Queue Loading Start) and (Location = Cut) and (Loader in proximity) and (Truck speed =0)
{
Queue Loading Finished = True
}
Loading
entry /
if ((Queue Loading finished or Return finshed) and (Loader in proximity) and (Loader cycles are detected) and (Location = Cut) and (Truck speed = 0) and (Truck weight increasing))
{
Loading Start = True
}
do /
Start Counting Loading Duration
exit /
if ((Loading Start) and (Location = Road) and (No Entities in proximity) and (Truck speed >0)
{
Loading Finished = True
}
Travel
entry /
if (Loading Finished) and (Location = Road) and (Truck speed > 0) and (Truck weight >0))
{
Travel Start = True
}
do /
Start Counting Travel Duration
exit /
if (Travel Start) and (Location = Fill) and (Truck or Spotter in proximity) and (Truck speed=0)
{
Travel Finished = True
}
Return
entry /
if (Dump Finished) and (Location = Road) and (Truck speed > 0) and (Truck weight >0))
{
Return Start = True
}
do /
Start Counting Return Duration
exit /
if (Return Start) and (Location = Cut) and (Truck or Loader in proximity) and (Truck speed=0)
{
Return Finished = True
}
OUT of Service
entry /
if ((Location = Service or Any) and (Truck speed = 0) and (Time since last state > Average Cycle Time)
{
Service Start = True
}
do /
Start Counting Service Duration
exit /
if (State changed)
{
Service Finished = True
}
Queue Dump
entry /
if (Travel Finshed) and (Location =Fill) and (Truck speed = 0) and (Truck weight >0)) and (Truck in Proximity) and (Tilt angle =0)
{
Queue Dump Start = True
}
do /
Start Counting Queue Dump Duration
exit /
if (Queue Dump Start) and (Tilt angle >0) and (Location = Fill) and (Truckweight>0)
{
Queue Dump Finished = True
}
Dump
entry /
if (Travel Finshed or Dump Queue Finished) and (Location =Fill) and (Truck speed = 0) and (Truck weight >0)) and (Spotter in Proximity) and (Tilt angle >0)
{
Dump Start = True
}
do /
Start Counting Dump Duration
exit /
if (Dump Start) and (Tilt angle =0) and (Location = Fill) and (Truck weight=0)
{
Dump Finished = True
}

Figure 4-37: Hauling Truck's Modes of Operation Reasoning Engine

Such algorithm does not require pre-existing large-scale dataset, however, its main drawback is its inability to represent uncertainty, but this can be alleviated by integrating fuzzy logics into the logical approaches. Table 4-6 lists the seven modes of operation and the sensors readings during these modes.

Table 4-6: Haul Truck Modes of Operation

Mode of Operation	Location	Speed	Load Weight	Proximity	Tilt Angle	Previous State
Load Queue	Loading Area	= 0	≈ 0	Trucks	= 0	Return
Load	Loading Area	= 0	++	Loader	= 0	Return or Q. Loading
Travel	Road	> 0	> 0	Non	= 0	Loading
Dump Queue	Dump Area	= 0	> 0	Trucks	= 0	Travel
Dump	Dump Area	= 0	--	Spotter	> 0	Travel or Q. Dump
Return	Road	> 0	≈ 0	Non	= 0	Dump
Service	Service Area	= 0	Any	Any	= 0	Any

An effort was made to enhance the robustness of the developed prototype and to provide fault tolerance, a fuzzy rule based reasoning engine was developed in MATLAB and converted to C++ to be loaded on the microcontroller. The developed fuzzy reasoning algorithm consists of the following steps:

- Fuzzification on-sensor node of both the sensor raw values and their differential variations using membership functions.
- For simplicity triangle membership functions were utilized.
- The fuzzified values are broadcasted to the gateway node.
- The gateway node merges the single nodes fuzzy values into a multi node fuzzy rules. These rules have the following structure:

IF (T_i=Loading) or (L_j=Busy Loading T_i) Then (T_i = Loading)

Where: T_i is Truck number i, and L_j is Loader number j.

Figure 4-38 illustrated the fuzzy reasoning engine for a hauling truck node. Such distributed reasoning scheme increase the degree of reliability and robustness of the developed prototype in case of a failure in one of the sensors or even the whole node.

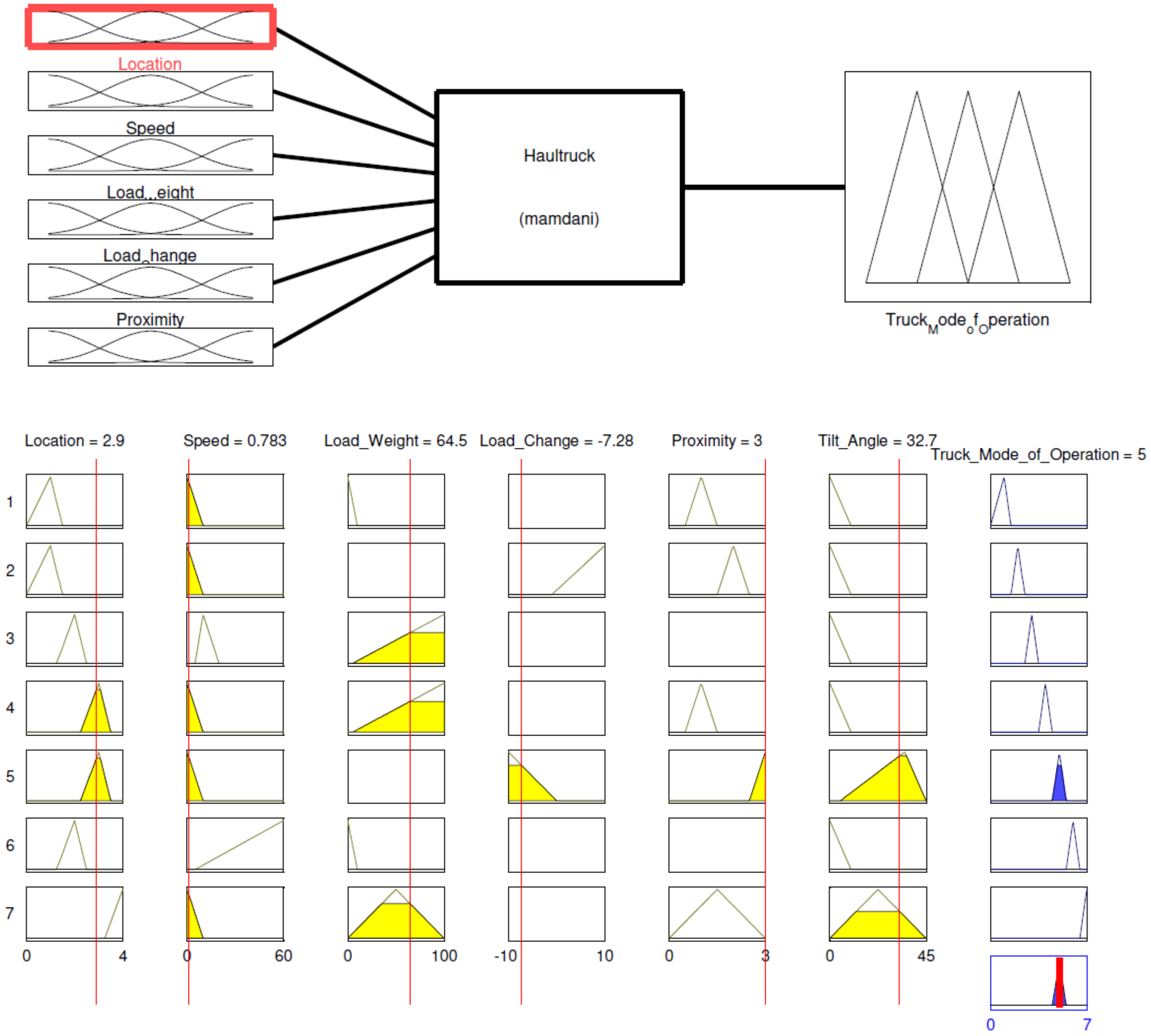


Figure 4-38: Hauling Truck Fuzzy Reasoning Engine

4.8 Framework Implementation

To link the developed SA-GPS with the automated framework described in chapter 3, three algorithms were developed: planning algorithm, progress estimation and forecasting algorithm, and productivity analysis algorithm as shown in Figure 4-39 (Ibrahim & Moselhi, 2014a). Their respective input and output is briefly summarized in Table 4-7, detailed description of these algorithms is presented subsequently.

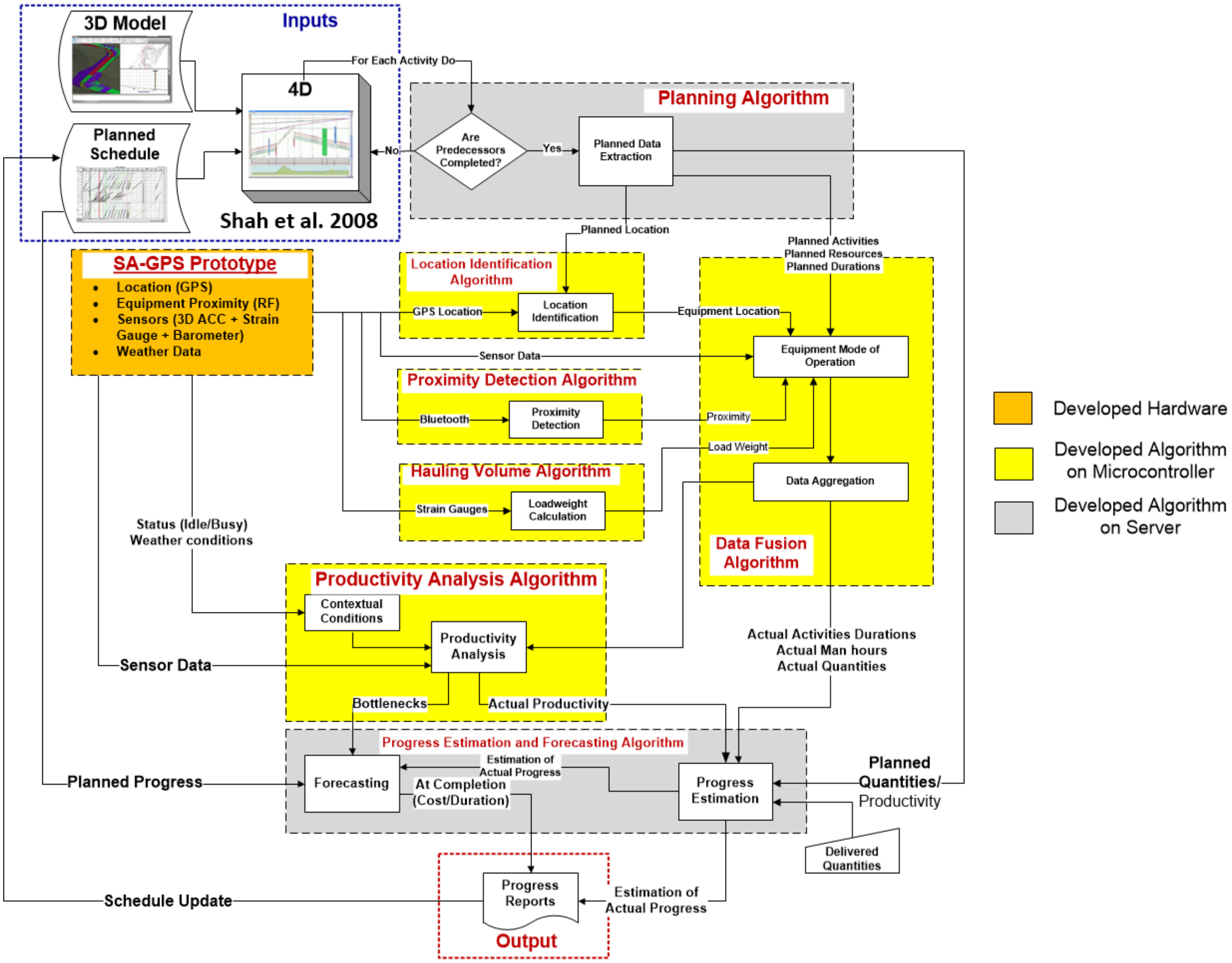


Figure 4-39: SA-GPS Software/Algorithms

Table 4-7: Developed Software Algorithms (Input / Output)

Algorithm	Input	Output
Planning Algorithm	4D Model	Planned Activities Planned Locations Planned Resources Planned Productivity Planned Quantities

Productivity Analysis	Equipment Modes of Operations Actual Durations Actual Excavated Soil Quantities Weather conditions	Actual Productivity Bottlenecks Aggressive Operator Flags Weather Impact Flags
Progress Estimation and Forecasting	Current Undergoing Activity Planned Quantities Actual Productivity Actual Durations	Real-Time Progress Reports

4.8.1 Planning Algorithm

The purpose of this algorithm is to extract input data from the 4-D model and save it in the framework database as outlined in chapter 3. This 4-D model is developed by integrating resource-loaded project schedule and the 3D terrain model of the earthmoving project (Shah et al., 2008). With these inputs, project's resources and their interaction onsite are effectively tracked with respect to their locations. The 4-D model houses necessary inputs for effective project control such as planned activities (start dates, finish dates, durations, and job logic), their physical locations, their planned resources (materials, labor, equipment) and planned quantities (cut, fill).

The planning algorithm depicted in Figure 4-40 identifies pending activities based on scheduled dates and job logic. Then extracts from the 4-D model the assigned resources for each pending activity, its planned work zone (location), and its planned quantities of cut and/or fill. The work zone is defined by a set of points, which serves as boundaries for equipment tracking.

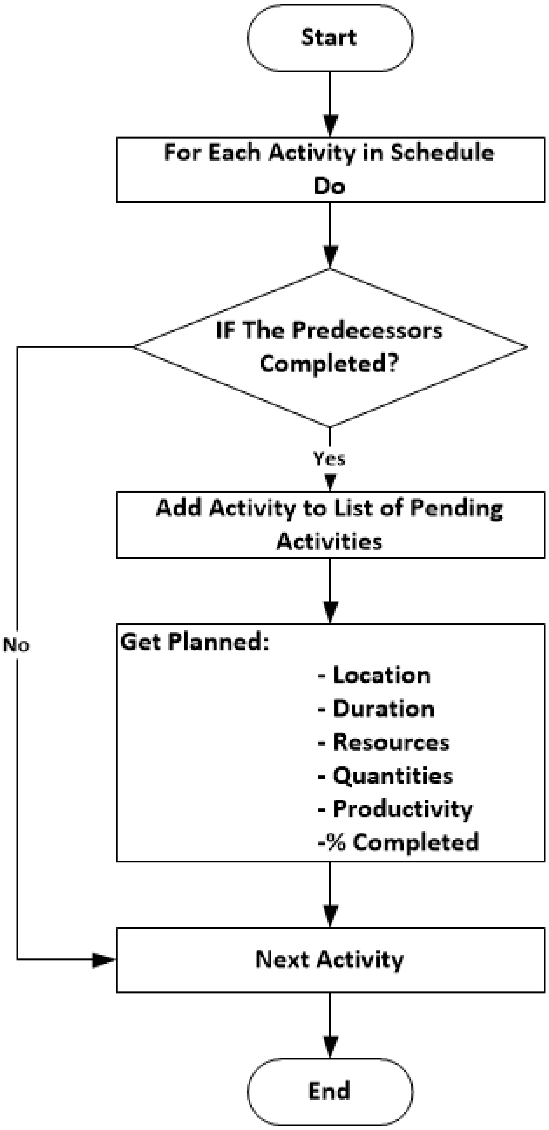


Figure 4-40: Planning Algorithms Flowchart

4.8.2 Productivity Analysis Algorithm

This Algorithm is responsible for calculating actual productivity of the earthmoving operation being considered and analyzing the productivity with respect to the contextual data to identify potential bottlenecks in the site operations. Productivity calculation is based on actual measured cycle times and actual estimates of excavated soil volumes, to calculate productivity per equipment. Overall productivity is calculated by summing the productivity of individual trucks in the fleet using Eq. (4-6).

$$Overall\ Productivity = \sum_{i=1}^{n} Truck\ Productivity_i \tag{4-6}$$

Where:

Truck Productivity$_i$ is truck i productivity in m^3/hr, and n is the number of trucks.

The truck productivity is calculated using Eq. (4-7).

$$Truck\ Productivity = \frac{Soil\ Volume\ (m^3) \times load\ factor}{Total\ Cycle\ time\ (hr)} \tag{4-7}$$

Where:

Total cycle time = sum of loading, travel, dumping, returning and service time

Load factor = factor for converting soil material to a compact state

The American Society for Testing and Materials (ASTM) standard practice for Job Productivity Measurement (ASTM E2691-09) provides a metric for measuring productivity differential. The Productivity differential can be calculated using Eq. (4-8):

$$Productivity\ Differential = \frac{(Average\ Productivity - Current\ Productivity)}{Average\ Productivity} \tag{4-8}$$

The data required to create a baseline for average productivity can be drawn from company's past practice or industry standards (ASTM E2691-09). JPM measures productivity changes, trends and anomalies and it can be considered as an early warning signal for construction productivity (ASTM E2691-09) problems. Five signals are proposed in the standard to represent anomalies and deviations from the reference point: Trends, shifts in the mean, extreme points, saw tooth pattern and missing data. For instance, if 6 or more consecutive points (productivity differentials) show an increasing or decreasing trend, the signal is representing "Trends" (ASTM E2691-09) as shown in Figure 4-41.

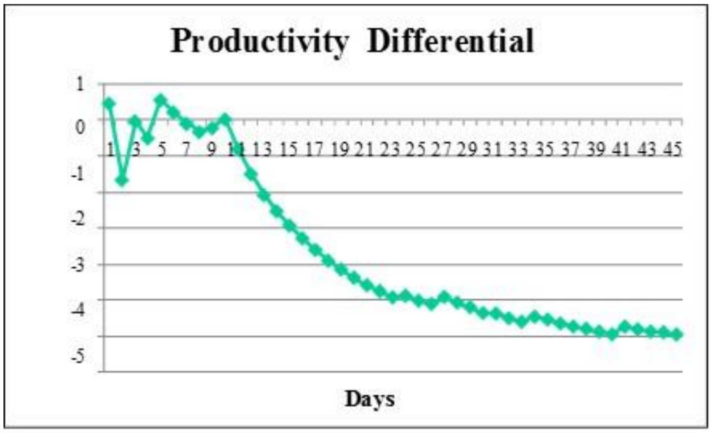


Figure 4-41: Example of Productivity Differential (Shahandashti et al., 2010)

The developed SA-GPS prototype is able to provide the necessary information for automated earthmoving productivity assessment for the different scenarios. Table 4-8 depicts 21 different scenario, where SA-GPS data are proposed to identify bottlenecks and problems in earthmoving productivity. The potential of using SA-GPS in automated productivity assessment is supported with the multisensory data capture technologies integrated in the hardware developed, as well as the data fusion algorithm described above.

Table 4-8: Scenarios for Automated Productivity Assessment using SA-GPS

NO	Scenario	Required information	Sensor data
1	Adverse loading site condition	• Increase in the average duration that a truck spends in the earthmoving site • Exclude the waiting times and loading times	• Proximity Detection (Bluetooth) • Truck Speed (GPS) • Location (GPS)
2	Adverse access road condition	• Increase in the average duration that a truck spends on access road	• Proximity Detection (Bluetooth) • Truck Speed (GPS) • Location (GPS)
3	Changing soil type	• Soil percent swell • Soil Density • Loading time	• Load Weight (Strain gauge) • Number of Buckets (Altimeter) • Loading Time (Bluetooth+GPS)
4	Excavator breakdown	• Increase in the average hauler queue time in loading • Excavator in service area	• Proximity Detection (Bluetooth) • Truck Speed (GPS)

			• Location (GPS) • Load Weight (Strain gauge) • Number of Buckets (Altimeter)
5	Depth of cut is changing	• Change in the number of passes to fill the bucket • Change in the number of passes to fill the truck • Change in the average loading time	• Proximity Detection (Bluetooth) • Truck Speed (GPS) • Location (GPS) • Load Weight (Strain gauge) • Number of Buckets (Altimeter)
6	Over time	• Labor hours related to workers who work overtime • Labor hours related to workers who do not work overtime	• Proximity Detection (Bluetooth) • Time (GPS)
7	Day and night shift	• Day-shift expended labor hours • Night-shift expended labor hours • Day-shift percent complete • Night-shift percent complete	• Proximity Detection (Bluetooth) • Time (GPS)
8	One-way hauling road	• Duration that a hauler waits to enter the hauling road	• Proximity Detection (Bluetooth) • Truck Speed (GPS) • Location (GPS)
9	Heavy traffic in hauling road	• Increase in the average duration that a hauler spends in hauling road	• Hauling Time (Bluetooth+GPS) • Truck Speed (GPS) • Truck Engine Idle (Accelerometer) • Location (GPS)
10	Changing dumping location	• Hauling distance	• Distance (GPS) • Hauling Time (Bluetooth+GPS) • Truck Speed (GPS) • Location (GPS)
11	Adverse weather conditions	• Temperature • Humidity • Precipitation	• Weather measurement station
12	Truck breakdown	• Increase in the average loading unit idle time • Truck in service area	• Loader Idle Time (Bluetooth, GPS, Altimeter) • Location (GPS) • Location + Speed + Load weight
13	Adverse dump site condition	• Increase in the average duration that a truck spends in the dump site • Exclude the actual dumping times	• Proximity Detection (Bluetooth) • Dump Time (Accelerometer) • Truck Speed (GPS) • Location (GPS)
14	Excessive Excavator Idle	• Increase in the average loading unit idle time	• Loader Idle Time (Bluetooth, GPS, Altimeter)
15	Excessive Hauler	• Increase in the average hauler unit idle time	• Proximity Detection (Bluetooth)

	Idle		• Truck Speed (GPS) • Location (GPS) • Load Weight (Strain gauge) • Number of Buckets (Altimeter)
16	Aggressive Hauler operator	• Harsh braking • Harsh acceleration	• Acceleration and Swing (Accelerometer)
17	Hauler over loading	• Payload exceed maximum payload	• Load Weight (Strain gauge)
18	Bucket Not Full	• Increased number of buckets to load	• Load Weight (Strain gauge) • Number of Buckets (Altimeter)
19	Low Dump Angle	• Increase in the average dump time	• Dump Time (Accelerometer) • Truck bed tilt angle (Accelerometer)
20	Sticky material	• Increase in average load weight while truck returning • Decreased number of buckets to load	• Load Weight (Strain gauge) • Number of Buckets (Altimeter)
21	Excavator repositions	• Time between excavator movements	• Location (GPS) • Arm Swing (Accelerometer)

Equipment operators are usually under a great deal of pressure to achieve target production rates. While the operator may push the equipment to the maximum to achieve the requested production, it is beneficial to the contractor to monitor the operator behavior for equipment abuse. Speeding is a huge factor in high fuel consumption. The developed SA-GPS prototype allows monitoring of the operator and equipment to flag and report any undesirable behavior. Alerts are triggered for excessive speeding, harsh breaking and excessive idling. A three axial accelerometer is used for detecting aggressive driver behavior such as sudden acceleration and breaking. Measured acceleration in x-axis and y-axis reflects the driver’s direct control of the vehicle while accelerating or applying the brakes. The margin for safe acceleration and brake is ±0.3g, while harsh acceleration and brake can reach ±0.5g (Fazeen et al., 2012). The algorithm shown in Figure 4-42 utilizes these limits to flag aggressive driving behavior.

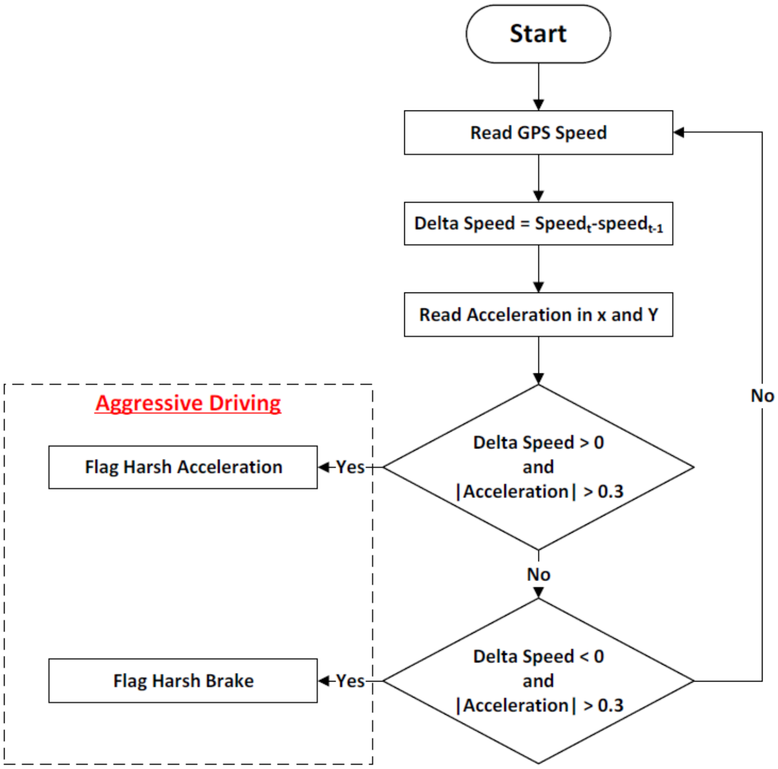


Figure 4-42: SA-GPS Operator Behavior Detection Flowchart

4.8.3 Progress Estimation and Forecasting Algorithm

This algorithm is responsible for estimating and forecasting project's progress and generating automated progress reports. The percentage completed of each activity is calculated based on the measured actual quantities of work completed on each activity. Then the percentage completed is used to calculate the budgeted cost of work performed (BCWP), which is rolled up to the project level. The Earned value analysis is performed to calculate both cost performance index and schedule performance index. The progress is forecasted based on the self-adaptive forecasting method of (Ibrahim & Moselhi, 2013b), which enhances forecasting accuracy through continues iteration to reduce the forecasting error. This method is based on the principles of iterated multi-step forecasting method. The proposed method forecasts the final cost of a project according to the following steps:

At time period t, the forecasting factor (δt) is calculated from the actual productivity, and the planned productivity,

$$\delta_{\mathrm{t}} = \frac{Actual\ Productivity_t}{Planned\ Productivity_t}, \tag{4-9}$$

Then, the estimated productivity for the next period t+1 is forecasted using the previously calculated forecasting factor at time period t.

$$\text{Estimated Productivity}_{\mathrm{t+1}} = \delta_{\mathrm{t}} \times \text{Planned Productivity}_{\mathrm{t+1}}, \tag{4-10}$$

At time period t+1, the forecasting factor ($\delta_{\mathrm{t+1}}$) is calculated,

$$\delta_{\mathrm{t+1}} = \frac{\text{Actual Productivity}_{\mathrm{t+1}}}{\text{Planned Productivity}_{\mathrm{t+1}}}, \tag{4-11}$$

The forecast error correction factor ($\varepsilon_{\mathrm{t+1}}$) is calculated by dividing the estimated productivity by the actual productivity,

$$\varepsilon_{\mathrm{t+1}} = \frac{\text{Actual Productivity}_{\mathrm{t+1}}}{\text{Estimated Productivity}_{\mathrm{t+1}}}, \tag{4-12}$$

Then, the estimated productivity for the next period t+2 is forecasted based on the forecasting factor and the forecast error correction factor,

$$\text{Estimated Productivity}_{\mathrm{t+2}} = (\delta_{\mathrm{t+1}} \times \varepsilon_{\mathrm{t+1}}) \times \text{Planned Productivity}_{\mathrm{t+2}}, \tag{4-13}$$

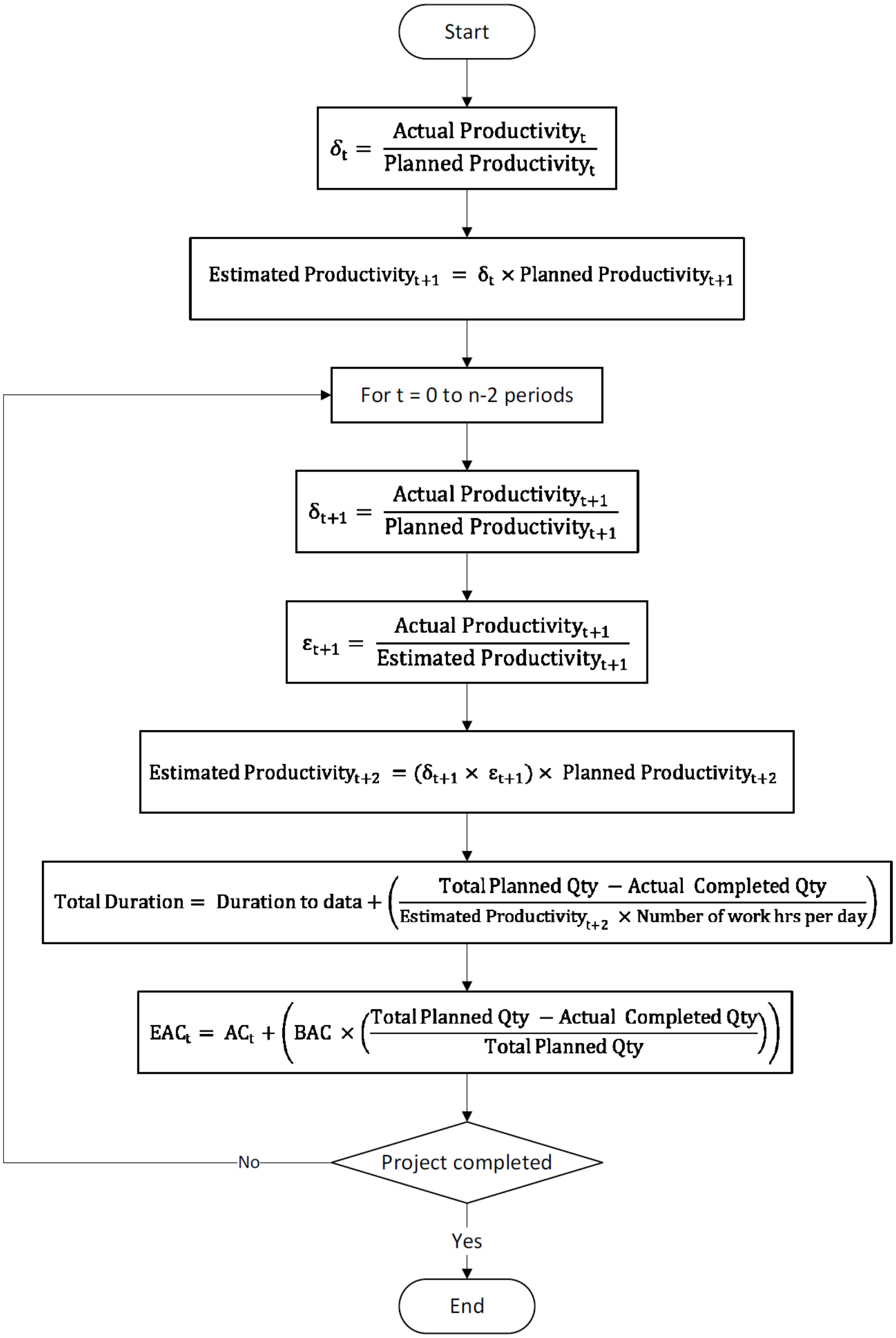
The total project duration is forecasted based on estimated productivity,

$$\text{Total Duration} = \text{Duration to data} + \left(\frac{\text{Total Planned Qty} - \text{Actual Completed Qty}}{\text{Estimated Productivity}_{\mathrm{t+2}} \times \text{Number of work hrs per day}}\right) \tag{4-14}$$

The total project cost ($\mathrm{EAC_t}$) is forecasted from the total project budget (BAC),

$$\mathrm{EAC_t} = \mathrm{AC_t} + \left(\mathrm{BAC} \times \left(\frac{\text{Total Planned Qty} - \text{Actual Completed Qty}}{\text{Total Planned Qty}}\right)\right) \tag{4-15}$$

The above steps are repeated for the following time periods as shown in Figure 4-43.



Start
$\delta_t = \frac{\text{Actual Productivity}_t}{\text{Planned Productivity}_t}$
$\text{Estimated Productivity}_{t+1} = \delta_t \times \text{Planned Productivity}_{t+1}$
For t = 0 to n-2 periods
$\delta_{t+1} = \frac{\text{Actual Productivity}_{t+1}}{\text{Planned Productivity}_{t+1}}$
$\varepsilon_{t+1} = \frac{\text{Actual Productivity}_{t+1}}{\text{Estimated Productivity}_{t+1}}$
$\text{Estimated Productivity}_{t+2} = (\delta_{t+1} \times \varepsilon_{t+1}) \times \text{Planned Productivity}_{t+2}$
$\text{Total Duration} = \text{Duration to data} + \left(\frac{\text{Total Planned Qty} - \text{Actual Completed Qty}}{\text{Estimated Productivity}_{t+2} \times \text{Number of work hrs per day}}\right)$
$\text{EAC}_t = \text{AC}_t + \left(\text{BAC} \times \left(\frac{\text{Total Planned Qty} - \text{Actual Completed Qty}}{\text{Total Planned Qty}}\right)\right)$
Project completed
No
Yes
End

Figure 4-43: Self-Adaptive Forecasting Algorithm (Ibrahim & Moselhi, 2013b)

4.9 Prototype Laboratory Validation

Laboratory experiments were conducted to test the developed SA-GPS prototype latency and energy consumption.

4.9.1 Prototype Latency Test

Latency is the measure of time delay between the sending and receiving of a message. Data latency is very critical especially for real time tracking and control applications. Therefore it was necessary to measure the latency of the developed prototype. The test-bed consisted of one gateway and five SA-GPS mobile nodes as shown in Figure 4-44.

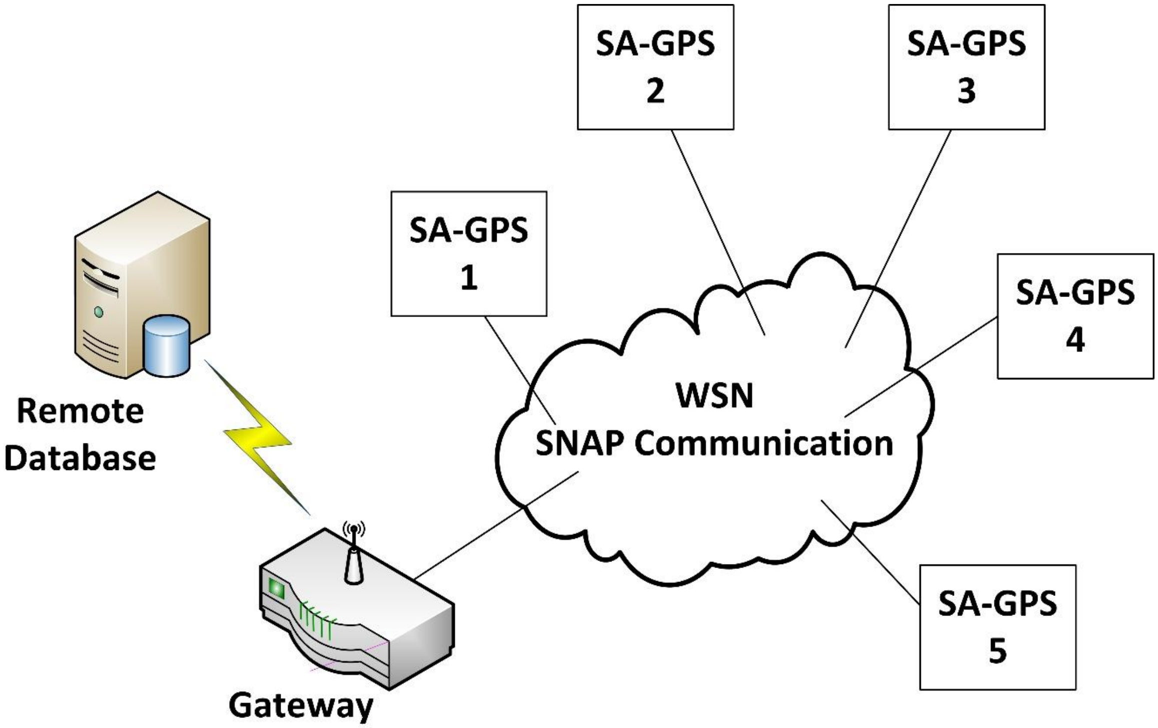


Figure 4-44: Latency Test-bed

The gateway is used for initiating the latency measurements, retrieving the measured times and storing these locally, then send these data over WLAN to server based database for processing. The gateway performs a double task, first, receiving and sending packets from/to the mobile node and second, communicating data through WLAN. Latency is measured in terms of the round trip time (RTT) measured from the start of data transmission from the gateway node, traveling to a destination node where it is then echoed back and returned to the gateway as shown in Figure 4-45. The RTT is measured by adding a start-time timestamp in the data packet when the packet is leaving the gateway, and later adding an end-time timestamp and a delta-time indicator (delta-time = end-time − start-time) in the same data packet when it returns back to the gateway. This measurement identify the in-network latency. The data size was varied from 1 to 8000 bytes. The 8kb size was validated by experiments to be the largest size of data to be transferred from the mobile unit to the gateway at any time. Each test was repeated for 100 runs then results were averaged and summarized in Table 4-9.

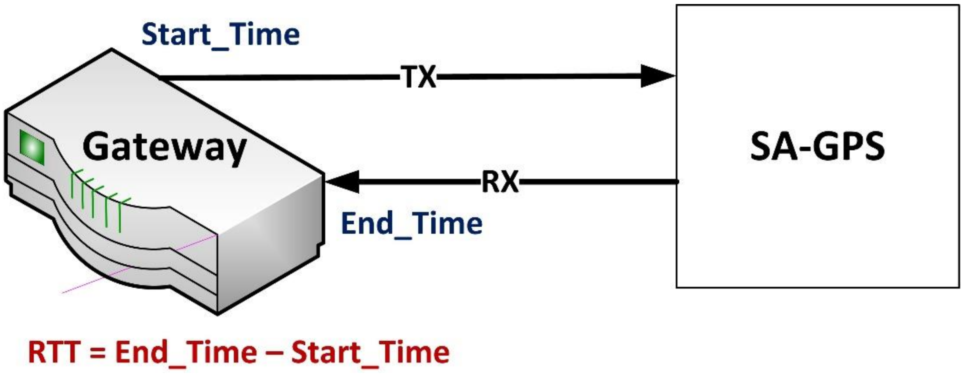


Figure 4-45: RTT Measurement

Table 4-9: SA-GPS Latency Test Results

Data Bytes	Average Latency (ms)	Maximum Latency (ms)
1	4.05	15.60
2	4.07	16.30
12	4.09	18.70
30	4.67	21.10
62	8.19	24.20
71	8.64	32.40
128	13.63	46.80
500	45.05	56.80
1000	87.72	93.60
2000	172.07	187.20
4000	344.04	358.80
8000	683.99	686.40

4.9.2 Prototype Energy Consumption Test

Measuring the energy consumption of the developed SA-GPS prototype was very important to validate the possibility to power the prototype from backup battery in case of failure in the external power supply. The energy consumption was measured as j/hr to accomplish the complete performance of the developed prototype. The measurement were performed in the lab over 8 hours period, where the values obtained averaged over that period. The energy consumption was measured per each application component: Data Sensing, Data Recording, Data Processing, and Data Transmission. The results summary presented in Table 4-10 indicated that, data transmission is the highest energy consuming application component with 38.40% of the total consumption. Data sensing came in the second place with 28.61%. The total power for the prototype can be supplied from 2xAA batteries for almost 8.4 hours. Keeping the data transmission to minimum is essential to improve the developed prototype energy performance.

Table 4-10: SA-GPS Energy Consumption Test Results

Application Component	**Energy Usage (j/hr)**	**Energy Consumption Percentage**
Data Sensing	752.33	28.61%
Data Processing	630.00	23.96%
Data Recording	237.60	9.04%
Data Transmission	1,009.80	38.40%
Total	**2,629.73**	**100.00%**
Expected Lifetime on 2XAA (hours)	8.40	

4.10 Prototype Validation / Case Study

The developed prototype was tested on a scaled case study, where remotely controlled scaled equipment (1:50) were used for a dirt road construction project as shown in Figure 4-46. The purpose of this case study was to validate results obtained in laboratory experiments and measure the prototype accuracy. In this project, several scenarios were simulated to test the performance of the developed prototype in estimating the project progress and in identifying productivity bottlenecks.

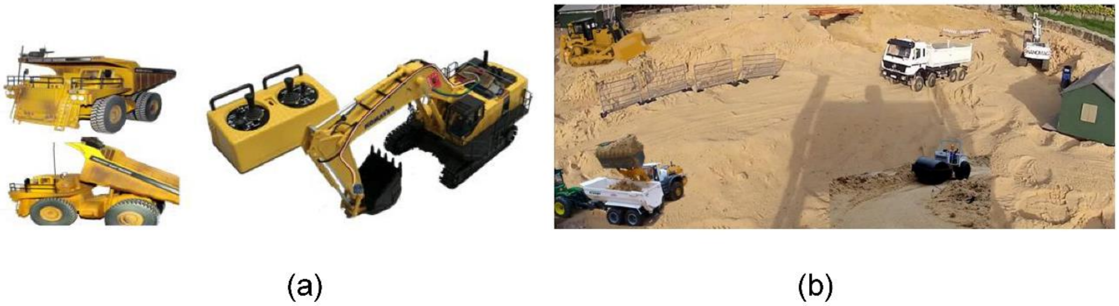
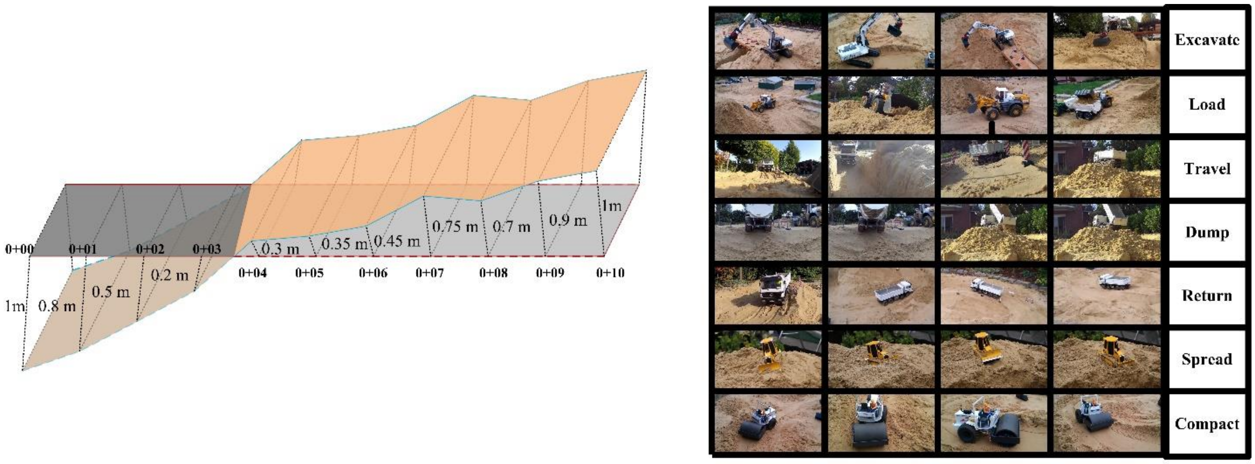


Figure 4-46: SA-GPS Prototype Mounted on Scaled Equipment

In this case study, a simple 10 m length and 1 m wide dirt road was constructed. The utilized construction equipment fleet composed of a dozer, excavator, wheeled loader, two trucks, and roller. Seven main activities were performed during the construction of this project, namely, excavate, load, haul, dump, return, spread and compact. A team of 8 personnel executed this project controlling the scaled equipment. The total quantity of the excavation was 2.9 m^3 and the total quantity of fill was 1.37 m^3. The project was completed in 15 hours with a total cost of $5382.15 (estimated based on hourly rate). The road design is shown in Figure 4-47, where the quantity of earthwork was calculated assuming typical trapezoidal sections at 1 m intervals along the road profile. A video camera was used to record the actual construction progress in order to compare it with the results obtained from the SA-GPS prototype.



(a) 3D Road Profile **(b) Video Recording of Construction**

Figure 4-47: Case Study Jobsite

The project plan was to cut 2.805 m3 of dry sand and back filling of 1.37 m3. However the actual total volume of soil in cut locations was 2.9 m3 due to the unstable sandy soil in the cut areas.

4.10.1 Results

In order to benchmark the accuracy of the developed prototype, the performance of developed prototype is compared to standalone GPS method developed by Montaser et al (2012). The project progress was also calculated manually with the aid of video camera recording and actual measurements of cut and fill quantities. Figure 4-48 illustrated the project progress using both methods. The results presented in Table 4-11 show that the standalone GPS method estimated the progress of earthwork with an average absolute percentage error of 12.26%, and the developed SA-GPS prototype estimated the project progress with an average absolute percentage error of 2.88%. The project duration was forecasted at T=9 h using both methods as shown in

Table 4-12. The forecasted project duration and cost using the standalone GPS method were 13.7 h and $4,915.70, respectively, with error of 8.66%. On the other hand, the forecasted project duration and cost using the developed framework where 15.7 h and $5,633.32, respectively, with error of 4.66%.

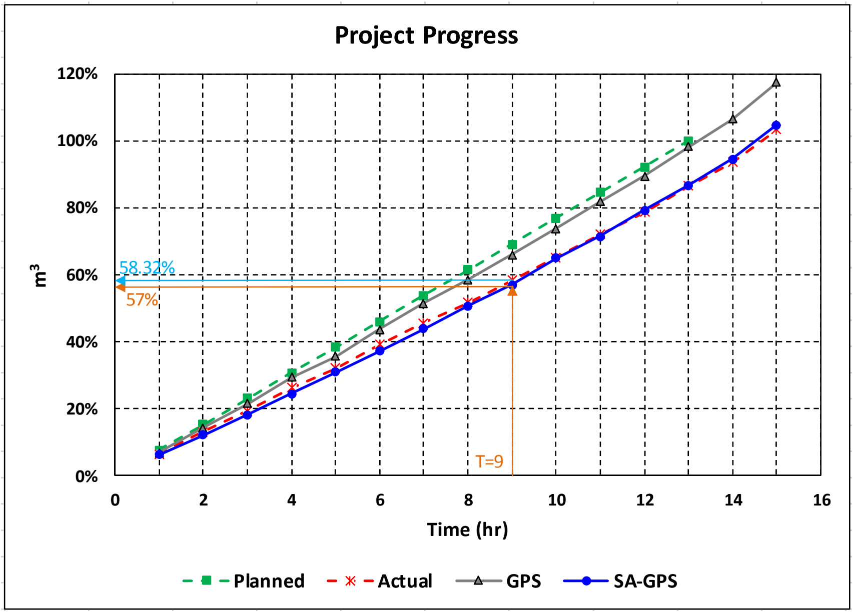


Figure 4-48: Project Progress Tracking

Table 4-11: Project Progress Comparison

Time (h)	Planned Qty m3	Actual Qty m3	Measurement Qty (m3)		Estimation Error		Absolute % Error	
			GPS	SA-GPS	GPS	SA-GPS	GPS	SA-GPS
1	0.216	0.182	0.197	0.178	8.31%	-2.41%	8.31%	2.41%
2	0.432	0.368	0.403	0.342	9.50%	-7.06%	9.50%	7.06%
3	0.647	0.544	0.606	0.511	11.30%	-6.17%	11.30%	6.17%
4	0.863	0.738	0.823	0.690	11.54%	-6.46%	11.54%	6.46%
5	1.079	0.904	1.003	0.872	10.92%	-3.57%	10.92%	3.57%
6	1.295	1.101	1.228	1.043	11.58%	-5.21%	11.58%	5.21%
7	1.511	1.281	1.44	1.231	12.68%	-3.92%	12.68%	3.92%

			4					
8	1.726	1.450	1.642	1.425	13.23%	-1.76%	13.23%	1.76%
9	1.942	1.636	1.850	1.599	13.06%	-2.27%	13.06%	2.27%
10	2.158	1.827	2.069	1.822	13.21%	-0.31%	13.21%	0.31%
11	2.374	2.020	2.299	2.008	13.84%	-0.56%	13.84%	0.56%
12	2.590	2.210	2.513	2.226	13.71%	0.71%	13.71%	0.71%
13	2.805	2.428	2.758	2.435	13.60%	0.32%	13.60%	0.32%
14		2.626	2.987	2.654	13.76%	1.07%	13.76%	1.07%
15		2.900	3.295	2.940	13.62%	1.38%	13.62%	1.38%
			Average		12.26%	-2.41%	12.26%	2.88%

Table 4-12: Project Estimate at Completion and Duration Forecasting

T = 9 h	GPS	SA-GPS	Planned	Actual
Progress to date (m3)	1.850	1.599	1.942	1.636
Percentage Complete	64.35%	57%	69.23%	58.32%
Forecasted Project duration (h)	13.7	15.7	13	15
Forecasted Cost	$4,915.70	$5,633.32	$4664.53	$5382.15
Forecast Error	-8.66%	4.66%		

The productivity estimation using both methods was also evaluated in this case study. Table 4-13 illustrated the results of the productivity comparison, where the average actual productivity measured using manual method was lower by 10.41% than the planned productivity as shown in Figure 4-49. The estimated average productivity using the GPS was higher than the average actual productivity by 13.62%, while the estimated average productivity using SA-GPS was only 1.38% higher than the average actual productivity. The productivity estimation using GPS had 13.65% mean absolute percentage error in comparison to 8.27% using SA-GPS prototype.

Table 4-13: Hourly Productivity Comparison

Time	Planned	Actual	Productivity	Error	Absolute %Error

(h)	Productivity	Productivity	GPS	SA-GPS	GPS	SA-GPS	GPS	SA-GPS
	m3/hr	m3/hr	m3/hr	m3/hr				
1	0.216	0.182	0.197	0.178	8.31%	-2.41%	8.31%	2.41%
2	0.216	0.186	0.206	0.164	10.66%	-11.61%	10.66%	11.61%
3	0.216	0.177	0.203	0.169	15.07%	-4.31%	15.07%	4.31%
4	0.216	0.194	0.217	0.180	12.20%	-7.27%	12.20%	7.27%
5	0.216	0.166	0.180	0.182	8.19%	9.23%	8.19%	9.23%
6	0.216	0.196	0.225	0.171	14.62%	-12.77%	14.62%	12.77%
7	0.216	0.181	0.216	0.188	19.37%	3.96%	19.37%	3.96%
8	0.216	0.169	0.198	0.194	17.37%	14.60%	17.37%	14.60%
9	0.216	0.186	0.208	0.175	11.77%	-6.22%	11.77%	6.22%
10	0.216	0.191	0.219	0.222	14.50%	16.44%	14.50%	16.44%
11	0.216	0.192	0.230	0.186	19.85%	-2.92%	19.85%	2.92%
12	0.216	0.191	0.214	0.218	12.27%	14.12%	12.27%	14.12%
13	0.216	0.217	0.245	0.210	12.53%	-3.58%	12.53%	3.58%
14		0.198	0.230	0.219	15.68%	10.25%	15.68%	10.25%
15		0.274	0.308	0.286	12.30%	4.32%	12.30%	4.32%
			Average		13.65%	1.46%	13.65%	8.27%

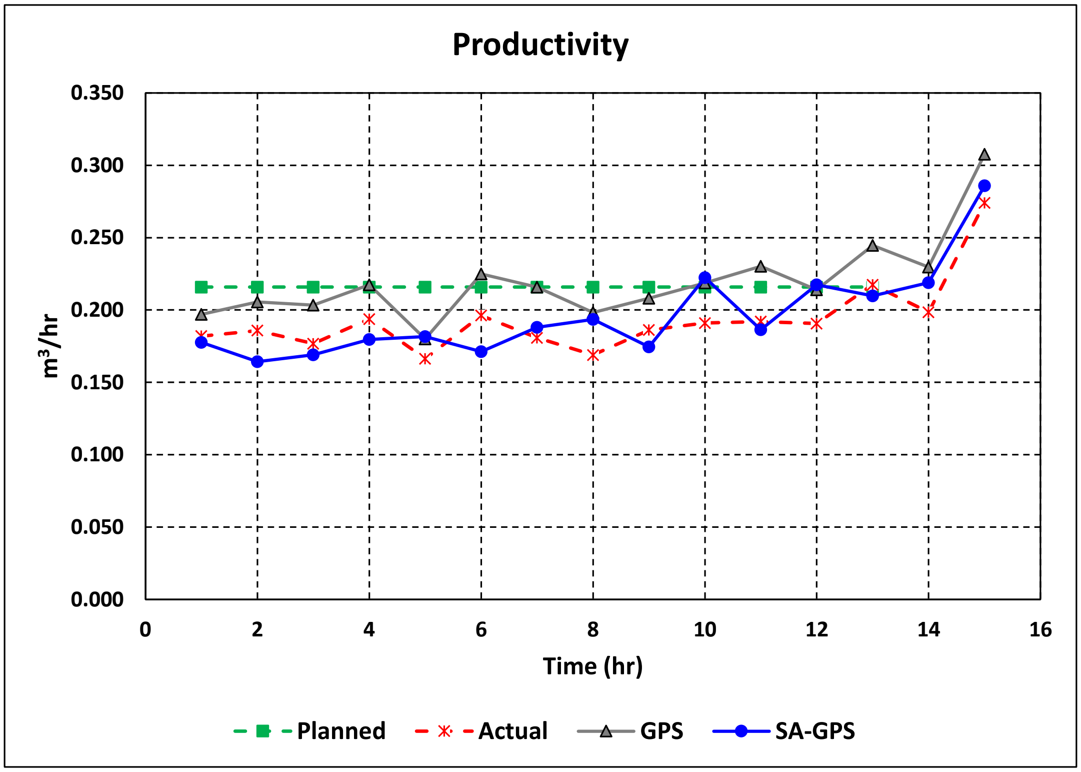


Figure 4-49: Project Hourly Productivity

Another Important performance measure is the differential productivity, which enables detection of trends and patterns in operations on-site. Table 4-14 and Figure 4-50 depict the actual differential productivity encountered on site, and the estimates using both methods. The actual average differential productivity was -10.41%, and it reflected the learning curve of the crew utilizing the remote controlled equipment, where the differential productivity started to trend up around T=9 and all the way to the end of the project duration. The SA-GPS had better estimate of the average differential productivity as -9.18%, while the standalone GPS estimate was way off with an average value of 1.79%.

Table 4-14: Differential Productivity Comparison

Time (h)	Planned Productivity m3/hr	Actual Productivity m3/hr	Productivity		Differential Productivity		
			GPS m3/hr	SA-GPS m3/hr	Actual	GPS	SA-GPS
1	0.216	0.182	0.197	0.178	-15.70%	-8.70%	-17.73%
2	0.216	0.186	0.206	0.164	-13.92%	-4.74%	-23.91%
3	0.216	0.177	0.203	0.169	-18.10%	-5.77%	-21.64%
4	0.216	0.194	0.217	0.180	-10.24%	0.71%	-16.76%
5	0.216	0.166	0.180	0.182	-22.93%	-16.63%	-15.82%
6	0.216	0.196	0.225	0.171	-9.09%	4.21%	-20.69%
7	0.216	0.181	0.216	0.188	-16.22%	0.01%	-12.90%
8	0.216	0.169	0.198	0.194	-21.72%	-8.13%	-10.30%
9	0.216	0.186	0.208	0.175	-13.74%	-3.59%	-19.10%
10	0.216	0.191	0.219	0.222	-11.50%	1.34%	3.05%
11	0.216	0.192	0.230	0.186	-11.01%	6.65%	-13.61%
12	0.216	0.191	0.214	0.218	-11.68%	-0.85%	0.79%
13	0.216	0.217	0.245	0.210	0.76%	13.39%	-2.84%
14	0.216	0.198	0.230	0.219	-8.04%	6.38%	1.39%
15	0.216	0.274	0.308	0.286	26.97%	42.58%	32.45%
			Average		-10.41%	1.79%	-9.18%

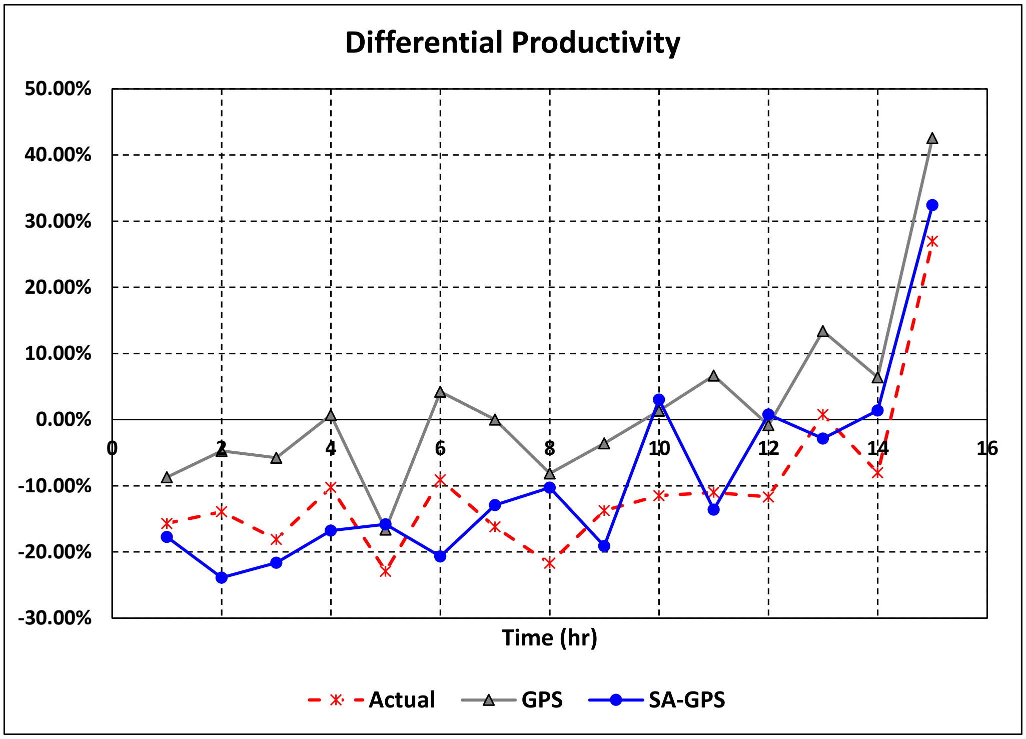


Figure 4-50: Project Differential Productivity

Equipment utilization was calculated using SA-GPS, the loader utilization was 29%, where the trucks utilization was 99% as shown in Figure 4-51.

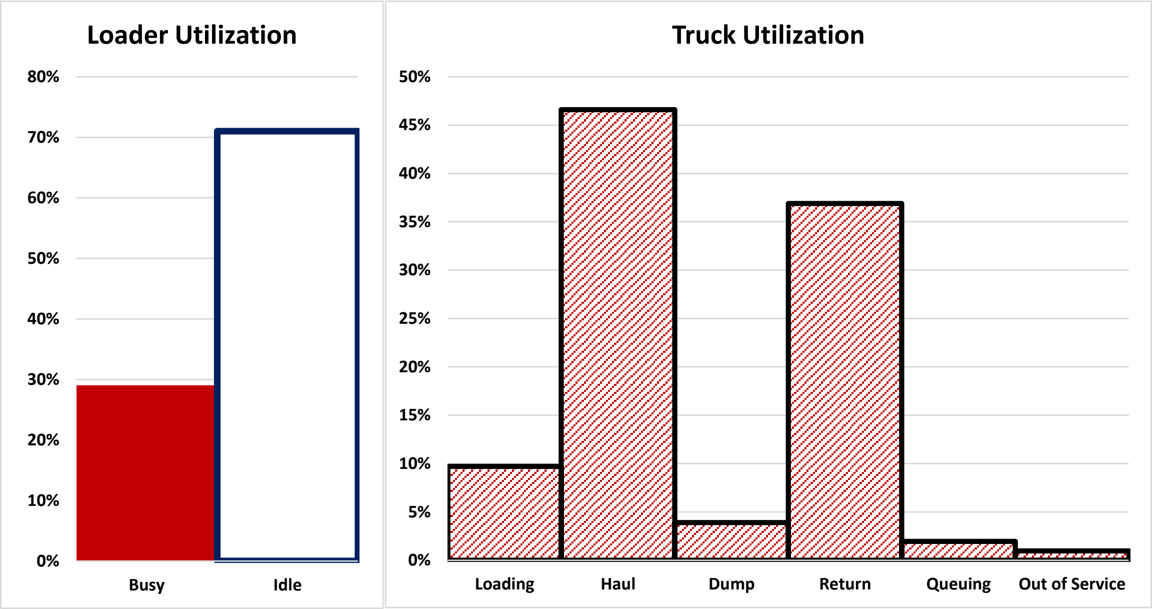


Figure 4-51: Equipment Utilization

In order to validate the functionality of the developed productivity analysis algorithm. Simulated scenarios are used to test the capabilities of the developed algorithm in identifications of bottlenecks in the earthmoving operations. The algorithm use IF-Then rules to check for conditions listed in table Table 4-8. For example, the rule to check for changing soil conditions is:

```
If
    {(Loading Time > Average Loading Time) and (Number of Buckets > Average Number of Buckets) and (Load Weight > Average Load Weight)}
    Or
    {(Loading Time < Average Loading Time) and (Number of Buckets < Average Number of Buckets) and (Load Weight < Average Load Weight)}
Then (Soil type is changing)
```

A scenario for a change in soil conditions was simulated by adding some rocks to the sandy soil. The loading time started to increase and the number of buckets required to fill the truck was increased from 5 to 6. A flag for changing soil was identified at cycle 16 as shown in Figure 4-52.

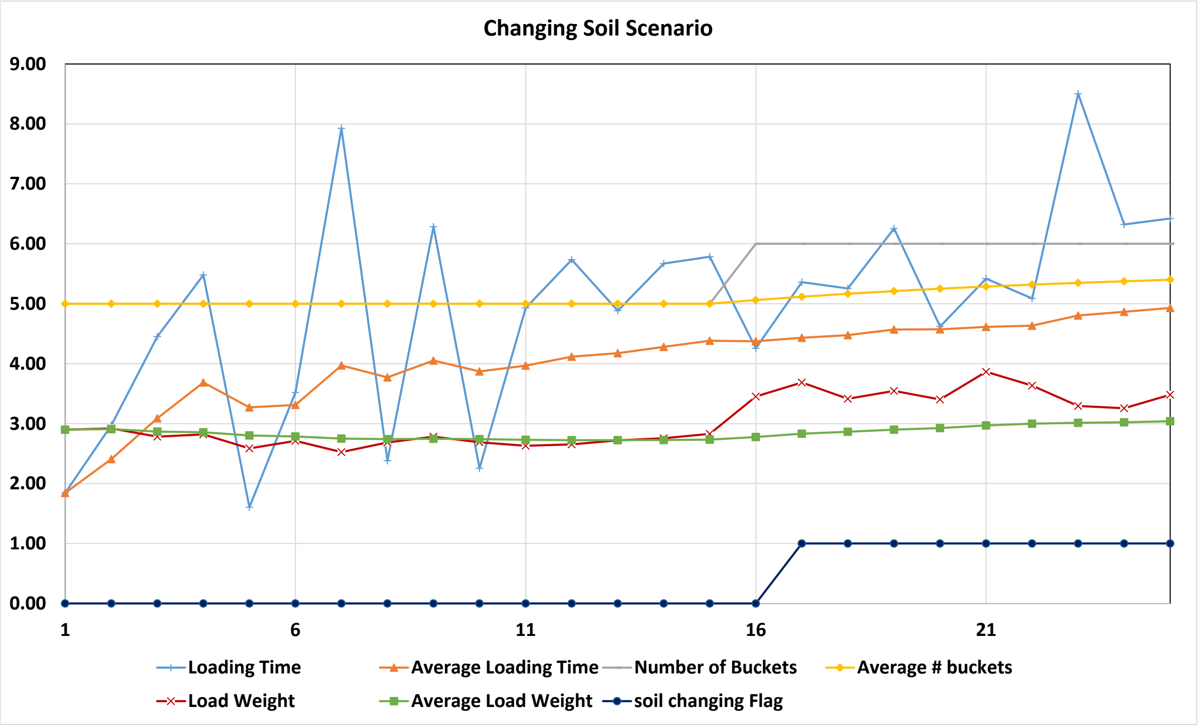


Figure 4-52: Changing Soil Conditions Scenario

Another scenario was test, a truck out of service on the road, which in turn cause excessive loader idle time. The rule to check for truck out of service conditions is:

```
If
    {(Loader Idle Time > Average Loader Idle Time) and (Truck$_i$ Cycle Time > Average Trucks Cycle Time)}
Then (Potential Truck$_i$ Out of Service)
```

This scenario was simulated by taking one truck out of service between times (12,770 to 13,770) and (26,360 to 27,360). The loader idle time started to increase and a flag was raised by the algorithm as shown in Figure 4-53.

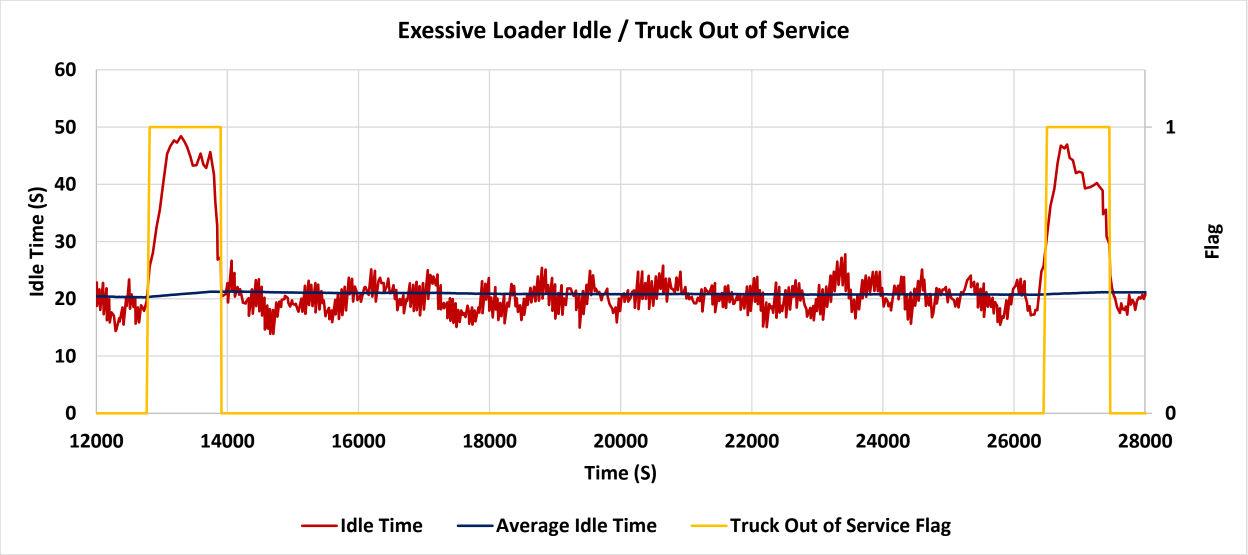


Figure 4-53: Truck Out of Service Scenario

4.10.2 Discussion of Results

The cumulative probability functions (CDF) of a system's estimation error is used to measure its precision. To compare the developed SA-GPS prototype with the standalone GPS method with respect to accuracies and precision, the technique whose CDF graph reaches high probability values faster is more preferable, because its estimation error is more concentrated in small values. With 95% confidence, the developed SA-GPS has a progress estimation accuracy of 93.36%, while the standalone GPS has a progress estimation accuracy of 85.05% as shown in Figure 4-54.

The lower accuracy of the standalone GPS method is attributed to its complete reliance on only one source of data, which is used to detect equipment location onsite. Then an algorithm correlates that location to activities (load, haul, dump and return) being executed, to identify their parameters such as start, finish and durations. It also assumes that hauling units are loaded to their full capacity. Although that GPS accuracy is independent of scale of jobsite, location detection accuracy is directly proportional to size and scale of jobsite. So given the small size of the jobsite considered in this case study, the location detection accuracy is lower compared to that expected on larger real-size jobsites.

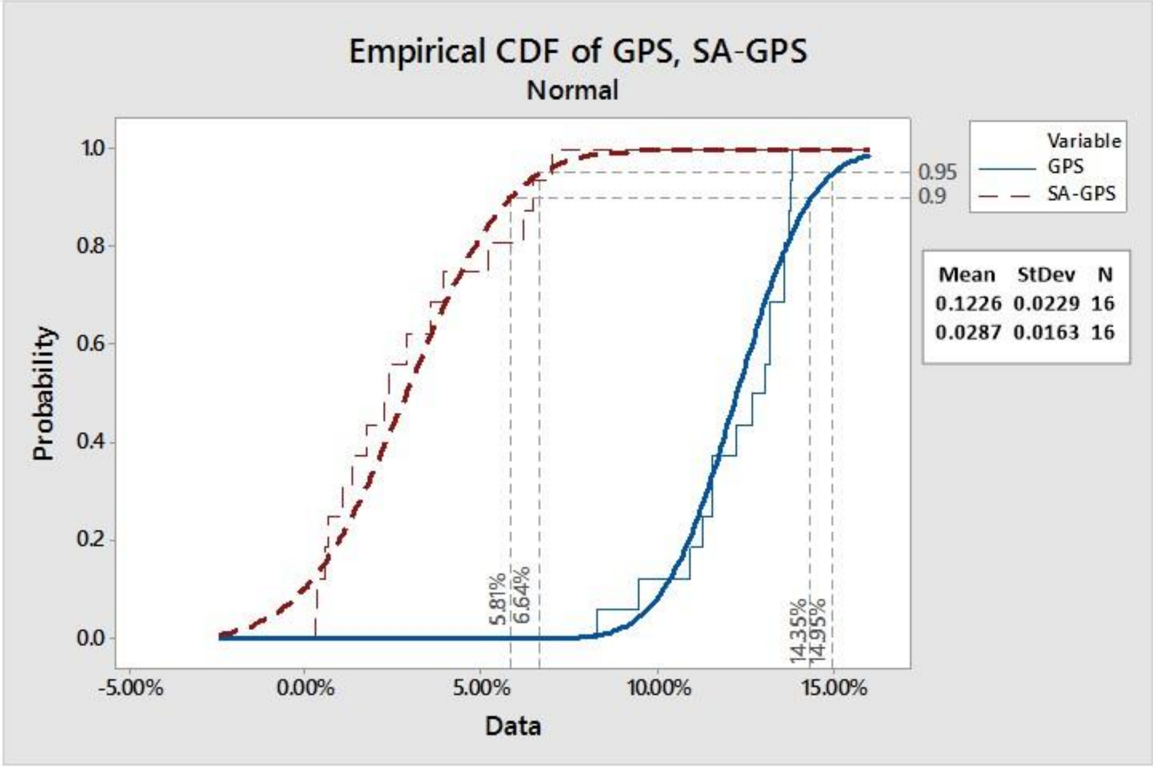


Figure 4-54: Progress Estimation Errors CDF

Contrary to the standalone GPS method, the accuracy of the developed SA-GPS prototype is not size or scale dependent of the construction operation being monitored. The prototype is designed and configured to detect equipment physical movement and their inter-action which also is independent of size and scale. The results showed that using multiple sensors in addition to GPS, augments its capabilities. The improved accuracy can be attributed to the whole cluster of sensors and the reasoning engine used to fuse sensors data as described earlier in the productivity assessment module.

The improved progress estimation of the developed prototype not only reduces the risk of project cost and duration overruns, but also enables project managers to observe and oversee their project's status in real-time with the support of intuitive and timely progress reports. It is expected that the accuracy of the developed prototype can be achieved on large-scale road and highway construction projects. This is attributed to the fact that the developed prototype does not require any scalable improvements on the hardware technology used, nor does it require any additional computational changes in the developed software. The performance of the developed SA-GPS prototype was measured against the five performance measures described in section 4.3. Table 4-15 summaries the performance measures for the SA-GPS:

Table 4-15: SA-GPS Performance Measures

Performance Measure	**Value**
Accuracy	93.36%
Precision	95%
Latency	Less than 700 milliseconds
Scalability	Up to 15,000 nodes can be connected at one time
Robustness	It is able to perform if sensor data is noisy, corrupted or missing
Cost	$100-150 (depends on number of sensors)

The developed prototype is able to capture real-time operating information such as equipment idle and out of service times, along with operator driving behavior and weather conditions, which facilitates operations optimization and potentially reduce their costs. The developed prototype paves the road to a wide range of applications in construction such as progress tracking, jobsite safety and security enhancement, productivity assessment and facilities management.

4.11 Summary

This chapter described the development of the SA-GPS prototype for outdoor automated data acquisition on construction jobsites. The developed prototype was designed with a special focus on automated tracking and control of earthmoving operations in near real time. The developments made in this chapter is contributing to the body of knowledge in:

- Experimental investigation to assist designing flexible and customized automated data acquisition prototypes utilizing latest innovations in sensory and wireless technologies.
- Testing and selecting wireless protocols, sensors for efficient coverage, data capturing and communication
- Design, configure, test and validate the developed prototype to address limitations in current practice in automated site data acquisition and off-the-shelf technologies.
- Design data fusion algorithm for timely extraction of actionable information in support of near real-time productivity analysis for earthmoving operations.
- Validation of concurrent design and testing of the developments made for automated site data acquisition using rapid prototyping techniques.
- Development of efficient data management scheme that utilizes integrated on-sensor node data processing and in-network data processing to transform raw data into high level useful and actionable information. In this context, data aggregation and processing render much faster near real-time progress measurements.
- Development of fuzzy reasoning data fusion and knowledge extraction algorithm to provide a higher level of system redundancy in case of sensors malfunctions.

Chapter 5 : SELF-CALIBRATED WSN PROTOTYPE

5.1 General

The aim of this chapter is to describe the study made leading to the development of Self-Calibrated Wireless Sensor Network (SC-WSN) for indoor tracking and progress reporting of construction operations, with a focus on improving the localization accuracy. Figure 5-1 depicts the main sections of this chapter.

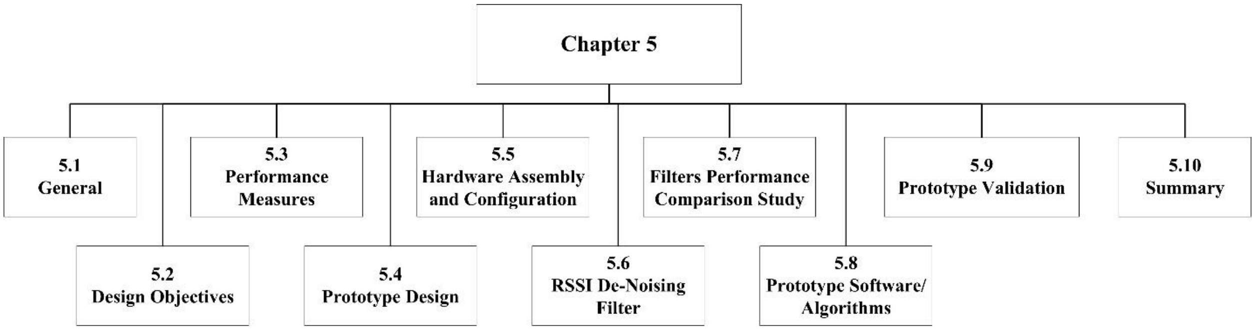

Chapter 5
5.1 General
5.2 Design Objectives
5.3 Performance Measures
5.4 Prototype Design
5.5 Hardware Assembly and Configuration
5.6 RSSI De-Noising Filter
5.7 Filters Performance Comparison Study
5.8 Prototype Software/ Algorithms
5.9 Prototype Validation
5.10 Summary

Figure 5-1: Chapter 5 Overview

5.2 Design Objectives

The main design objectives for the SC-WSN prototype are:

- Select best filtering technique for RSSI for indoors localization.
- Improve indoor localization accuracy.
- Design a dynamic and adaptable path-loss model.
- Test and validate the prototype design using simulation and laboratory testing.

5.3 Performance Measures

The performance measures are similar to those utilized in chapter 4, with a focus on the localization accuracy. The Euclidean distance is used to measure the localization error. The average distance error is adopted as the performance metric, which is calculated using equation (5-1):

$$E(distance\ error) = E(\sqrt{(x-x')^2 + (y-y')^2}) \tag{5-1}$$

5.4 Prototype Design

Similar to the iterative process of rapid prototyping design presented in chapters 3 and 4, a similar approach was utilized. The SC-WSN prototype is designed to address the limitations highlighted in chapter 3. The developed prototype consists of fixed readers, gateway and mobile tags as shown in Figure 5-2. This prototype is based on radio frequency technology, and its hardware components are selected to satisfy the performance matrices (accuracy, precision, robustness and cost). Tags are the mobile units, which will be carried by or mounted on tracked resources (labor, material and equipment).

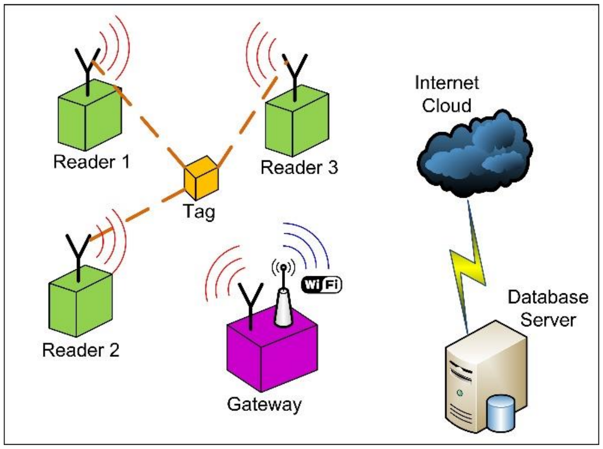


Figure 5-2: Indoor Positioning prototype Components

Readers are the fixed units, which are mounted at key locations on a construction site to provide appropriate RF coverage. Each reader has a unique ID, and a predefined location. A reader is continually scanning for nearby tags, once a tag is detected; it measures its signal strength. Then the microcontroller converts the measured signal strength to distance according to a predefined

RF propagation model. It transmits the distance value with the reader coordinates (x,y) back to the tag. Once it receives three readings from three different readers, a 2D trilateration is applied to calculate the tag location in 2D, then it adds the third dimension (elevation) from the reading of its pressure sensor. Finally the tag transmits its ID, and location to the nearest gateway. Gateways are fixed units with higher processing and storage capabilities. A gateway consists of a microcontroller with RF transceiver, a wireless LAN communication module, a real time clock, a data logger and a power supply. The gateway collects the data from the tags and readers, then performs pre-data processing. Locations are recorded with time stamp on the gateway internal storage and transmitted periodically to a cloud based algorithm for linking with BIM and post processing. In order to increase accuracy without placing more readers, the model employs the idea of having extra fixed location reference tags to help location calibration. These reference tags serve as reference points in the system as shown in Figure 5-3.

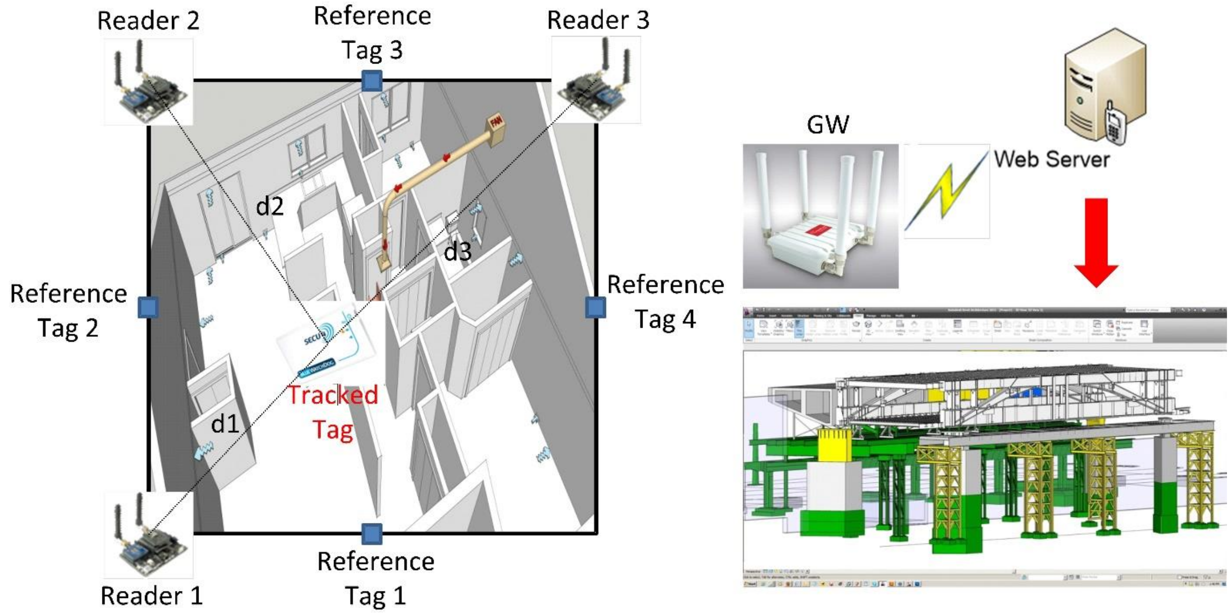


Figure 5-3: Reference Tags for Self Calibration

A two stage process is implemented in this prototype, prediction stage and calibration stage as shown in Figure 5-4. Each reader measures the signal strength (RSSI) from nearby tags and filter it to remove uncorrelated noise. The prediction stage is initiated by converting the filtered RSSI to its corresponding distance using the dynamic path-loss model. The initial settings for the signal propagation model parameters are calculated using indoor experimentations as explained by (Ibrahim & Moselhi, 2014d). Once three distances from three readers for a given tag are available, its location is estimated using the LSE trilateration algorithm. The localization accuracy is continually monitored by measuring the errors in location estimation generated based on a number of reference tags. These reference tags are deployed on site at pre-defined locations. When the system accuracy is degraded due to on-site interferences, a system calibration request is initialized. The user can define the accuracy limits to initiate the calibration requests. The calibration stage utilizes a particle swarm optimization (PSO) to find the best values for the path-loss model parameters which maximize the system localization accuracy. Finally, at the end of the calibration stage, the dynamic path-loss model is updated with a new set of optimized parameters.

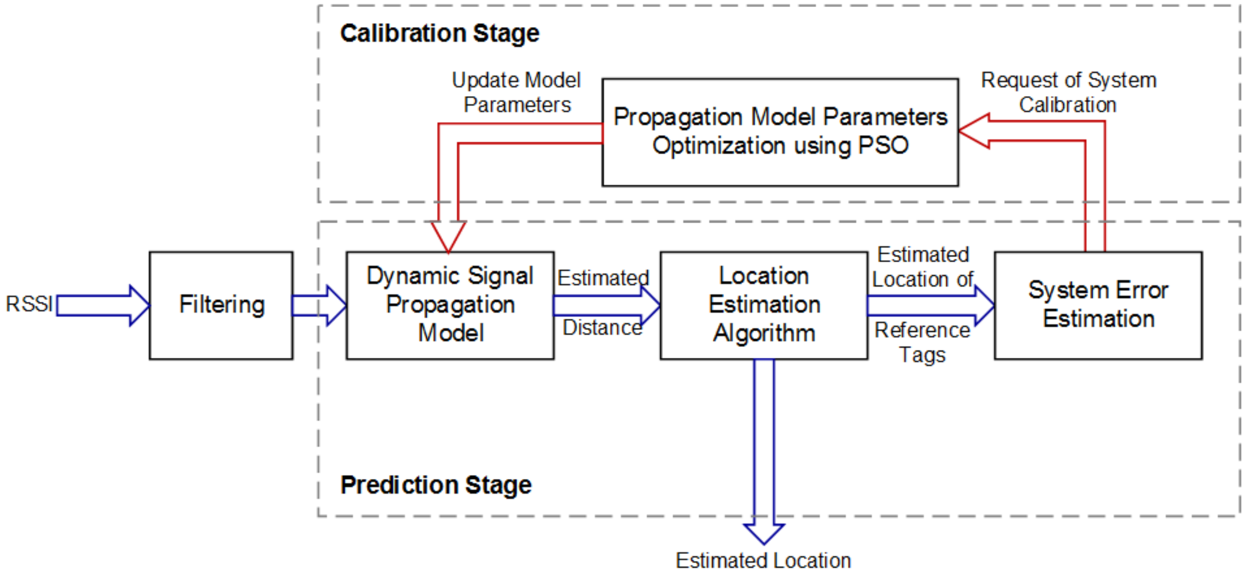

Calibration Stage
Update Model Parameters
Propagation Model Parameters Optimization using PSO
Request of System Calibration
RSSI
Filtering
Dynamic Signal Propagation Model
Estimated Distance
Location Estimation Algorithm
Estimated Location of Reference Tags
System Error Estimation
Prediction Stage
Estimated Location

Figure 5-4: Developed Localization Prototype Overview

Several researchers have used different wireless technologies at 2.4 GHz and RF lower frequency bands (433 MHz - 928 MHz) in indoor localization research. When choosing a frequency band for an indoor positioning system; key criteria are power management, network reliability, and network density. Many researchers are tempted to choose the 2.4 GHz band because it is the most globally accepted. However, its popularity carries increasing demand on the band, which results in deteriorating performance and requires complicated RF protocols to overcome interference. (Ibrahim & Moselhi, 2014d) indicated that when considering several factors such as coverage range, data transfer rate and radio interference, the 900 MHz frequency band is consider the best trade-off between all these factors (considering equal weights) as shown in Figure 5-5.

900 MHz has a much longer wavelength, which provides greater physical barrier penetration and is far less popular, which makes the frequency range less crowded. Because of these two factors, given enough transmit power and receptor sensitivity, it is considered the best alternative for indoor localization.

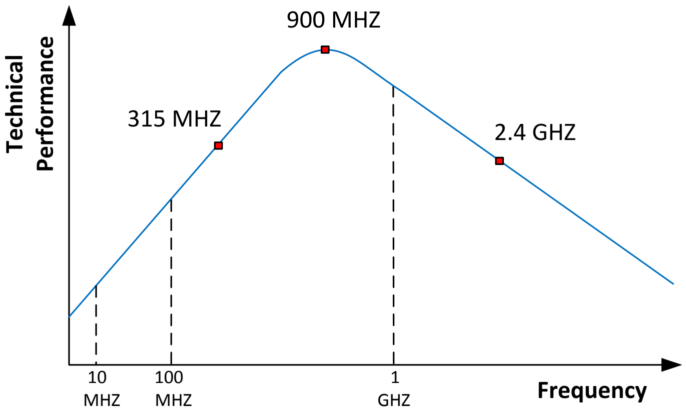


Figure 5-5: Operating Frequency Vs Performance

Three major hardware components are used to build the prototype: microcontroller, RF module and Barometric pressure sensor. Waspmote microcontroller is used which is a more advanced microcontroller than the Arduino Uno used in the SA-GPS as shown in Figure 5-6. It is a 32 bit microcontroller with 128 Kb of flash memory, which enables executing advanced optimization algorithms. It can be programmed using C++, and it has an Open Source API to program using high level functions without having to deal with hardware-specific commands.

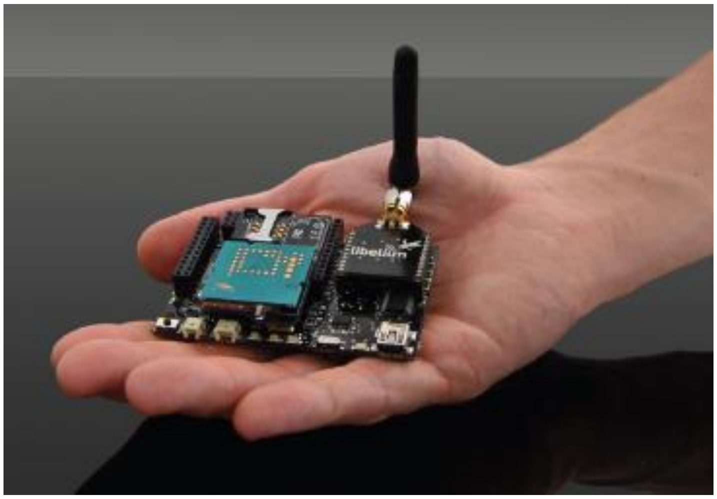


Figure 5-6: Waspmote Microcontroller

Synapse SNAP radio frequency module RF300 is used based on the experimental study presented in chapter 4. BMP180 barometric pressure sensor as explained in chapter 4, is used for elevation measurement identical to that used in chapter 4.

5.5 Hardware Assembly and Configuration

Considering the function and design objectives, the prototype is designed to be consisted of three main parts: the tag, the reader and the gateway. Each of them encompass hardware components and firmware scripts. The firmware scripts are programmed to set the main parameters required for the prototype and to control hardware functions. The tag is equipped with the Synapse RF300PC engine and the BMP180 pressure sensor. The communication between the BMP180 and RF300PC is established through Inter-Integrated Circuit (I2C) communication. The required minimum power is 2.5V. So the tag will be powered by a Li-Ion Rechargeable Coin Cell batteries (PD3555) to keep the tag small as shown in Figure 5-7.

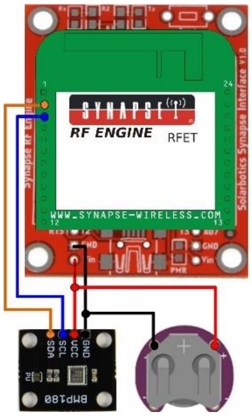


Figure 5-7: Tag Wiring Diagram

Tag's firmware consists of two main scripts. The first script performs a multicast function to surrounding readers. The multicast function pings the nearby readers and passes the tag MAC address to the readers to perform the RSSI measurement as shown in Figure 5-8. The second script performs the localization when three distance are received from surrounding readers, and it measures the barometric pressure then converts into corresponding altitude. Finally, it transmits the 3D location of the tag to the nearest gateway as shown in Figure 5-9.

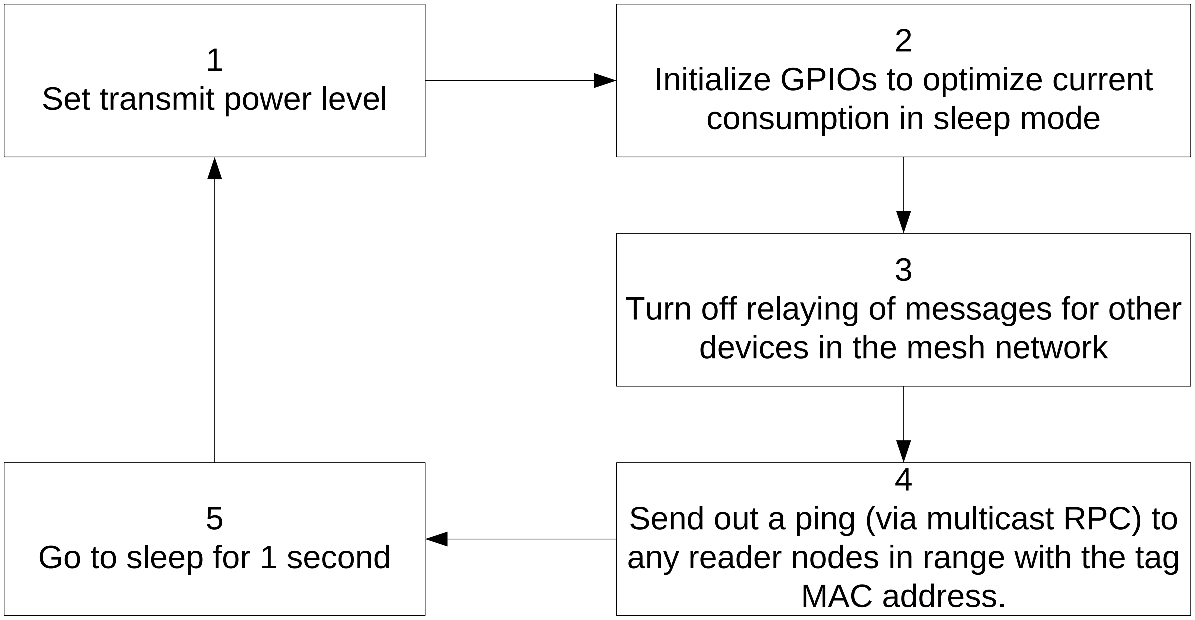

1
Set transmit power level
2
Initialize GPIOs to optimize current consumption in sleep mode
3
Turn off relaying of messages for other devices in the mesh network
4
Send out a ping (via multicast RPC) to any reader nodes in range with the tag MAC address.
5
Go to sleep for 1 second

Figure 5-8: Tag Multicast Script Flowchart

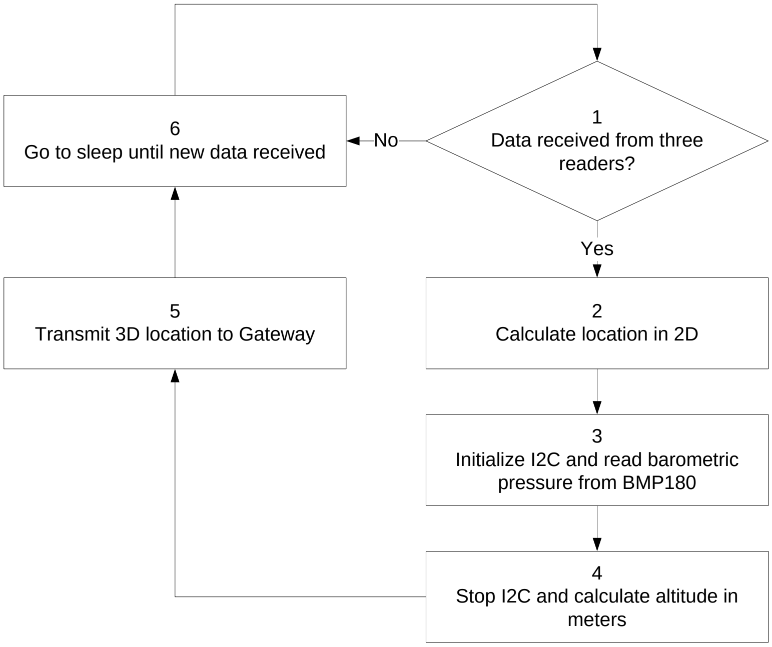

1
Data received from three readers?
No
Yes
2
Calculate location in 2D
3
Initialize I2C and read barometric pressure from BMP180
4
Stop I2C and calculate altitude in meters
5
Transmit 3D location to Gateway
6
Go to sleep until new data received

Figure 5-9: Tag Localization Script Flowchart

The reader is equipped with the Synapse RF300PC engine and powered by a 3 V DC 400 mA Power Adapter. The RF module has Omni directional antenna for extended range. Similar to the tag wiring setup, (-) on the DC plug goes to the pin 24 on the module and (+) goes to pin 21 on the module, as shown in Figure 5-10. The reader listens to ping requests from tags. Once a request is received the reader records the RSSI of the transmitted packet, then it is filtered and converted to its corresponding distance. The reader then sends the distance with the reader coordinates back to the tag as shown in Figure 5-11.

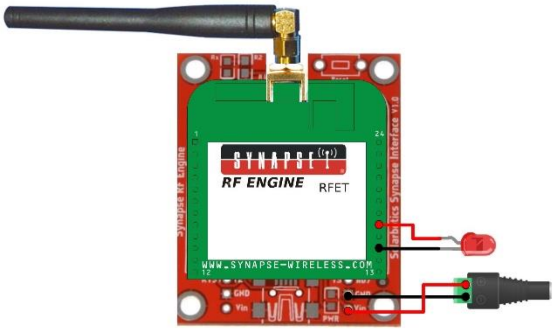


Figure 5-10: Reader Wiring Diagram

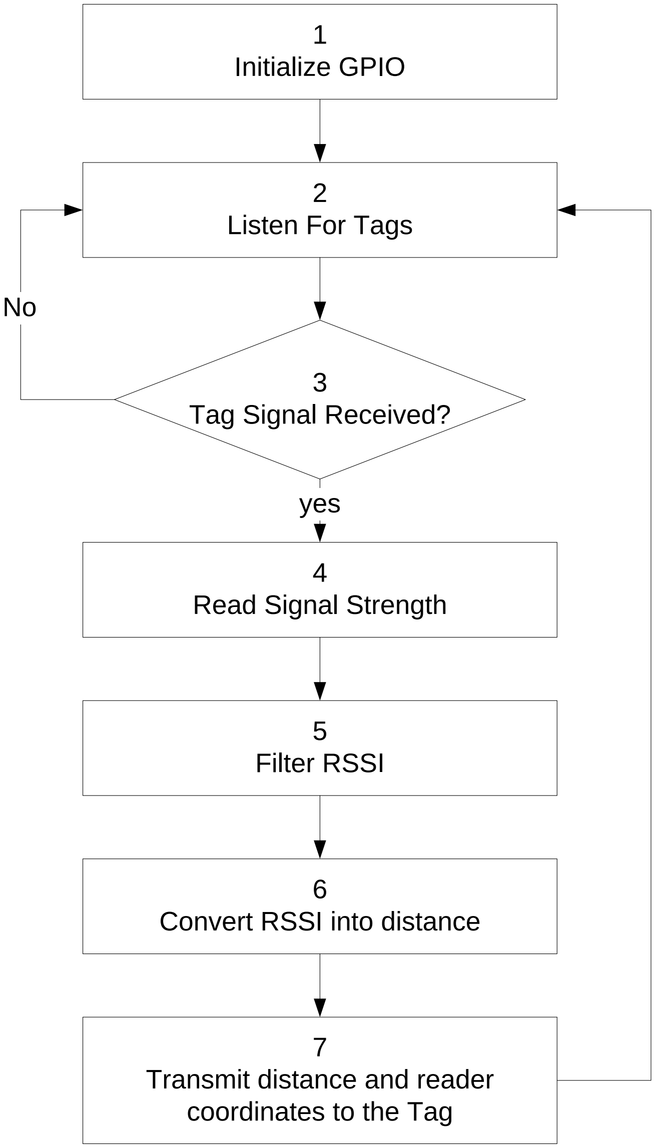


Figure 5-11: Reader Script Flowchart

The Gateway node is used to collect data from the readers, and tags; then transmit the data to the cloud based database for post-processing. The Gateway node also responsible for monitoring the system performance and auto calibrating the system when the localization accuracy is low as shown in Figure 5-12. It consists of a Waspmote microcontroller, Synapse RF300 module, RN-171 WLAN module and 2GB SD card as shown in Figure 5-13.

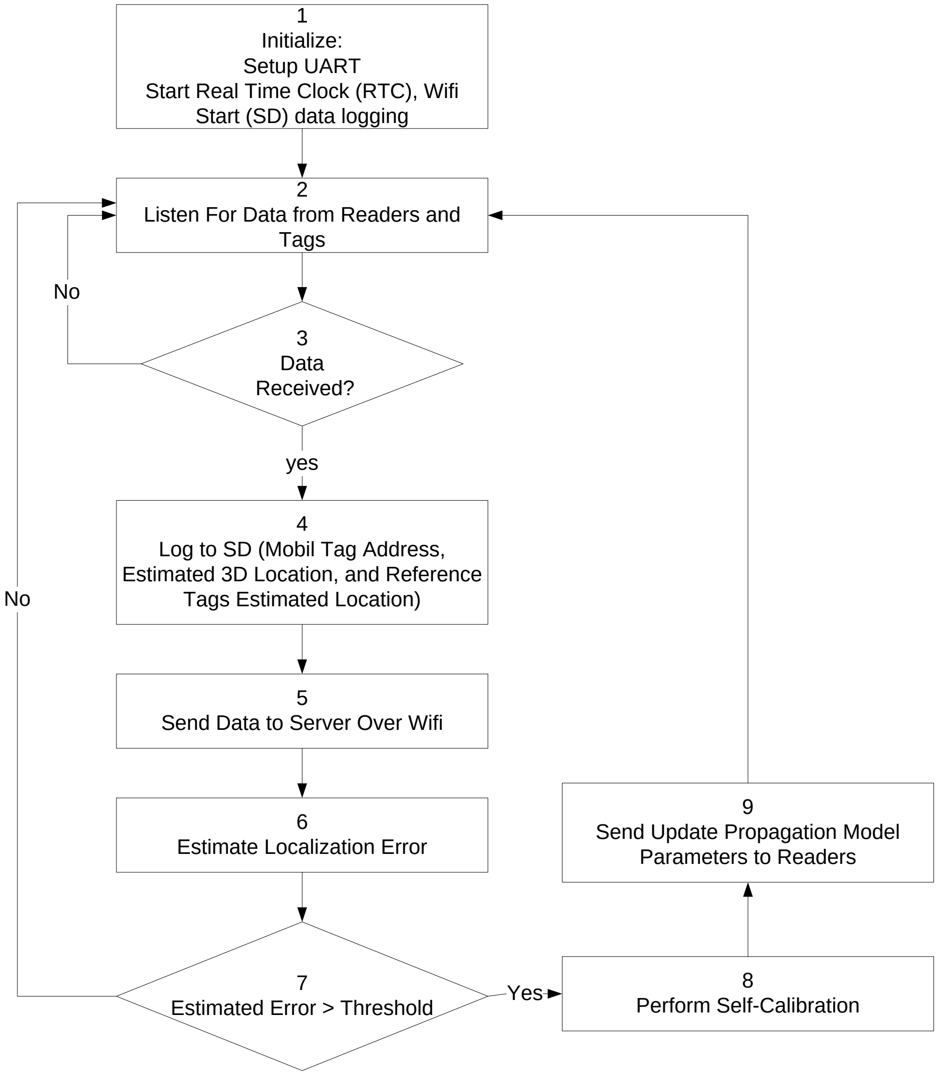

1
Initialize:
Setup UART
Start Real Time Clock (RTC), Wifi
Start (SD) data logging
2
Listen For Data from Readers and Tags
No
3
Data
Received?
yes
4
Log to SD (Mobil Tag Address, Estimated 3D Location, and Reference Tags Estimated Location)
No
5
Send Data to Server Over Wifi
6
Estimate Localization Error
9
Send Update Propagation Model Parameters to Readers
7
Estimated Error > Threshold
Yes
8
Perform Self-Calibration

Figure 5-12: Gateway Script Flowchart

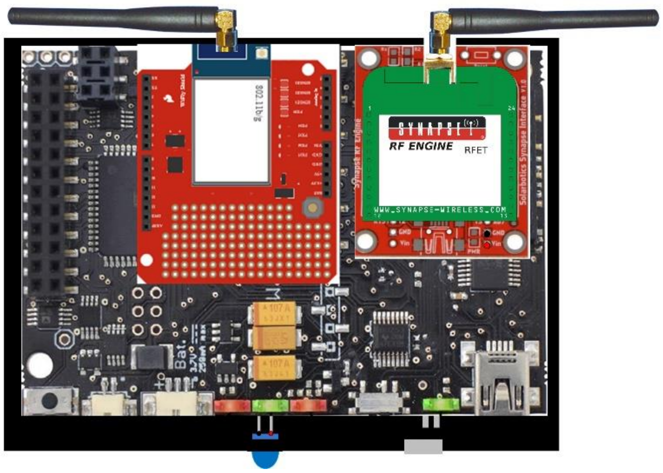


Figure 5-13: Gateway Node Wiring Diagram

5.6 RSSI De-Noising Filter

Theoretical laws of electromagnetic wave propagation describe propagation losses when waves are travelling in ideal free-space situations, which become very challenging when applied to actual indoor localization situations. These challenges arise from the lack of prediction methods for actual propagation losses on complex and dynamic jobsites. In ideal free-space situations, electromagnetic waves travel or propagate in direct rays from transmitter to receiver. However in actual situations, waves pounce as they are reflected and scattered from surrounding environment such as floor, ceiling, walls and various objects. Which in turn cause multipath waves, which can be either constructive or destructive, resulting in a positive or a negative effect on the received signal strength. Such interference is more complicated in complex and dynamic environments such as construction jobsites, where losses are continually changing. Theoretically, RSSI is inversely proportional to the distance between a transmitter and a receiver. However, interferences such as multipath and shadow fading affects RSSI as shown in Figure 5-14. It is obvious that the RSSI heavily depends on surrounding environment. The comparison between electromagnetic signal propagation in free-space (Anechoic chamber) and corridor environments shows the multipath effect, where the electromagnetic signal is pouncing in a random manner.

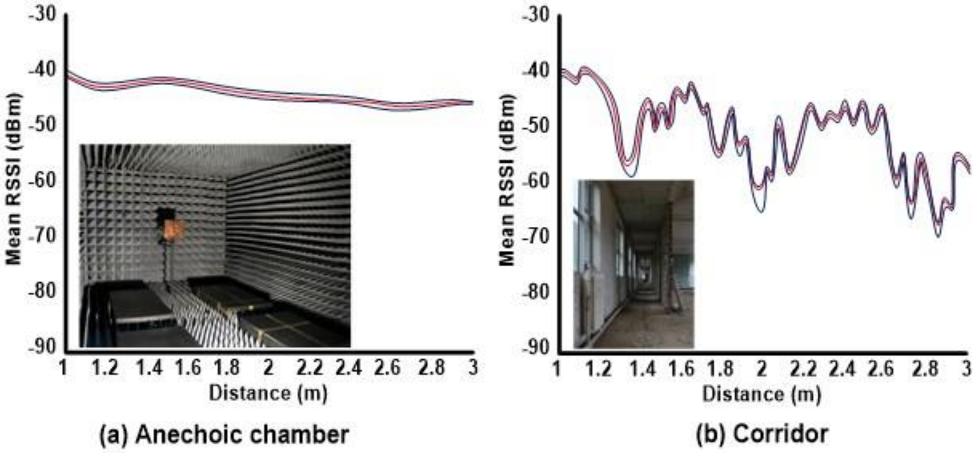


Figure 5-14: Signal Propagation in Different Environments (Rensfelt, 2012)

From the collected RSSI data as explained in details in chapter 4, it was clear that RSSI is affected by the periodic/random changes in the physical properties of the surrounding environment or even a group of people passing around the transmitter or receiver as shown in Figure 5-15. Such variations in the RSSI readings (even when the node is at standstill) produce huge errors in the estimate distance. A simple moving average could be used to filter out small oscillations in the RSSI, however further investigation is required to select the best filtering and smoothing method. The following section will present a deep analysis for selecting the most effective filtering scheme for the localization problem.

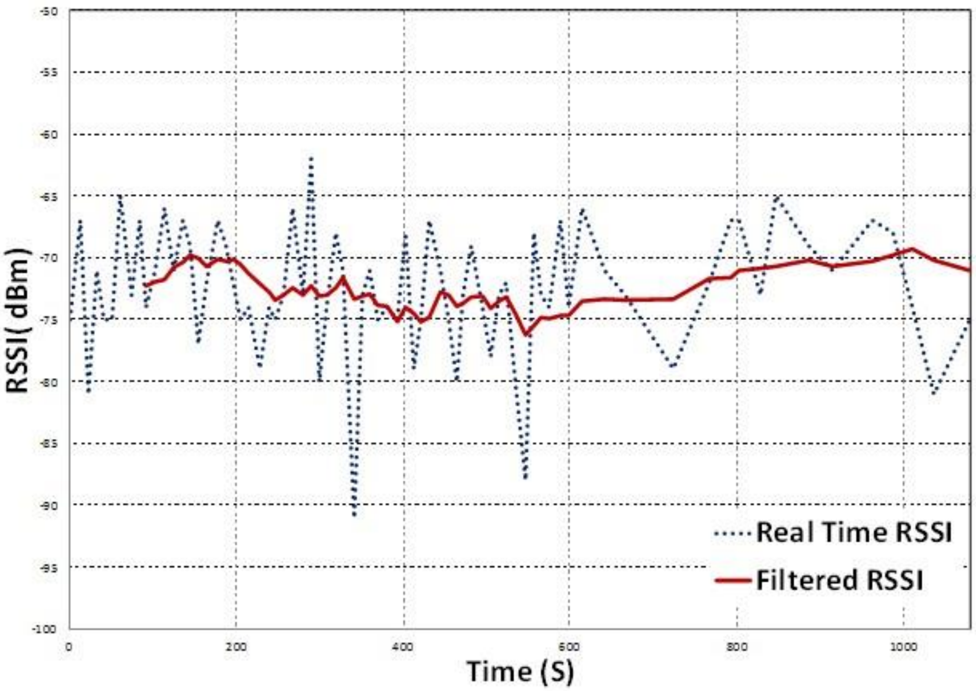


Figure 5-15: Real-Time RSSI VS Moving Average Filtered RSSI

RSSI measurements presented chapter 4 can be unreliable for localization due to the noise interference. Signal de-noising is required to generate a signal which is representative of the original RSSI but less noisy and suppresses interferences caused by surrounding environment (Ibrahim & Moselhi, 2015a). The goal of a filter is to reduce noise while maintaining the shape and height of waveform peaks. Moving average filter is the most common filter used for RSSI (eg. (Montaser & Moselhi, 2014)), however it has several limitations that are explained in detailed in the following section. It was important to investigate different filtering techniques in order to choose the best filtering technique for RSSI with a focus on indoor localization application. Three main filters are considered in this study: Moving Average filter, Savitzky-Golay filter, and Kalman filter.

5.6.1 Moving Average Filter

The moving average is the most common filter in digital signal processing, mainly because it is easy to understand and use. It is optimal for reducing random noise while retaining a sharp amplitude response. However, it has a major limitation due to its inability to separate band frequencies. The moving average operates by averaging a number of points from the input signal to produce each point in the output signal. In equation form, this is written:

$$Y(i) = \frac{1}{m}\sum_{j=0}^{m-1} X(i+j) \qquad (5\text{-}2)$$

Where X(i+j) is the input signal, Y(i) is the output signal, and m is the number of points in the average.

5.6.2 Savitzky-Golay Filter

Savitzky and Golay (1964) presented an algorithm for data smoothing using least squares fit of a small set of consecutive data points to a polynomial and then calculates central point of the fitted polynomial curve as the new smoothed data point. The main advantage of this filter is its ability to keep features of the original data distribution such as relative maxima and minima which are often flattened by other smoothing techniques such as moving average (Hassanpour, 2008; Luo, Ying, & Bai, 2005; Savitzky & Golay, 1964). The mathematical formula for Savitzky-Golay algorithm is given by the following equation:

$$Y_k = \frac{\sum_{i=-n}^{n} A_i X_{k+i}}{\sum_{i=-n}^{n} A_i} \tag{5-3}$$

Where Y_k is the smoothed data point, X_{k-n} --- X_{k+n} are the original data points, (2n+1) is the window width for the filter, and A_i are the convolution integers which depends on the filter width and the polynomial degree. Typical sets of convolution integers for “quadratic smooth” are shown in the Table 5-1.

Table 5-1: Convolution Integers for “Quadratic Smooth”

	Filter width (2n+1)			
i	11	9	7	5
-5	-36			
-4	9	-21		
-3	44	14	-2	
-2	69	39	3	-3
-1	84	54	6	12
0	89	59	7	17
1	84	54	6	12
2	69	39	3	-3
3	44	14	-2	
4	9	-21		
5	-36			

The performance of Savitzky-Golay filter is usually better than the standard averaging filters. It is able to preserve the signal’s high frequency, however it is not effective in rejecting noise.

Kalman filter was first introduced in 1960 to present a solution for discrete data linear filtering problem (Kalman, 1960). Since then, extensive research and applications had been proposed particularly in the areas of robotics and navigation. The key advantage of the Kalman filter is its simple computational algorithm, adaptive recursive nature, and its status as the optimal estimator for one-dimensional linear systems with Gaussian error statistics (Anderson & Moore, 2012).

5.6.3 Kalman Filter

Kalman filter estimation process is based on a feedback loop control system. Which first estimates the process's state at a point in time and then obtains feedback of measurements. This feedback measurement is used to adjust the model parameters for next estimate. The model assumes that the state of a system at a time t evolved from the prior state at time t-1 according to the equation:

$$X_t = A_t X_{t-1} + B_t u_{t-1} + w_t \tag{5-4}$$

where X_t is the process state vector at time t, A_t is the state transition matrix which is applied to the previous state X_{t-1}, u_t is the control input vector, B_t is the control-input model which is applied to the control vector u_t, and w_t is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance Q_t.

At time t a measurement Z_t of the true state X_t is calculated according to

$$Z_t = H_t X_t + v_t \tag{5-5}$$

Where H_t is the measurement model for mapping true state space into measurement space and v_t is the measurement noise which is assumed to be zero mean Gaussian white noise with covariance R_t.

The Kalman filter recursive estimator model as shown in Figure 5-16 has two phases, the prediction phase, which estimates the priori process state at next observation time, and the correction phase, which incorporates a new measurement into the a priori estimate to obtain an improved a posterior estimate.

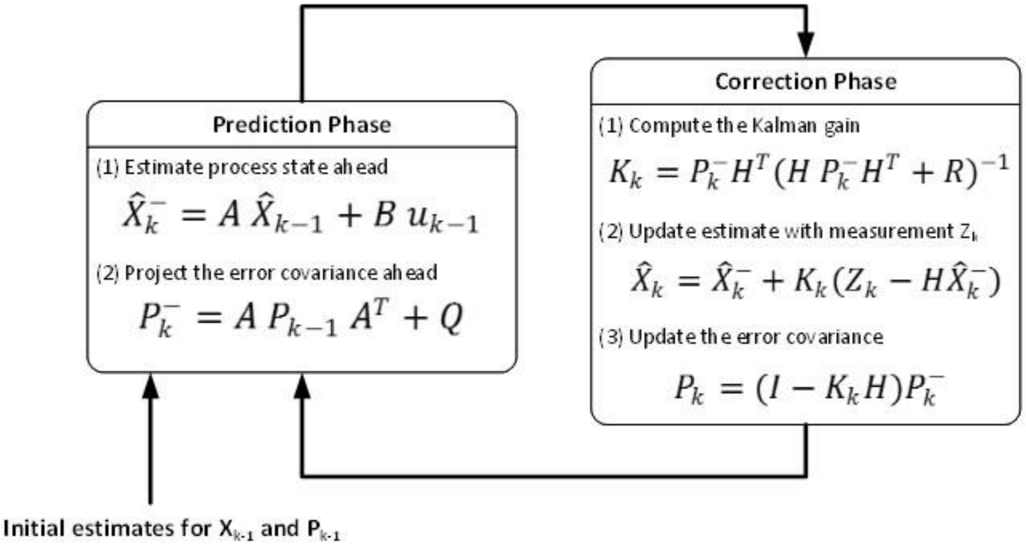

Prediction Phase
(1) Estimate process state ahead
$\hat{X}_k^- = A\,\hat{X}_{k-1} + B\,u_{k-1}$
(2) Project the error covariance ahead
$P_k^- = A\,P_{k-1}\,A^T + Q$
Correction Phase
(1) Compute the Kalman gain
$K_k = P_k^- H^T (H\,P_k^- H^T + R)^{-1}$
(2) Update estimate with measurement Z_k
$\hat{X}_k = \hat{X}_k^- + K_k(Z_k - H\hat{X}_k^-)$
(3) Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates for X_{k-1} and P_{k-1}

Figure 5-16: Kalman Filter Recursive Estimator Model

In the context of RSSI de-noising, a simplified version of the above equations will be used. It will be assumed that the process is governed by a linear equation:

$$X_k = X_{k-1} + w_k \tag{5-6}$$

With a measurement equation:

$$Z_k = X_k + v_k \tag{5-7}$$

And Hence the Kalman filter prediction phase equation can be rewritten as:

$$\hat{X}_k^- = \hat{X}_{k-1} \tag{5-8}$$

$$P_k^- = P_{k-1} + Q \tag{5-9}$$

And the measurement update equations are:

$$K_k = P_k^-(P_k^- + R)^{-1} = \frac{P_k^-}{P_k^- + R} \tag{5-10}$$

$$\hat{X}_k = \hat{X}_k^- + K_k\left(Z_k - \hat{X}_k^-\right) \tag{5-11}$$

$$P_k = (1 - K_k)P_k^- \tag{5-12}$$

It is assumed that the process has a very small variance Q=1e^{-5} (for filter tuning flexibility). The initial seed for the filter, X_{k-l} will be assumed to be zero. Similarly the initial value for P_{k-1}, which is called P_0 will be any value but not equal to zero. The measurement variance R will be initially assumed very large number in order to express the uncertainty in the measurement accuracy.

5.7 Filters Performance Comparison Study

To compare the performance of the above mentioned three filters, they were applied to the collected RSSI raw data. The goal of this comparison is to determine which filtering algorithm is preferable for improving indoor localization accuracy. The performance of these filter was measured based on local and global measures. The local measures consists of four matrices: (1) comparison between the original signal and filtered signal in terms of shape and height of waveform peaks, (2) mean and standard deviation, (3) Signal-to-Noise Ratio (SNR), and (4) Correlation coefficient (R) between original and filtered signals. The global measure is the absolute distance estimation error.

The three filters settings were selected to make closest possible performance as following:

- Moving average filter with 9 observations window.
- Savitzky-Golary filter with 9 observations window.
- Kalman filter with process error equal 1e^{-5}; and measurement error equal 0.15.

5.7.1 MATLAB Simulation

In order to generate RSSI signals in MATLAB, the collected RSSI data in chapter 4 was plotted versus the distance as shown in Figure 5-17, then the RSSI average and standard deviation were calculated as shown in Figure 5-18 and Figure 5-19. Two equation were generated using regression to represent the RSSI mean and standard deviation with respect to distance:

$$RSSI_{Average} = -7.635 \ln(distance) - 40.635 \quad (5\text{-}33)$$

$$RSSI_{StdDev} = 0.5658 \ln(distance) + 4.0113 \quad (5\text{-}34)$$

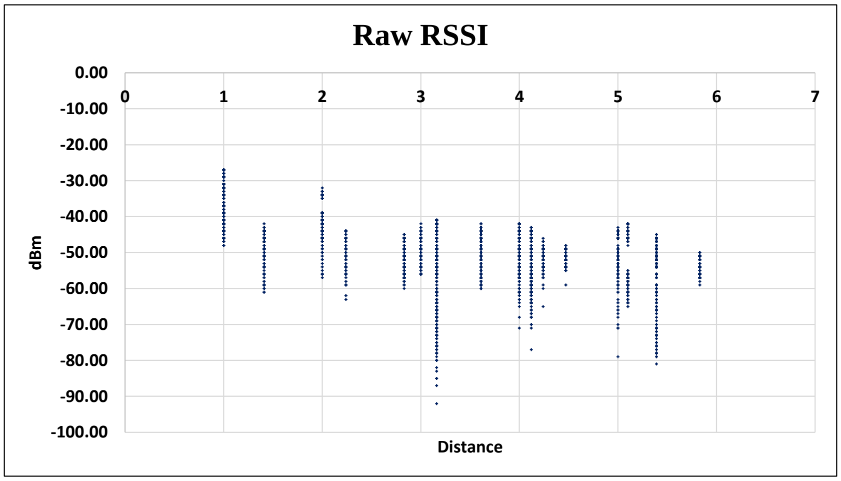


Figure 5-17: Raw RSSI Vs Distance

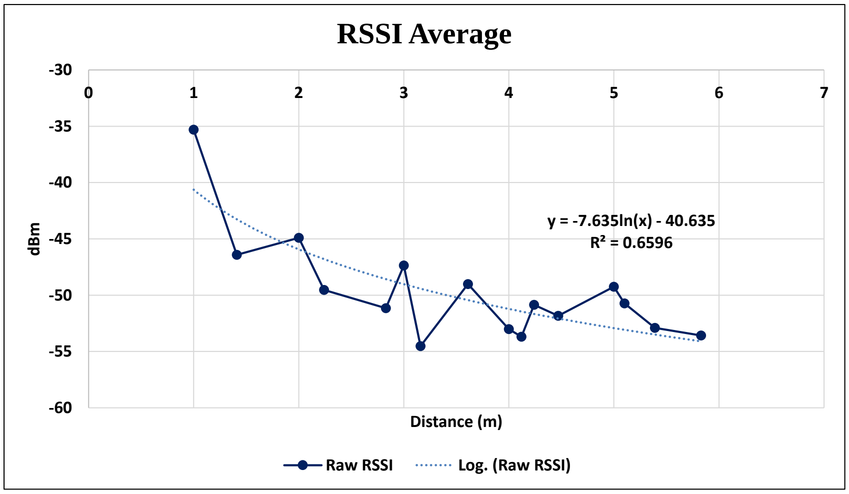


Figure 5-18: Raw RSSI Average

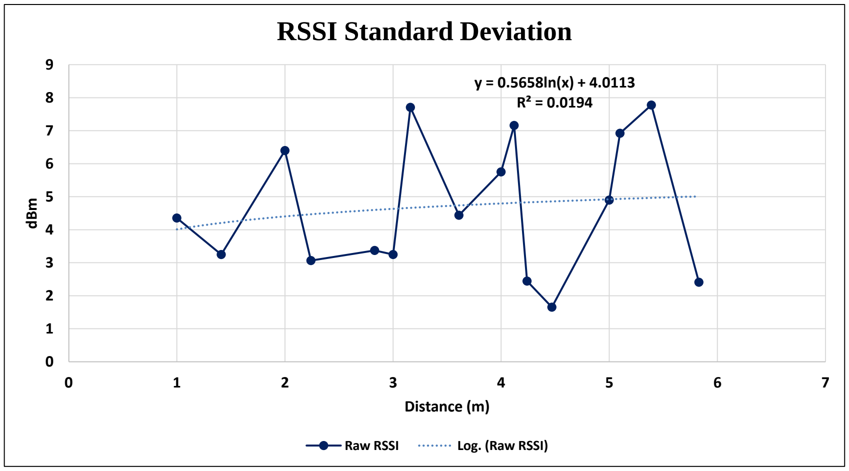


Figure 5-19: Raw RSSI Standard Deviation

The two generated equations (5-33) and (5-34) are used to build the path loss model and generate RSSI signals in the simulator. The second step of the simulator test is to validate the RSSI filtering techniques, where the moving average filtering was compared to Savitzky-Golary filter and Kalman filtering technique. The distance estimation errors were calculated using each filter and presented in the following figures.

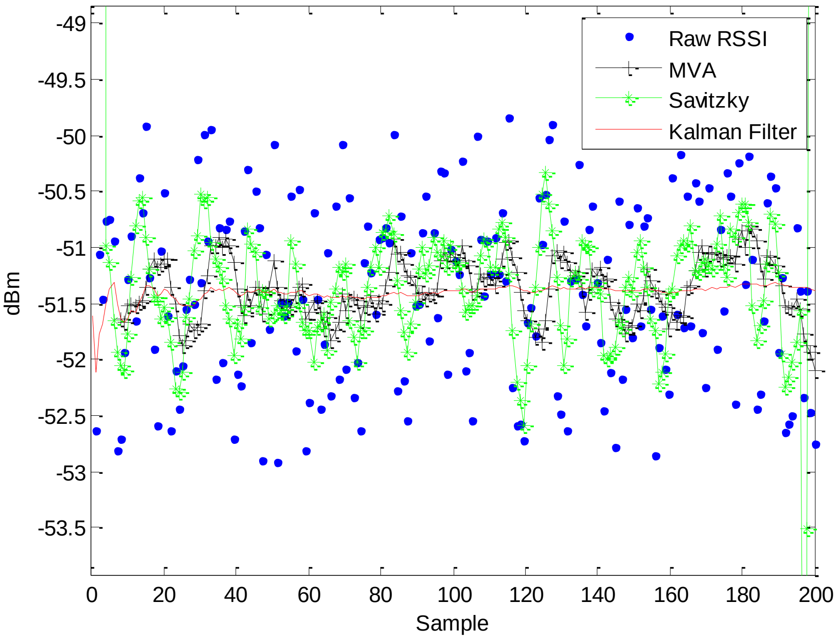


Figure 5-20: RSSI Signal Filtering at Distance 5 m

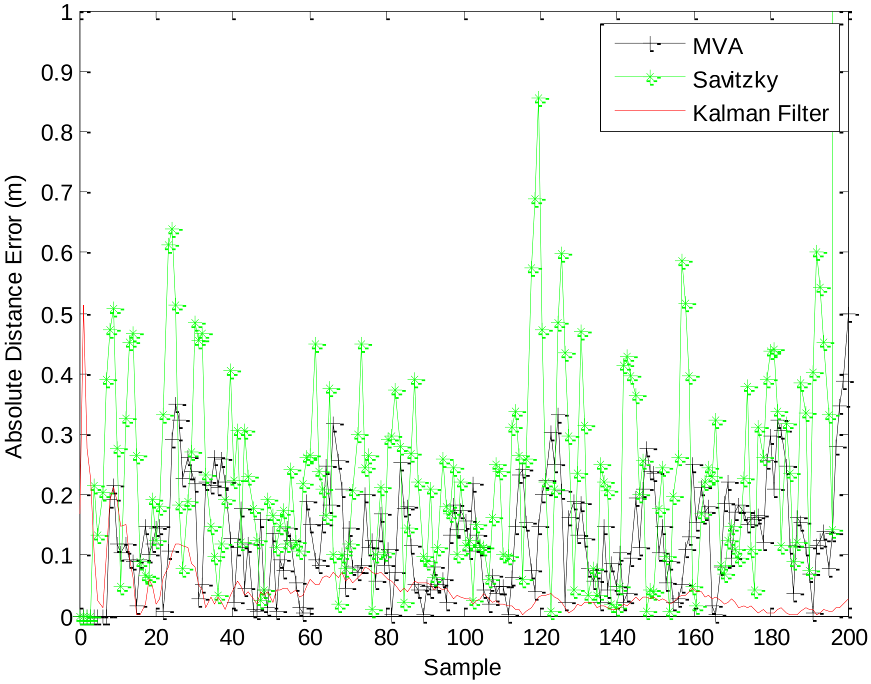


Figure 5-21: Absolute Distance Estimation Error at 5 m

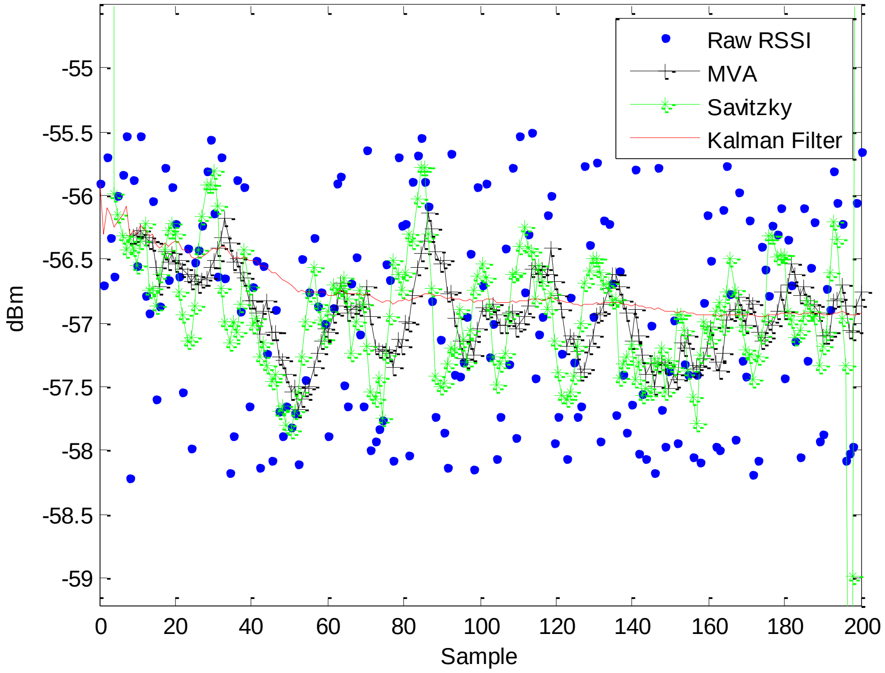


Figure 5-22: RSSI Signal Filtering at Distance 10 m

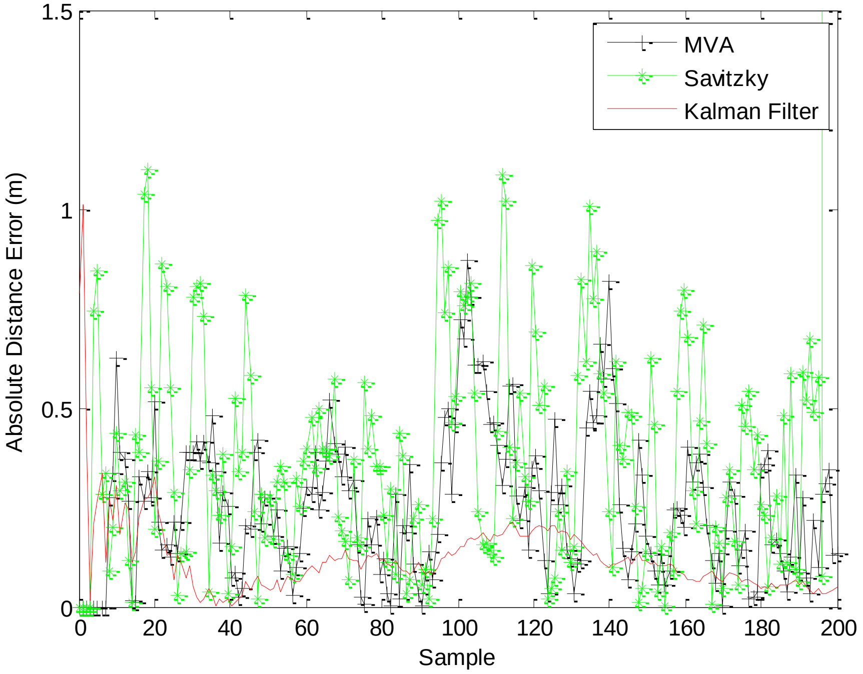


Figure 5-23: Absolute Distance Estimation Error at 10 m

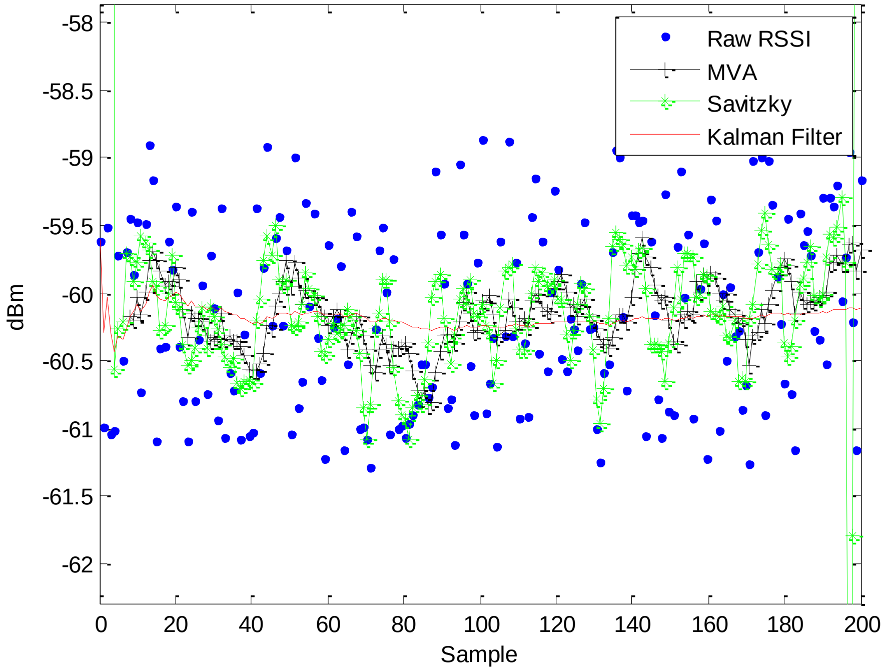


Figure 5-24: RSSI Signal Filtering at Distance 15 m

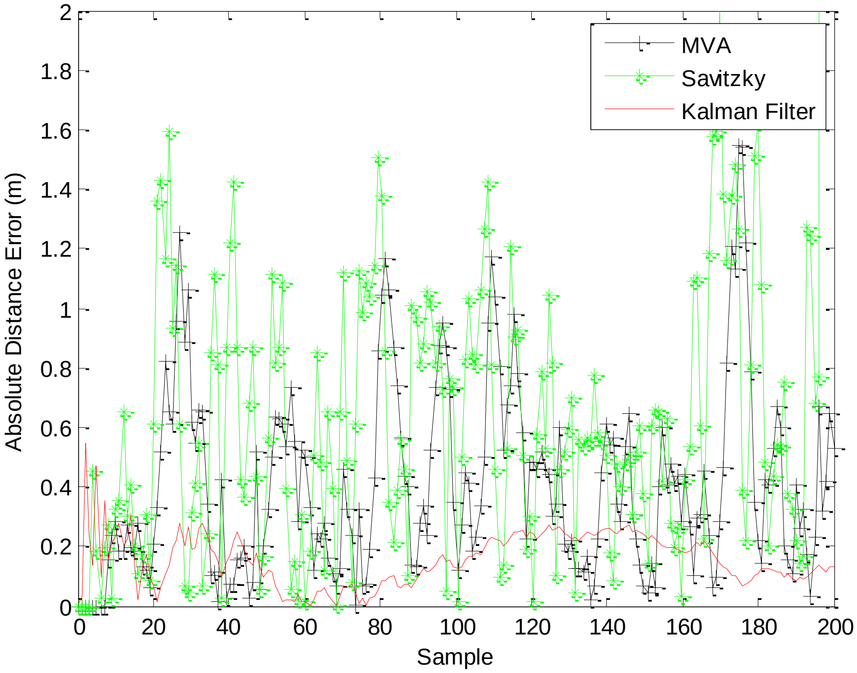


Figure 5-25: Absolute Distance Estimation Error at 15 m

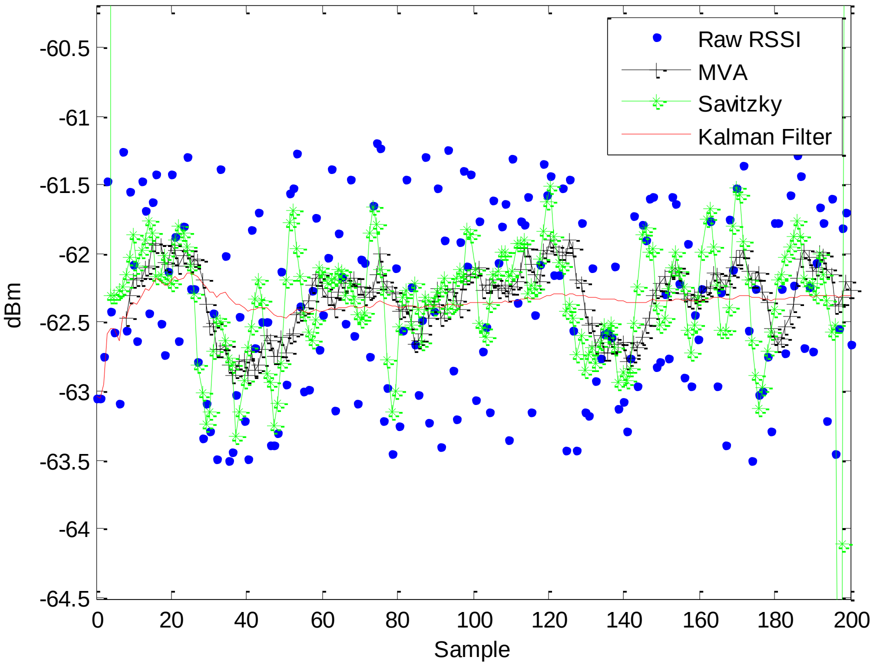


Figure 5-26: RSSI Signal Filtering at Distance 20 m

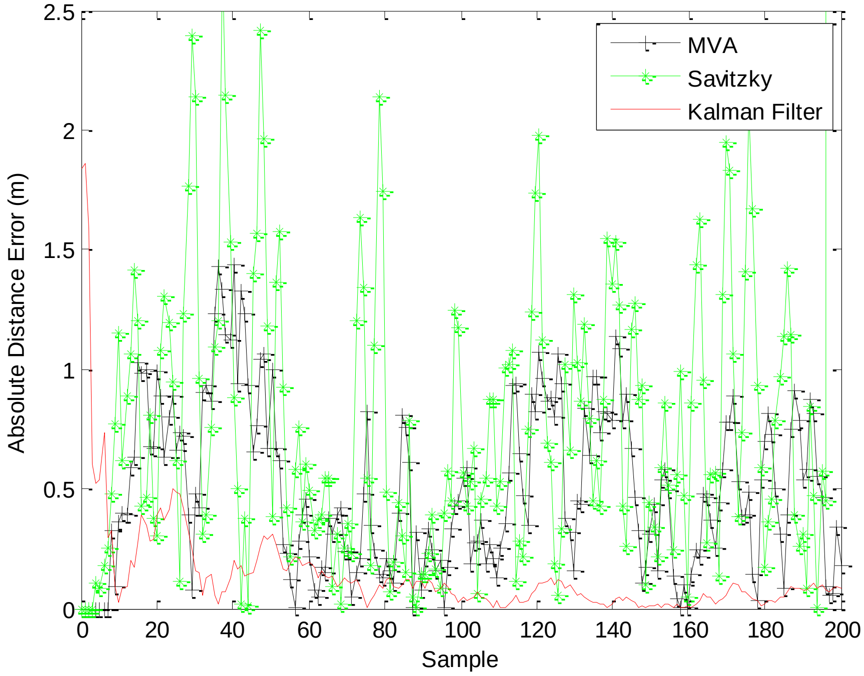


Figure 5-27: Absolute Distance Estimation Error at 20 m

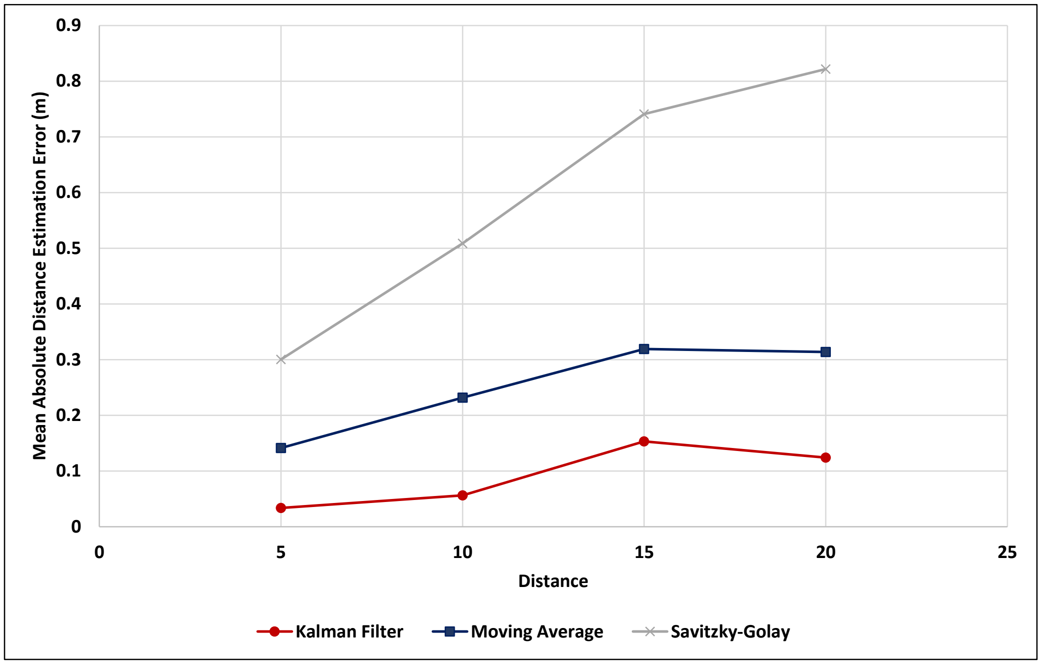


Figure 5-28: Mean Absolute Distance Estimation

The results indicated the low mean absolute distance estimation error using the Kalman filter in comparison to the moving average and Savitzky-Golay filters.

5.7.2 Laboratory Experiments

Figure 5-29 present the original RSSI signal and the signal after applying the three filters. By comparing original and filtered signals, it is obvious that moving average filter did not remove coherent noise from the original signal. For example noise in time interval between the samples 5498 to 5766 as shown in Figure 5-30.

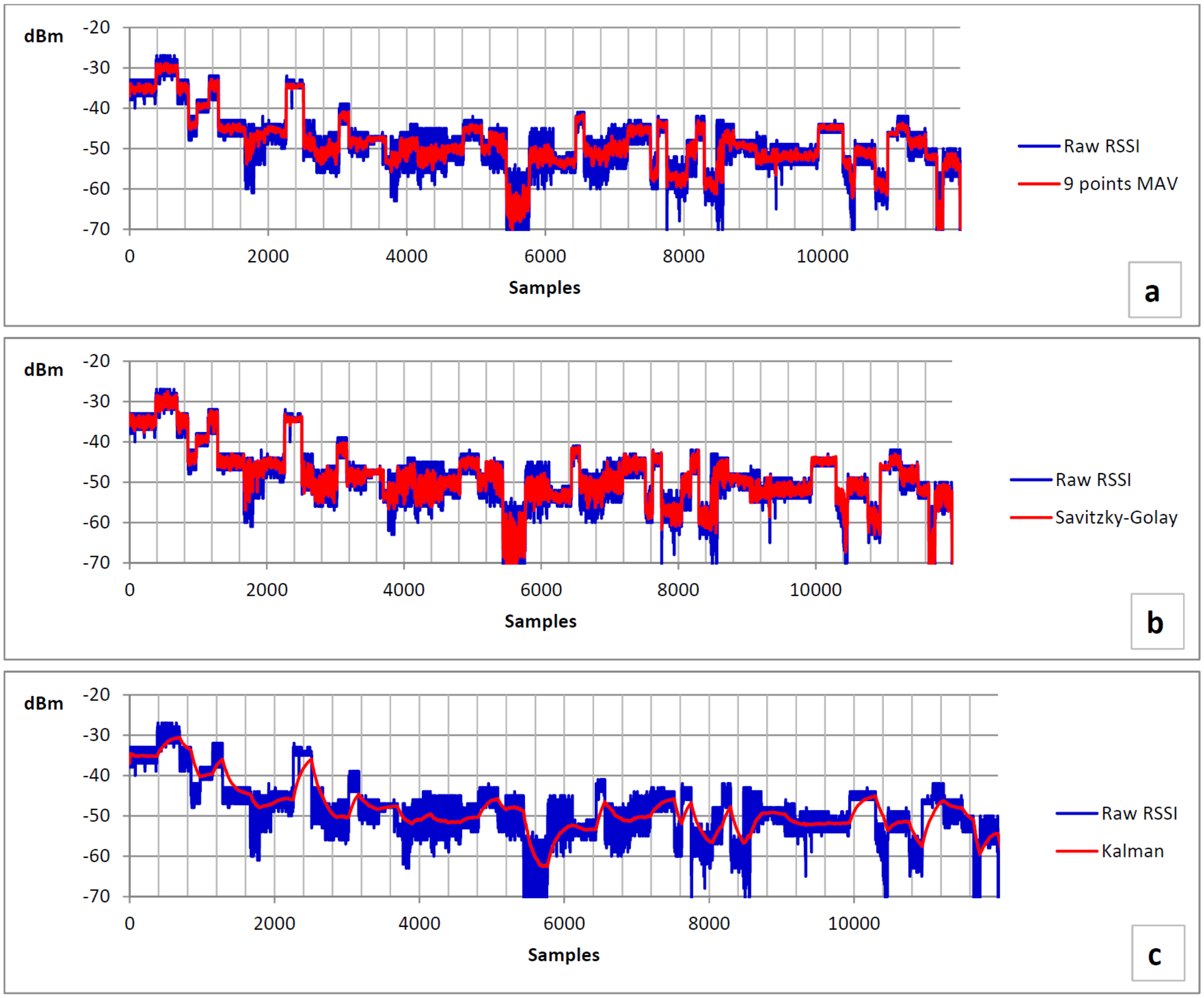


Figure 5-29: Raw RSSI Vs Filtered RSSI

The Kalman filter performed better in clearing the original signal from the underlying noise, while providing fast convergence for on-line estimations. The filter perfectly removed most of uncorrelated noise (such as small variations and narrow spikes) and reduced bias components. Meanwhile, the Savitzky-Golary resulted in a significantly worse performance. The smoothing effects of the moving average and Savitzky-Golay filters is less aggressive than the Kalman filter and the distortion is comparatively limited. However, it should be stressed that both the first mentioned filters are providing signal smoothing with only cosmetic value.

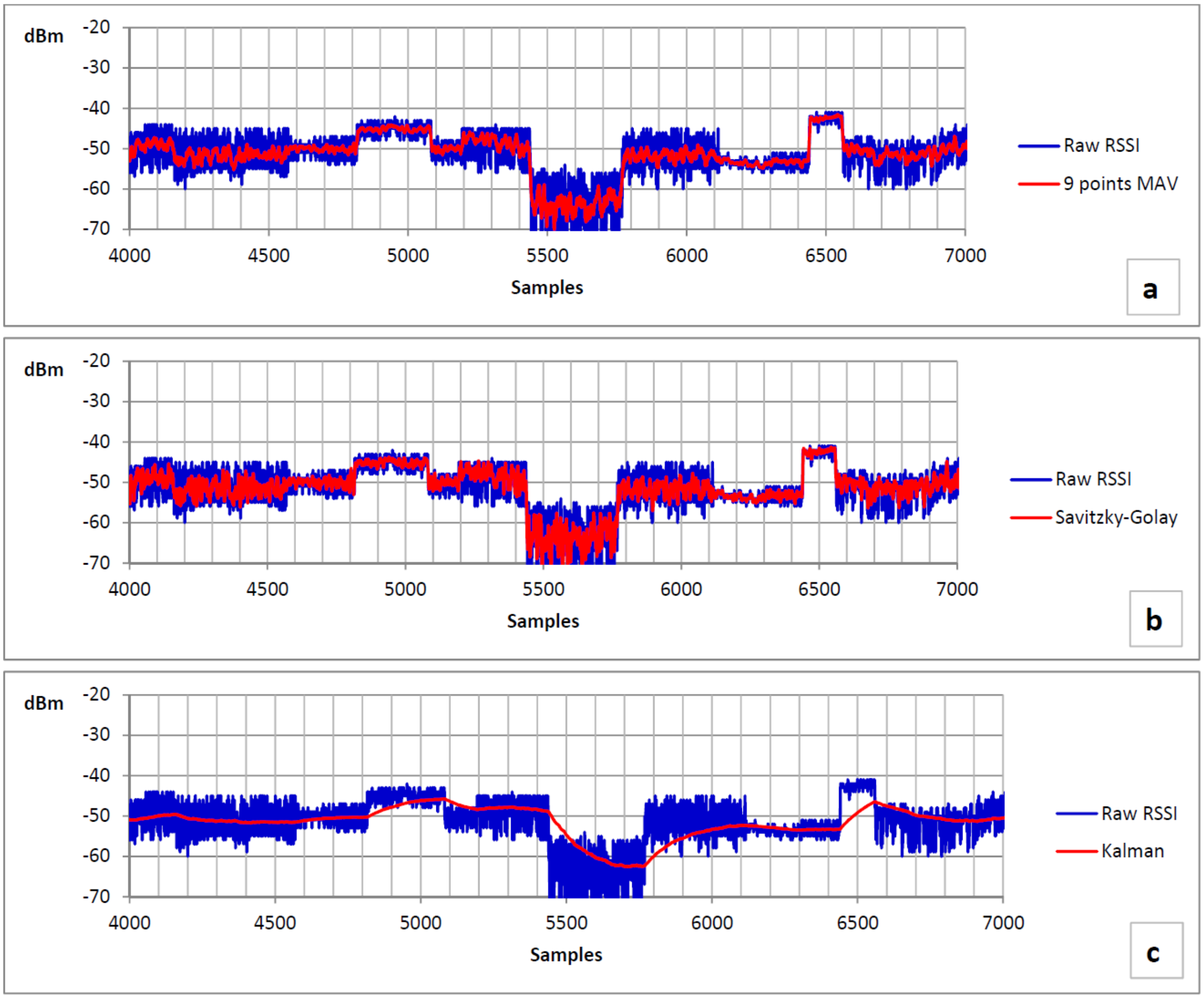


Figure 5-30: Coherent Noise in Raw RSSI Vs Filtered RSSI

In order to establish the validity of the above results, it is important to quantify the uncertainty in filter performance. Statistical uncertainties in the measured signal should be reduced by filtering out uncorrelated noise from measured signal. Figure 5-31 compares the original signal probability distribution and the filtered signals distribution at different distances.

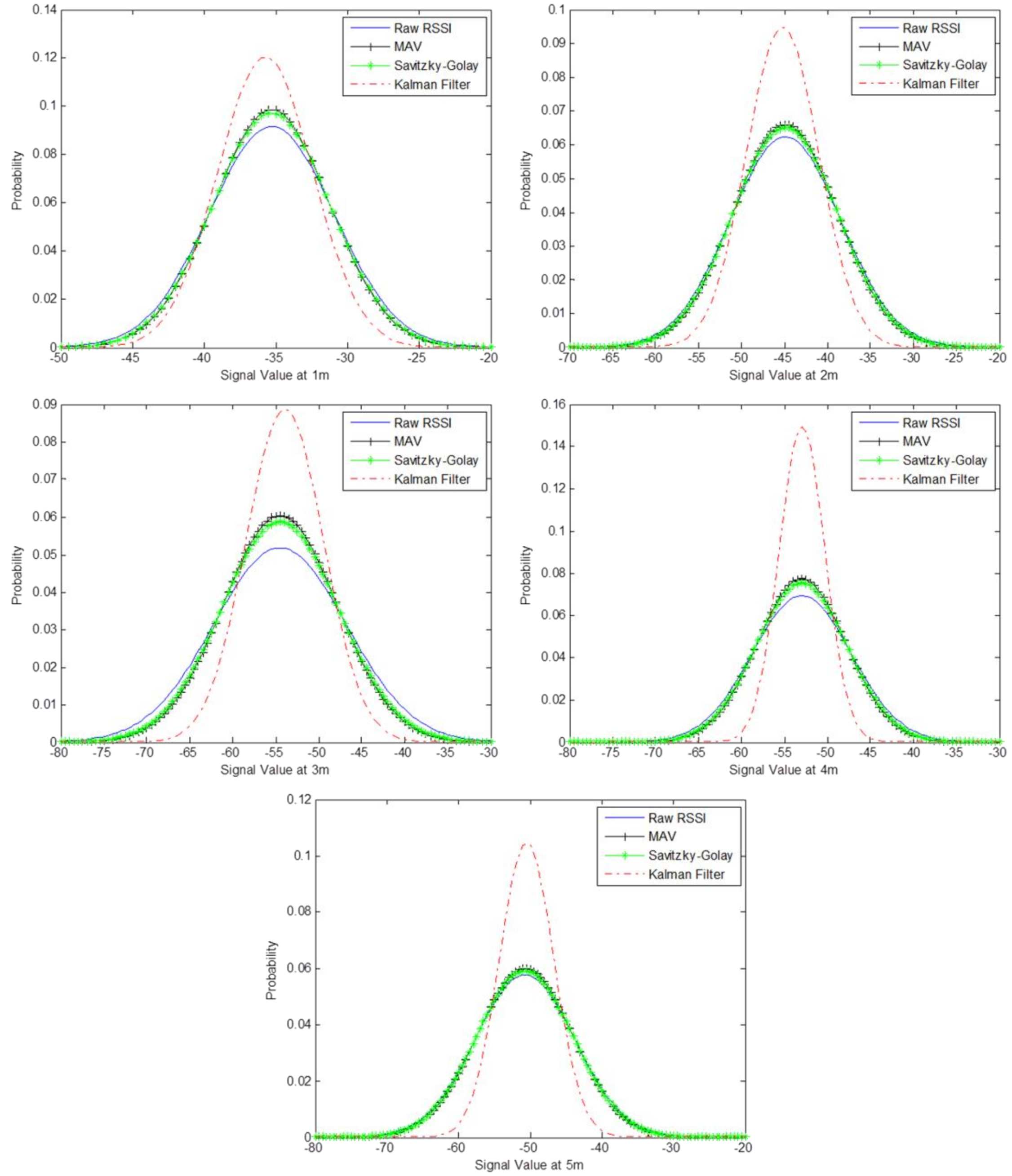


Figure 5-31: Original and Filtered Signals Probability Distribution at Different Distances

Given that all filters has the same number of observation samples, a smaller standard deviation of the filter output, means better filter performance. The Kalman filter reduced variance in the filtered signal, which in turn enhanced the confidence and lowered the uncertainty in the RSSI value at all distances as shown in Table 5-2, Figure 5-32 and Figure 5-33.

Table 5-2: Filtered Signals Statistics

distance	Raw RSSI		MAV		Savitzky-Golay		Kalman	
	Average	StdDev	Average	StdDev	Average	StdDev	Average	StdDev
1	-35.29	4.35	-35.30	4.05	-35.30	4.10	-35.74	3.31
2	-44.90	6.40	-44.91	6.04	-44.90	6.12	-45.22	4.21
3	-54.52	7.71	-54.51	6.60	-54.52	6.79	-53.96	4.49
4	-53.01	5.75	-53.02	5.17	-53.02	5.30	-52.97	2.67
5	-50.74	6.92	-50.73	6.65	-50.74	6.73	-50.50	3.82

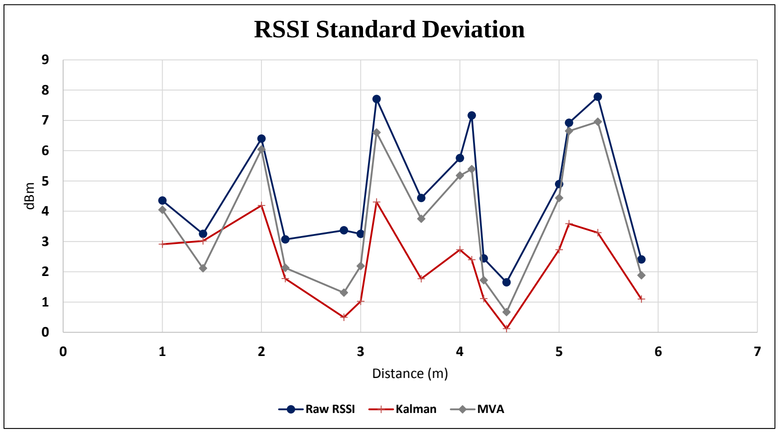


Figure 5-32: Filtered Signals Standard Deviation

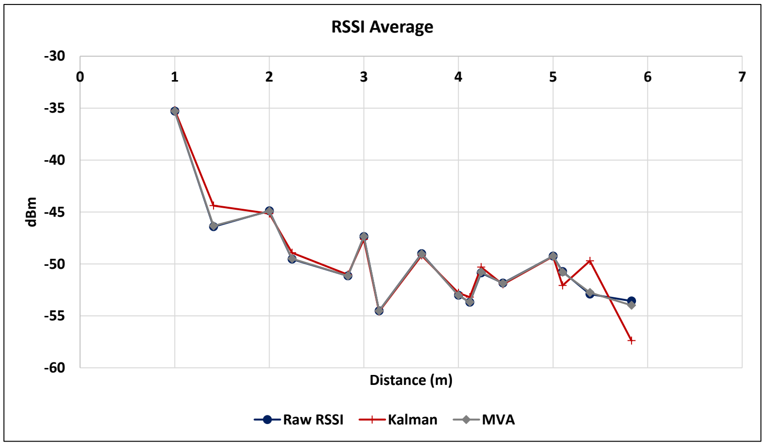


Figure 5-33: Filtered Signals Average

Signal-to-noise ratio (SNR) is a measure used in electromagnetic engineering to estimate the level of a measured signal to the level of background noise. The higher the SNR indicates more signal than noise. Several calculation methods can be used for calculating SNR. In this research the SNR is calculated as the ratio of mean to standard deviation of the RSSI signal

$$SNR = \frac{\mu}{\sigma} \tag{5-13}$$

Where μ is the signal mean and σ is the standard deviation of the signal.

The raw RSSI signal had an average SNR of 7.75 dB, after applying the filters to the RSSI; the Kalman filter has an average SNR of 13.33 dB. In comparison to the moving average and Savitzky-Golay, which had an average SNR of 8.46 and 8.30 dB respectively as shown in Figure 5-34.

The final test for the applied filters is the coefficient of correlation (R), which reflects the degree of linear relationship between two sets of data. It has a value between -1 and +1. A value of +1 means that there is a perfect positive linear relationship between the two data sets. A value of -1 means that there is a perfect negative linear relationship, and a value of 0 means there is no linear relationship at all between the data sets. The R is calculated as follows:

$$R = \frac{\sum_{n=0}^{n-1} X(n)Y(n)}{\sqrt{\left(\sum_{n=0}^{n-1} X^2(n) \sum_{n=0}^{n-1} Y^2(n)\right)}} \tag{5-14}$$

Where X(n) and Y(n) refer to the filtered signal and reference signal respectively.

Table 5-3, presents the calculated R for each filter. All filters show good performance with respect to correlation to the original signal, with a slight better performance for Savitzky-Golay filter.

Table 5-3: Filters Coefficient of Correlation (R)

	Moving Average	**Savitzky-Golay**	**Kalman Filter**
Coefficient of Correlation (R)	0.997 ± 0.001	0.999 ± 0.001	0.996 ± 0.002

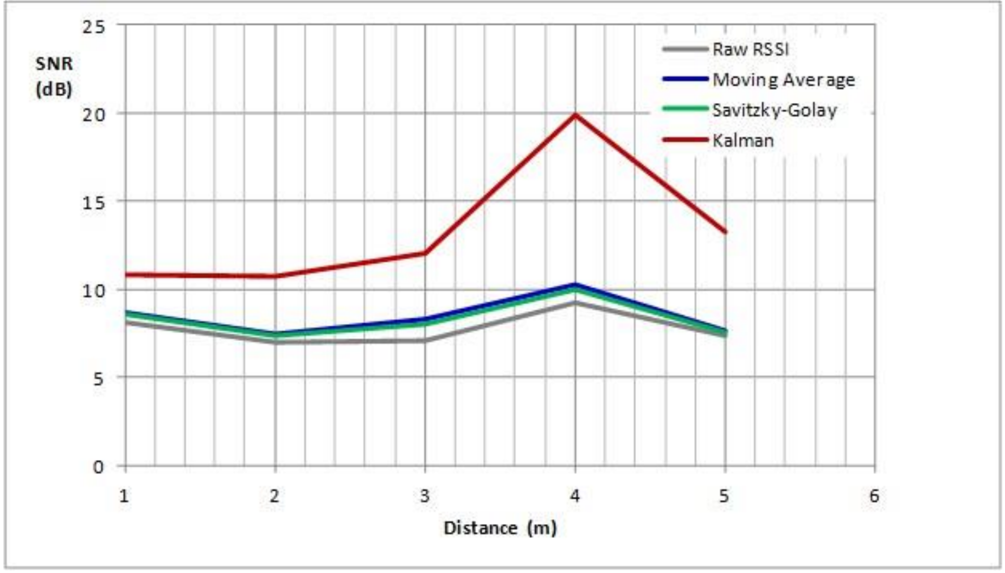


Figure 5-34: Signal to Noise Ratio at Different Distances

In conclusion of the above mentioned results, it is clear that the Kalman filter outperformed other filters in removing the noise, and provided an output filtered signal with less variations. The result obtained using Kalman filter is less noisy, consistent and reliable compared with the other filters presented above.

5.8 Prototype Software/Algorithms

The developed SC-WSN is composed of two stages, prediction stage and calibration stage as shown in Figure 5-35. Each reader measures the signal strength (RSSI) from nearby tags and filter it to remove uncorrelated noise. Then the filtered RSSI data is converted to its corresponding distance using a newly developed dynamic path-loss model, and forwarded to the tag again for location calculation. Based on three distances from three readers for a given tag, its location is estimated using the LSE trilateration algorithm (Karl & Willig, 2007). The localization accuracy is continually monitored by measuring the errors in location estimation generated based on a number of reference tags. These reference tags are installed on site at pre-defined locations. When the system accuracy is degraded due to on-site interferences, a system calibration request is initialized. The threshold of an acceptable error was set to 20% in this research, which defines the accuracy limits to initiate the calibration requests. Finally, at the end of the calibration stage, the dynamic signal propagation model is updated with new set of optimized parameters.

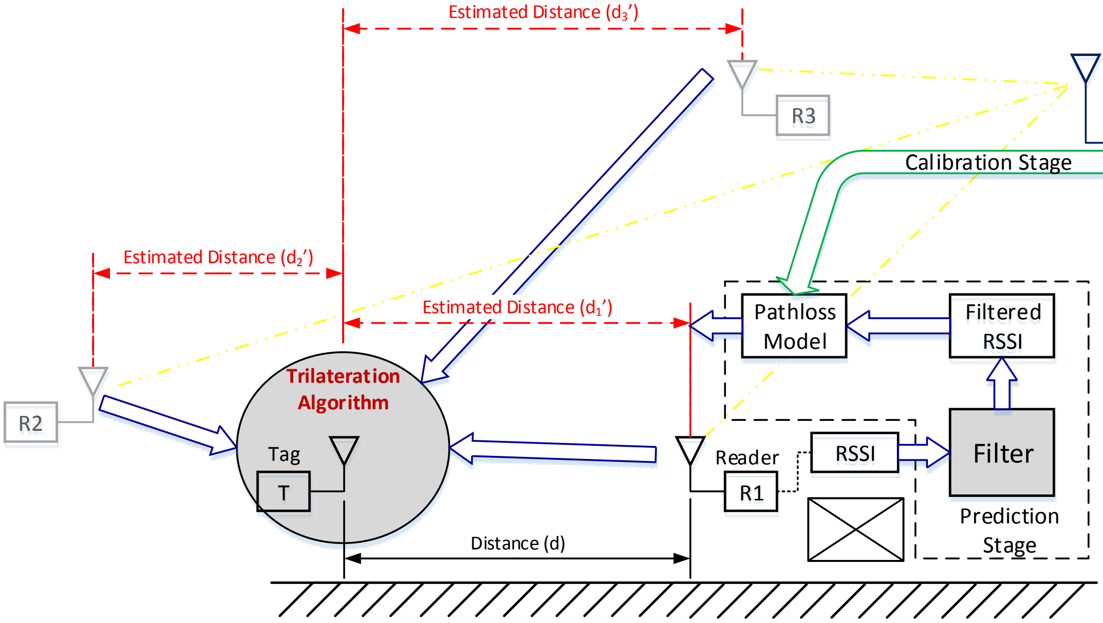


Figure 5-35: Developed SC-WSN Localization Overview

5.8.1 Dynamic Path-Loss Model

Free-space path-loss models are not suitable for indoor localization in real world environment due to the presence of shadow fading and multipath effects. It is important to investigate signal propagation in real situations in order to design a more realistic path-loss model which is able to handle uncertainties and noise in RSSI measurements. In the following section, real signal propagation scenarios are analyzed in order to provide solutions for indoor localization in construction jobsites environment.

Both theoretical and measurement based propagation models indicate that average received signal power decreases logarithmically with distance. Empirical models help in reducing computational complexity as well as increasing the accuracy of the predictions. The empirical model used in this study is Log-distance Path Loss Model. The path loss PL (d) for a transmitter and a receiver with distance d is:

$$PL(d) \propto \left(\frac{d}{d_0}\right)^n \tag{5-15}$$

$$PL(dB) = PL(d_0) + 10\, n\, ln\left(\frac{d}{d_0}\right) + \sigma^2 \tag{5-16}$$

Where n is the path loss exponent which indicates the rate at which path loss increases with distance d. The reference distance (d_0) is determined from measurements at 1 meter distance from the transmitter. σ^2 is the shadowing variance in mdB.

The value of n depends on the specific propagation environment, i.e., type of construction material, architecture, location within building. Table 5-4 lists typical path loss exponents obtained in various radio environments (Rappaport, 1996).

Table 5-4: Path Loss Exponents for Different Environments (Rappaport, 1996)

Environment	**Path Loss Exponent, n**
Free Space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of sight	1.6 to 1.8
Obstructed in buildings	4 to 6
Obstructed in factories	2 to 3

Given that d_0 = 1 m, Equation (5-16) can be simplified as:

$$PL(dB) = A + B\ ln(d) \tag{5-17}$$

Where A & B are the parameters for the signal propagation model. In this research, the signal propagation model parameters (A&B) are estimated using lab experiments, then automatically adjusted on-site using the Particle Swarm Optimization.

The Kalman filtered signal collected in laboratory experiments were used to generate the RSSI propagation model, which is used for distance estimation and hence location estimation. Figure 5-36 illustrates the Kalman filtered RSSI with respect to the actual distance between the transmitter and receiver nodes. Least square method is used to fit this relation in exponential equation format:

$$d = e^{\left(\frac{RSSI - A}{B}\right)} \tag{5-18}$$

Where A & B are constant confidents and d is the distance between the transmitter and receiver nodes. From the Figure 5-36, the distance can be estimated as:

$$d = e^{\left(\frac{RSSI_{filtered} + 38.909}{-8.989}\right)} \tag{5-19}$$

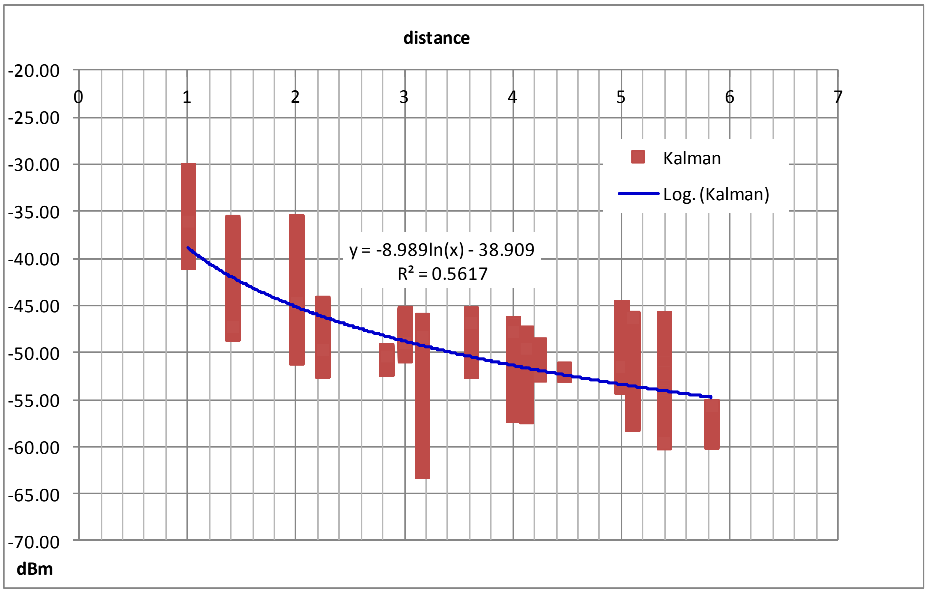


Figure 5-36: Kalman Filtered RSSI Vs Actual Distance between Tx And Rx Nodes

5.8.2 Positioning Algorithm

The multilateration (Karl & Willig, 2007) algorithm is applied, where the position of an object (tag) is calculated based on its estimated distances from fixed location devices (readers) (Stüber & Caffrey, 1999). When three readers are used, it is called trilateration, as shown in Figure 5-37.

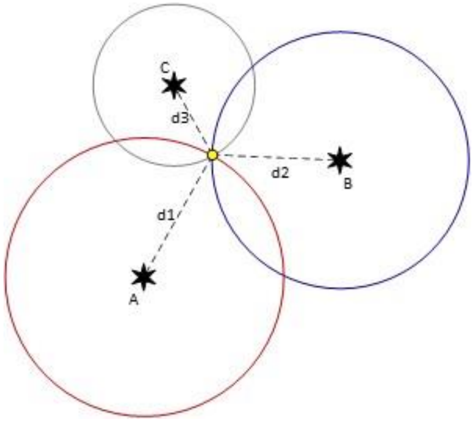


Figure 5-37: Localization using Trilateration

The intersection of the three circles gives an exact solution for the tag's location under ideal free space signal propagation (no fading or shadowing effect). However, in real environment, the three circles might not even intersect due to errors in distance estimates by RSSI path-loss model. (Stüber & Caffrey, 1999) presented an optimal localization using least square estimation (LSE) as shown in Figure 5-38.

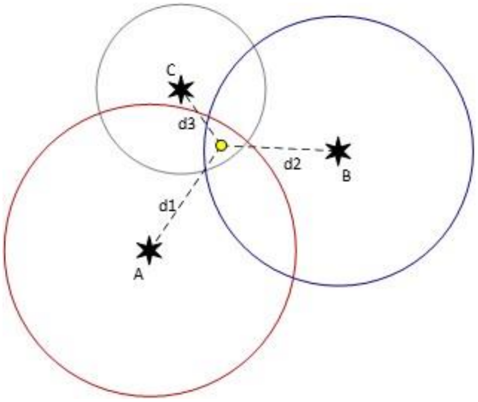


Figure 5-38: Localization using Least Square Estimation Trilateration

Given the readers coordinates as following: A(x_1, y_1), B(x_2, y_2), and C(x_3, y_3); and their corresponding distances to the tag are d_1, d_2, and d_3; the three circles equations can be formatted as following:

$$(x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \tag{5-20}$$

$$(x_2 - x)^2 + (y_2 - y)^2 = d_2^2 \tag{5-21}$$

$$(x_3 - x)^2 + (y_3 - y)^2 = d_3^2 \tag{5-22}$$

Where x and y are the coordinates for the tag location.

Using the LSE method the tag coordinates can be calculated using the following equation:

$$2\begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ (d_2^2 - d_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \end{bmatrix} \tag{5-23}$$

5.8.3 Self-Calibration Algorithm

In order to improve the positioning accuracy in indoor location algorithm based on RSSI, the developed prototype monitors the indoor propagation conditions and adapts its parameters when necessary (Ibrahim & Moselhi, 2015c). Static lognormal path-loss models are commonly utilized, where their parameters are calculated during an offline measurement phase before the deployment (Razavi & Moselhi, 2012). A disadvantage of this approach is its inability to adapt to different environments.

The developed self-calibration algorithm relies on periodic calculation of reference tags locations and computation of system accuracy, and then find near optimum value for the path-loss model parameter in order to minimize the Mean Absolute location error.

$$Z = min\left(\frac{1}{m}\sum_{i=1}^{m}\left|d_{actual_i} - d_{measured_i}\right|\right) \tag{5-24}$$

This objective function is non-linear which does not allow for an exact solution. Evolutionary optimization methods such as Genetic Algorithm (GA) or Particle Swarm Optimization (PSO) can be used to solve these kind of problems. PSO has the same effectiveness (finding the true global optimal solution) as the GA but with significantly better computational efficiency (Hassan, Cohanim, de Weck, & Venter, 2005). The computational efficiency of the selected optimization algorithm is very important for the developed method due to the limited computational resources of the microcontroller (memory and speed). Therefore it was used in the developed prototype.

In the context of path-loss model optimization, each particle consists of two members: the coefficient A and B of the path loss model as shown in Figure 5-39. In this problem space, each particle keeps track of its A and B values, which are associated with the best solution (fitness) it has achieved so far. This value is called pBest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gBest. The developed PSO algorithm for optimizing the solution is formulated with the following procedures:

- Step 1: Measure the RSSIs between the reference nodes and readers.
- Step 2: Generate a random population of N particles, a random initial set of velocities, pBest and gBest. Each particle has 2 dimensions which are A and B.
- Step 3: For each particle of the population, compute d from Equation (5-18):

$$d_{measured} = e^{\left(\frac{RSSI-A}{B}\right)} \tag{5-25}$$

- Step 4: Determine the fitness function, which is the Mean Absolute distance error for m readings collected from readers:

$$|d_{error}| = |d_{actual} - d_{measured}| = \left|d_{actual} - e^{\left(\frac{RSSI-A}{B}\right)}\right| \tag{5-26}$$

$$\overline{|d_{error}|} = \frac{1}{m}\sum_{i=1}^{m}\left|d_{actual_i} - e^{\left(\frac{RSSI_i-A}{B}\right)}\right| \tag{5-27}$$

- Step 5: Repeat steps 4 and 5 for all the N particles in the population.
- Step 6: Compare each particle’s fitness with its pBest. If it is better, then its value and pBest are updated. The best fitness value among the pBest is used as the gBest.
- Step 7: Let c1 and c2 be two constants. Set the member velocity v of each particle as follows:

$$v_{i,A}(k+1) = \omega_{i,A}(k) + c_1\lambda_1.\left(pBest_i - A_i(k)\right) + c_2\lambda_2.\left(gBest_A - A_i(k)\right) \tag{5-28}$$

$$v_{i,B}(k+1) = \omega_{i,B}(k) + c_1\lambda_1.\left(pBest_i - B_i(k)\right) + c_2\lambda_2.\left(gBest_B - B_i(k)\right) \tag{5-29}$$

where i =1,2,….., N. $v_{i,A}$, $v_{i,B}$ represents the velocity of the ith particle for A and B , and ω is the inertia weight that controls the exploration of local and global search space. λ_1 and λ_2 are random numbers between 0 and 1. The constants c_1 and c_2 are also known as the acceleration constants. They are the cognition and social components respectively that determine the speed a particle would accelerate towards the pBest and the gBest. In this research, the inertia weight is varied linearly according to the following equation:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{k_{max}} \times k \tag{5-30}$$

Where k is the current iteration number and k_{max} is the maximum iteration number.

- Step 8: The particle position is modified by adding the new velocity to the existing particle position as follows:

$$A_i(k+1) = A_i(k) + v_{i,A}(k+1) \tag{5-31}$$

$$B_i(k+1) = B_i(k) + v_{i,B}(k+1) \tag{5-32}$$

- Step 9: If the number of the iteration is less than the maximum number of iterations and the stopping criterion is not met, repeat step 3. Otherwise, proceed to step 10.
- Step 10: The resulting gBest gives the near optimum coefficient A, B.

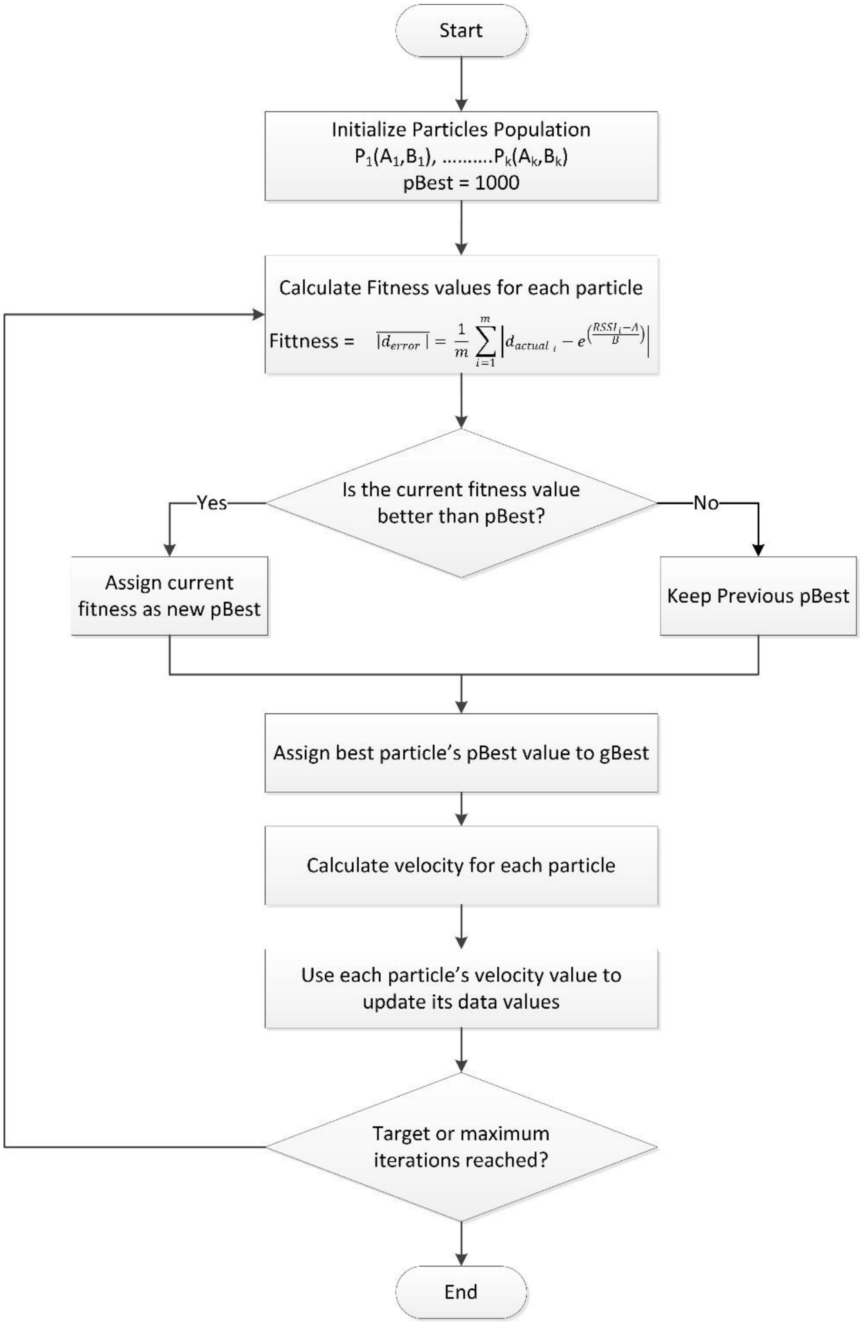

Start
Initialize Particles Population
$P_1(A_1,B_1)$, ……….$P_k(A_k,B_k)$
pBest = 1000
Calculate Fitness values for each particle
Fittness = $\overline{|d_{error}|} = \frac{1}{m}\sum_{i=1}^{m}\left|d_{actual\ i} - e^{\left(\frac{RSSI_i - A}{B}\right)}\right|$
Is the current fitness value better than pBest?
Yes
No
Assign current fitness as new pBest
Keep Previous pBest
Assign best particle's pBest value to gBest
Calculate velocity for each particle
Use each particle's velocity value to update its data values
Target or maximum iterations reached?
End

Figure 5-39: Path Loss Model Parameters Optimization using PSO

5.9 Prototype Validation

The validation of this prototype was conducted in two phases. Phase one, MATLAB simulation. The prototype design was fine-tuned based on simulation results. Then, phase two, where laboratory experiments for the final design of the prototype were conducted to validate simulation results.

5.9.1 Phase One: MATLAB Simulator Testing

MATLAB simulator is programmed to mimic the indoor localization in lab environment according to the collected data in phase one. Three simulation tests were performed. First, localization using raw RSSI signal, this simulation is intended to calculate the expected localization accuracy using the raw RSSI signal. Second, localization with Kalman filtered RSSI signal, the purpose of this simulation is to calculate expected localization accuracy using the Kalman filtered RSSI signal in order to evaluate it is applicability. Third, the self-calibration algorithm using PSO is tested to validate its capability in reducing the localization error when the environment changes or the signal interference increases.

The localization with raw RSSI signal was tested by generating 100 random tags in a space 20 m x 20 m, with 3 readers. The RSSI was generated using equations (5-33) and (5-34). The localization error was measured as the Euclidian distance between the actual tag location and the estimated tag location using the following equation:

$$Location_{error} = \sqrt{(X-a)^2 + (Y-b)^2} \tag{5-35}$$

Where: (X, Y) is the actual tag location, and (a, b) is the estimated tag location.

The mean location error was 2.547 meters and a standard deviation of 1.841 meter. Figure 5-40 illustrated the actual tags location verses the estimated tags location using raw RSSI. The cumulative distribution function of the localization error in Figure 5-41 illustrates 5.574 m error with 95% confidence.

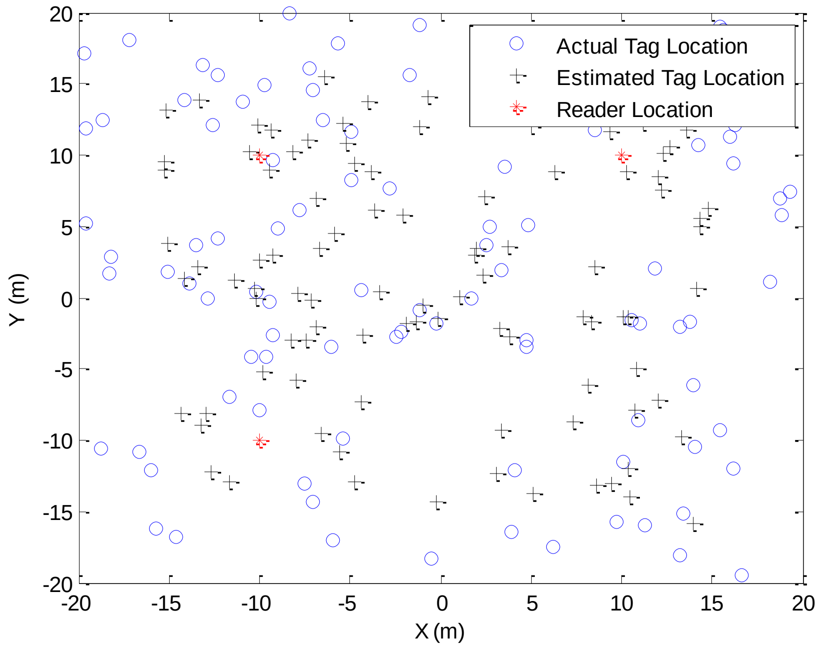


Figure 5-40: Localization using Raw RSSI

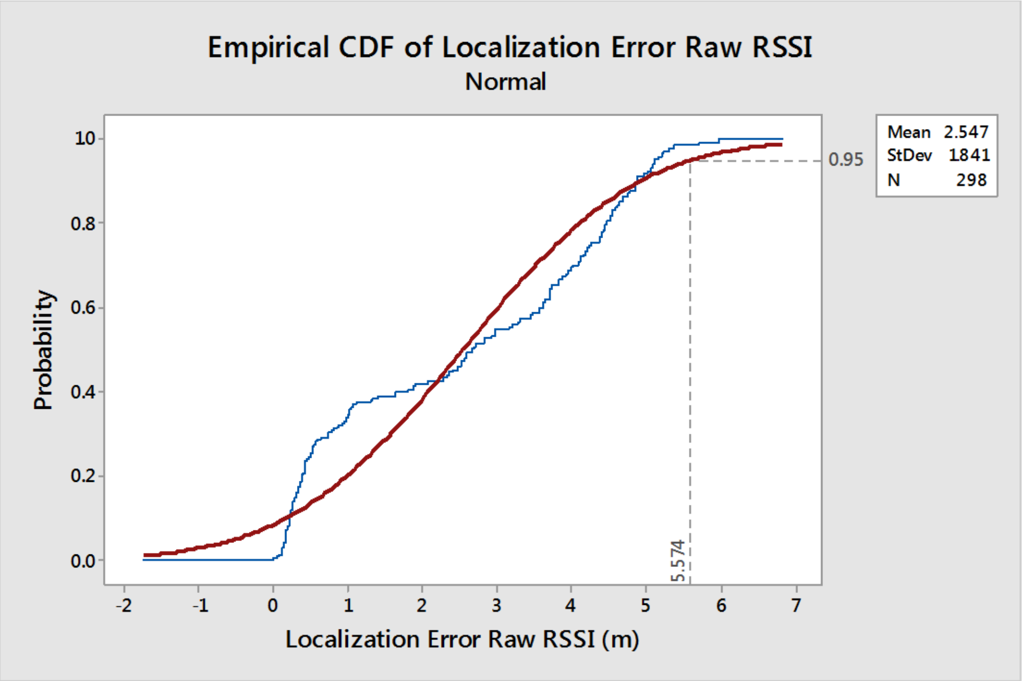


Figure 5-41: Localization Error CDF using Raw RSSI

It is clear from the above figures that the Kalman filter is outperforming the commonly used moving average filter in de-noising the RSSI, where it provide very stable signal that can be used in distance estimation with higher accuracy.

The distance was ranged from 1 to 100 meter and the mean Kalman filtered RSSI signal was calculated and plotted against the distance as shown in Figure 5-42. Regression was used to derive the equation for the path loss model as following:

$$RSSI_{Filtered} = -7.935 \ln(distance) - 38.57 \tag{5-36}$$

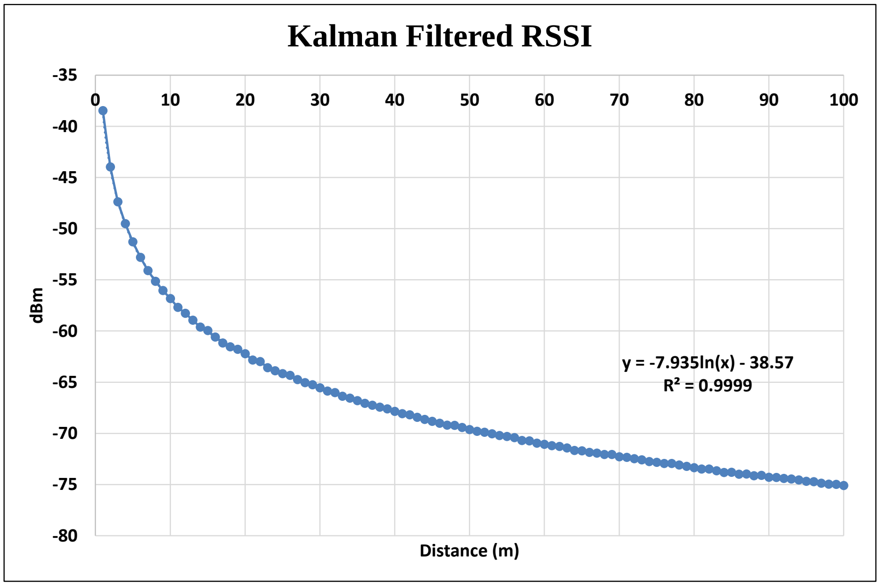


Figure 5-42: Kalman Filtered RSSI Signal Vs Distance

The third step is to test the localization accuracy using the Kalman filtered RSSI. 100 random tags were generated in a space 20 m x 20 m, with 3 readers. The RSSI was generated using equations (5-33) and (5-34). The localization error was calculated using Equation (5-35) and the tag's distances from readers were estimated using the Kalman filtered RSSI path-loss model Equation (5-36). Figure 5-43 illustrated the actual tags location verses the estimated tags location using Kalman filtered RSSI, where the mean location estimation error was 0.4406 meter and standard deviation of 0.3354 meter.

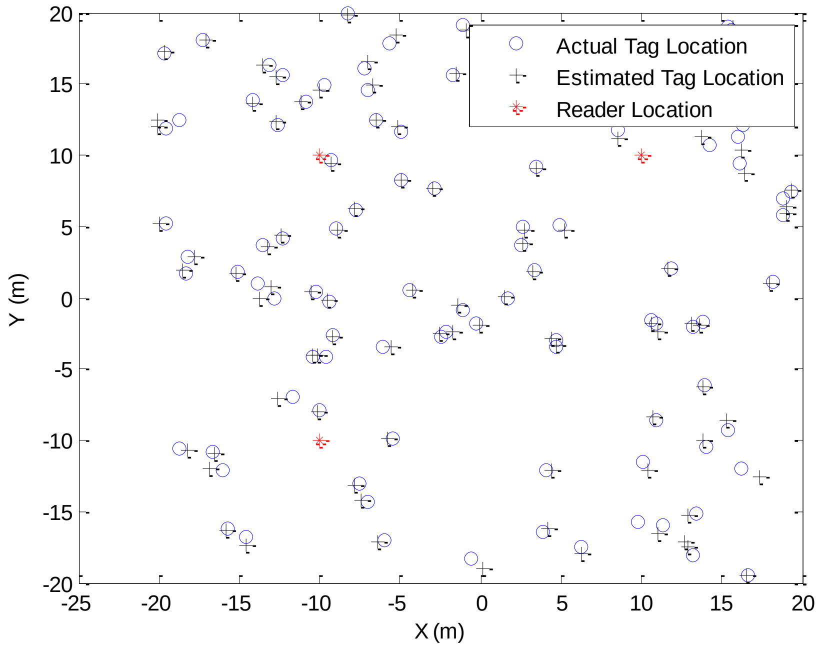


Figure 5-43: Localization using Kalman Filtered RSSI

The cumulative distribution function of the localization error in Figure 5-44 shows 0.992 m error with 95% confidence.

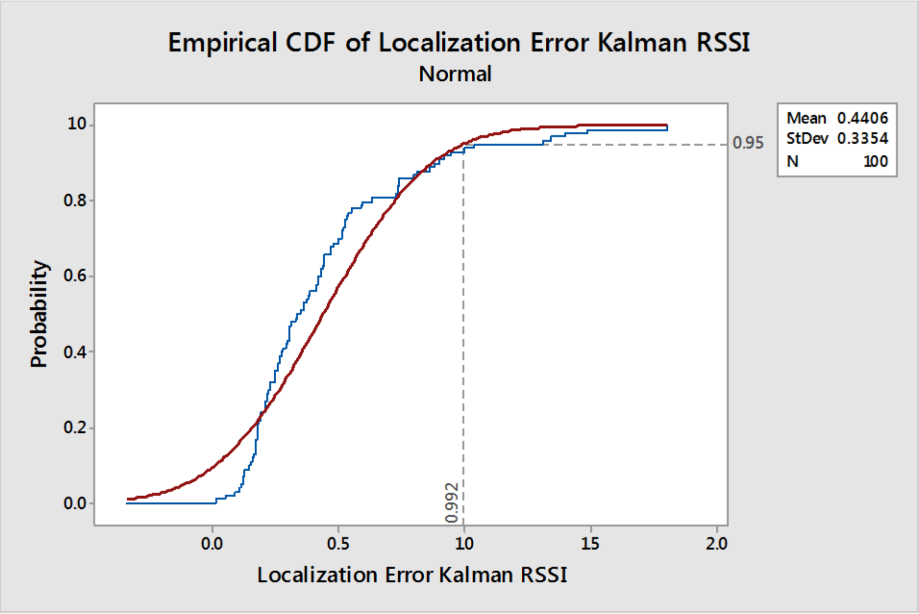


Figure 5-44: Localization Error CDF using Kalman Filtered RSSI

In order to show the improvement from using the Kalman filtered RSSI in comparison to raw RSSI, the two error CDF were compared in Figure 5-45. The Kalman filtered RSSI technique has better performance than the commonly used moving average method.

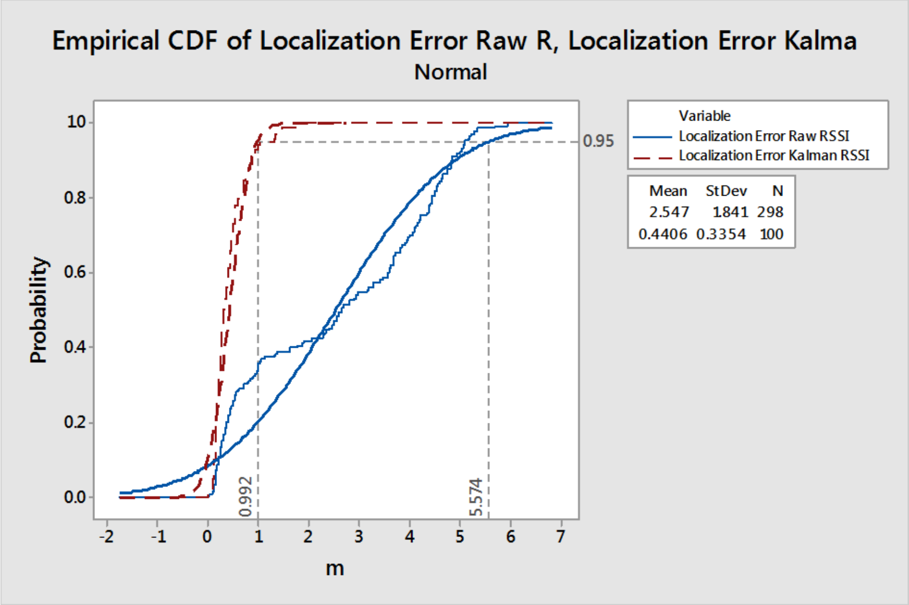


Figure 5-45: Localization Error CDF Comparison

The final step of the simulator test is to validate the Self-Calibration algorithm described in section 5.4.5. In order to simulate a change in the surrounding environment (interference), the standard deviation for the raw RSSI is increased to reflect higher noise in the system, and hence introduce higher noise in the location estimation. The following equation was used to reflect the higher noise in the environment:

$$RSSI_{StdDev} = 0.5658 \ln(distance) + 6.0113 \qquad (5\text{-}37)$$

100 random tags were generated in a space 20 m x 20 m, with 3 readers. The RSSI was generated using equations (5-33) and (5-37). The localization error was calculated using Equation (5-35) and the tag's distances from readers were estimated using the Kalman filtered RSSI path loss model Equation (5-36). Figure 5-46 illustrated the actual tags location verses the

estimated tags location using Kalman filtered RSSI, where the mean location estimation error was 3.368 meter and standard deviation of 1.359 meter. The cumulative distribution function of the localization error in Figure 5-47 shows 5.604 m error with 95% confidence.

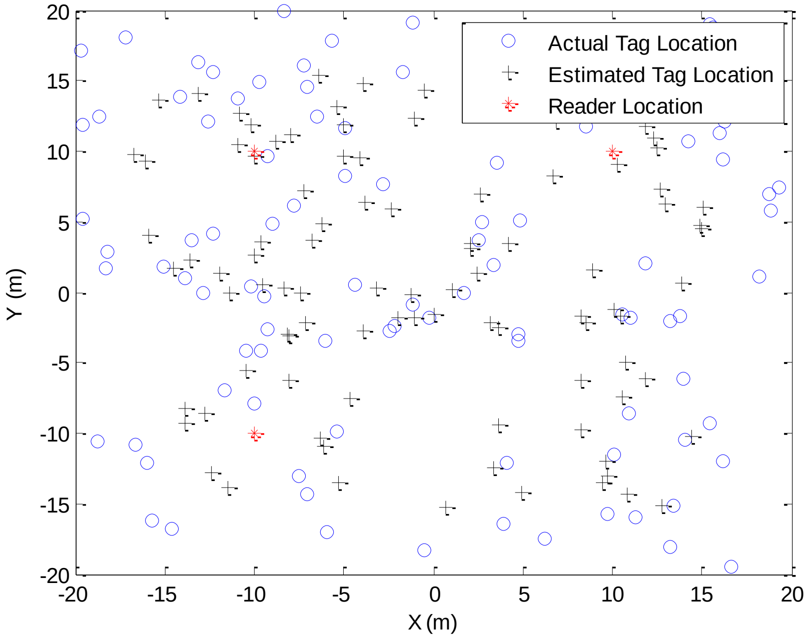


Figure 5-46: Localization using Kalman Filtered RSSI in High Noise Environment

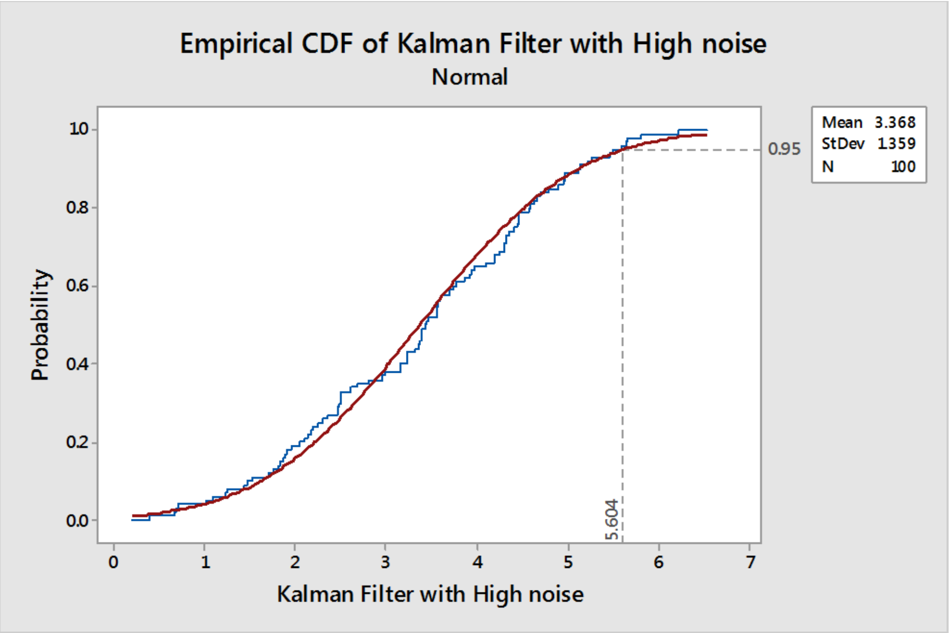


Figure 5-47: Localization Error CDF using Kalman Filtered RSSI in High Noise Environment

The Particle Swarm Optimization algorithm was applied to the noisy data to optimize the propagation model parameters. The A and B parameters were optimized after 39 iterations as shown in Figure 5-48. The path-loss after optimization is:

$$RSSI_{PSO} = -37.6061 - 7.9361\ln(d) \qquad (5\text{-}38)$$

Tag's actual location is plotted verses the estimated tag's location after the optimization as shown in Figure 5-49, where the mean location error was 0.837 meter and standard deviation of 0.639 meter as shown in Figure 5-50.

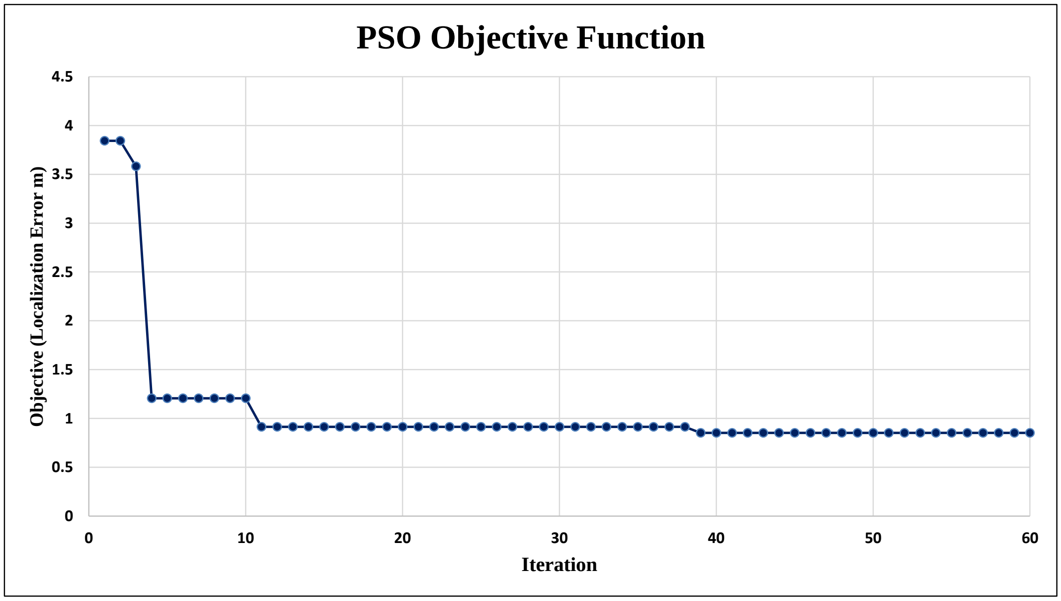


Figure 5-48: Objective Function Optimization

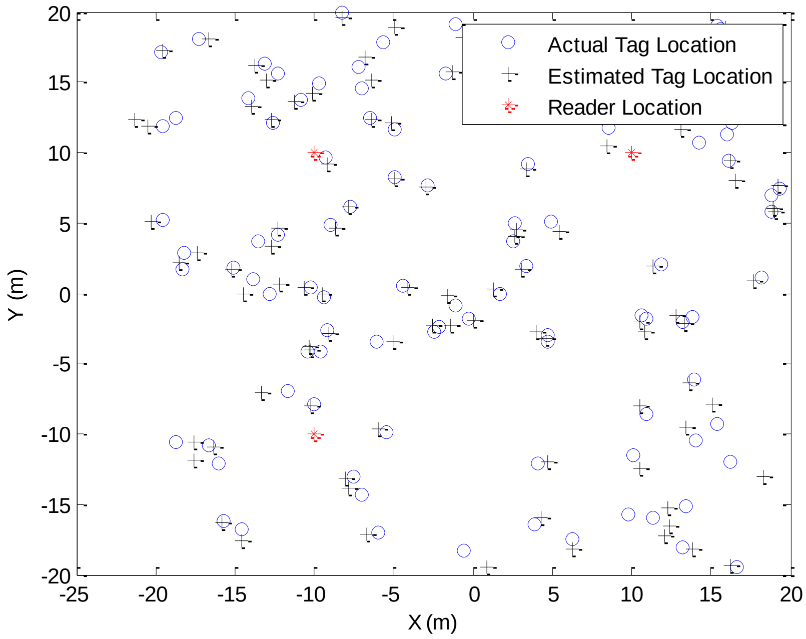


Figure 5-49: Localization using Kalman Filtered RSSI after Optimization

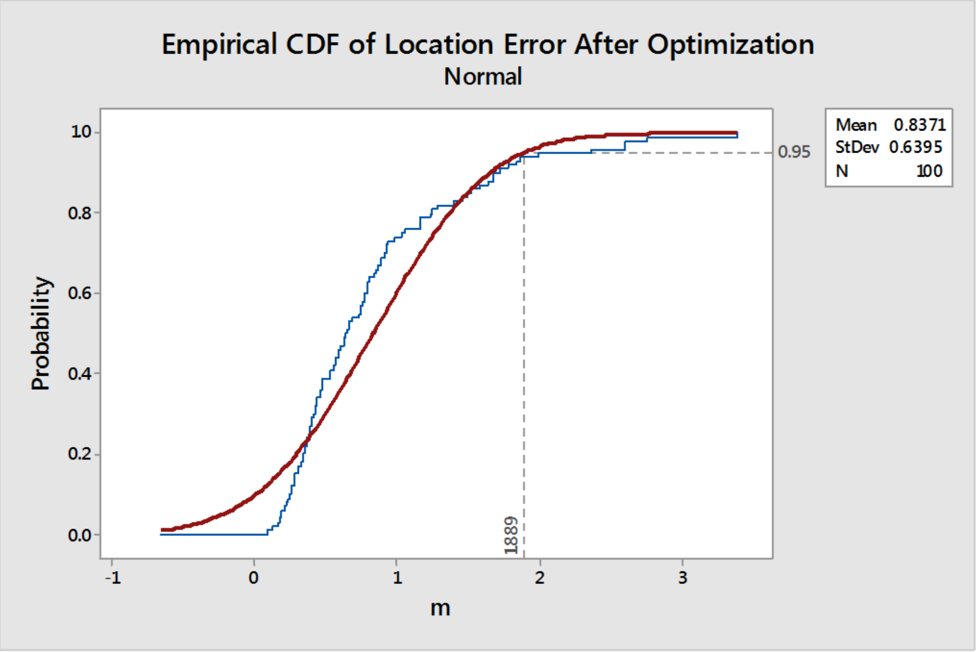


Figure 5-50: Localization Error CDF using Kalman Filtered RSSI after Optimization

The comparison between the localization error before and after the optimization illustrated in Figure 5-51 and Table 5-5 shows that the self-calibration algorithm is able to improve the localization accuracy and cope with changes in the surrounding environment.

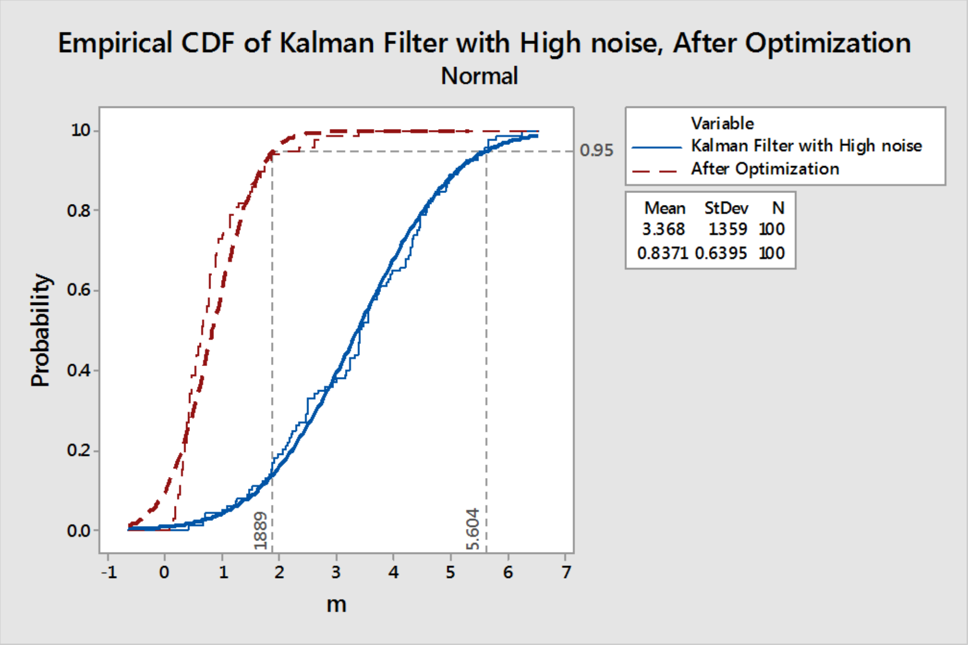


Figure 5-51: Localization Error CDF Comparison Before and After Optimization

Table 5-5: Mean Localization Errors before and after Optimization

	Before Optimization	After Optimization
Location Estimation Error (m)	3.368	0.837
Standard Deviation	1.359	0.639

5.9.2 Phase Two: Laboratory Experiments

To validate the results obtained by simulation and to test the overall performance of the developed prototype, several experiments were conducted in Concordia University Construction Automation Lab, apartment building, underground parking and warehouse environment. Table 5-6 shows the characteristics of each experiment, including the total number of data sets collected, date, location, total surface area and number of tags used in each experiment. The location's error is calculated as the distance in meters between the estimated and actual locations using Equation (5-35).

Table 5-6: Experiments Characteristics

Experiment	**Exp. 1**	**Exp. 2**	**Exp. 3**	**Exp. 4**
Date	01/12/2014	05/12/2014	08/12/2014	18/12/2014
Location	Lab Environment	Apartment Building	Underground Parking	Warehouse Environment
No of data sets	130	105	86	94
Surface Area (m2)	15	42	140	92
No of Deployed Tags	21	15	36	27
No of Readers	3	3	6	5
Area per tag m2	1.40	2.80	3.89	3.41

A grid formation test bed was utilized as shown in Figure 5-52, where readers are installed at the corners of the area, then tags where placed one meter apart in the grid formation.

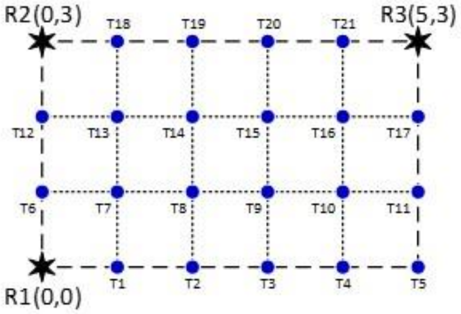


Figure 5-52: Sample of the Grid Formations used in Experiments

Figure 5-53 shows graphical display of a sample for actual verses estimated locations of tags using both raw RSSI and Kalman filtered RSSI signals. The yellow triangles represent the actual locations, the black crosses represent the calculated locations using raw RSSI and the red circles represent the calculated location using Kalman filtered RSSI method. The results show higher uncertainty and variances in location estimation using raw RSSI, which can be identified from the scattered nature of the calculated location. On the other hand, the locations estimated using the proposed filtered RSSI indicated higher certainty and less variances in the estimated locations. Such higher certainty is translated into less location estimation errors as shown in Figure 5-54, where the mean location error using raw RSSI and Kalman filtered RSSI method were 1.67 and 0.66 meters, respectively.

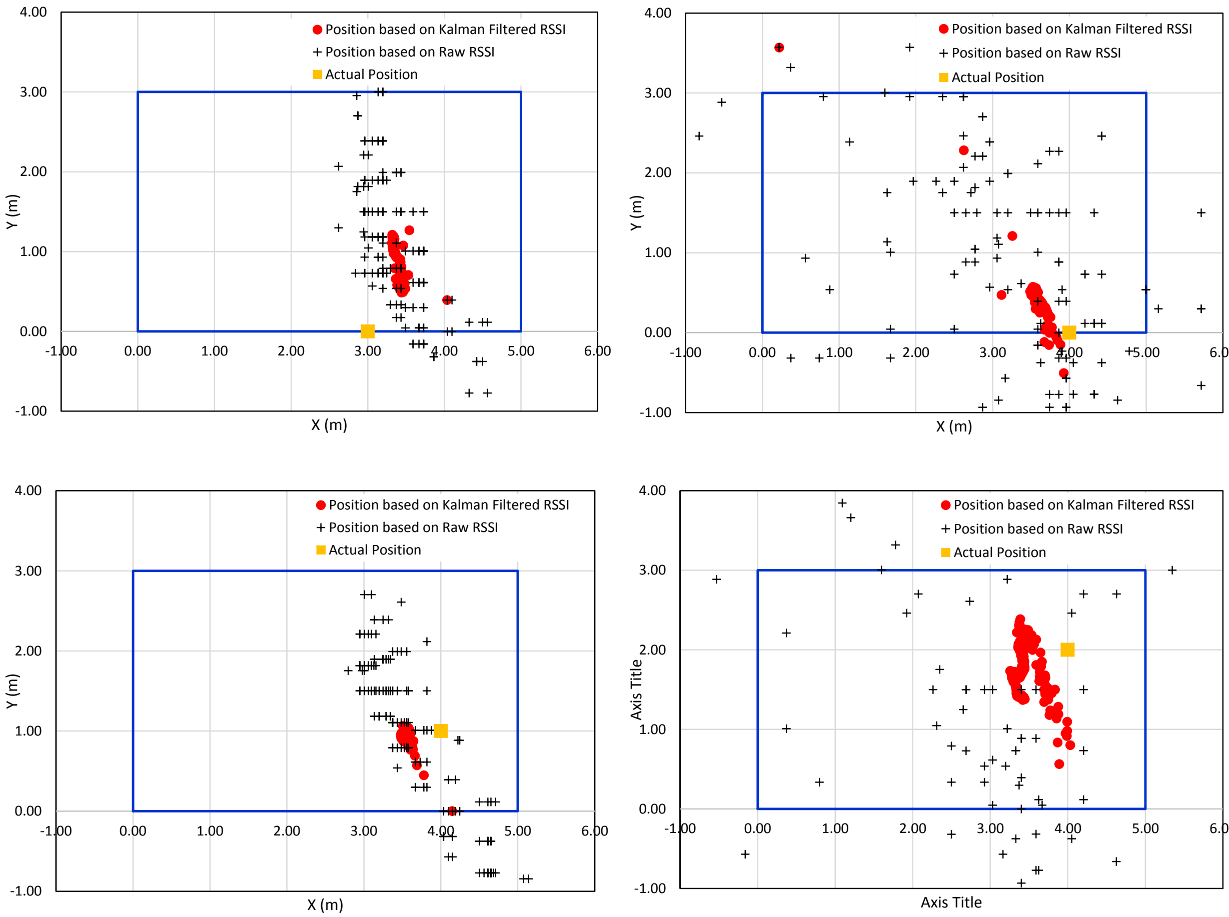


Figure 5-53: Graphical Representation of Actual vs Estimated Tag's Locations (Exp. 1)

A comparison between the histogram for the localization errors using raw RSSI and Kalman filtered RSSI, presented in Figure 5-54, shows that the location error using raw RSSI has a mean value of 1.673 m and standard deviation of 1.765 m (for 415 data sets). The developed Kalman Filtered RSSI method had a mean error of 0.66 m and standard deviation of 0.58 m.

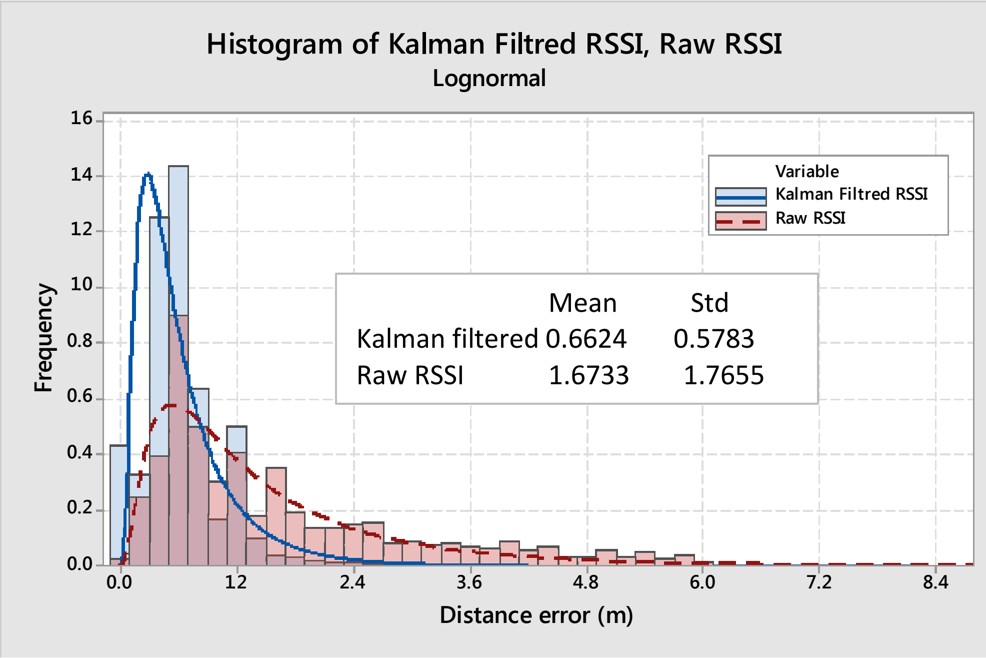


Figure 5-54: Location Errors Histograms (Exp. 1)

For validating the self-calibration algorithm, experiments were conducted using a grid formation test bed, where readers are installed at the corners of the area, then tags were placed one meter apart in grid formation as shown in Figure 5-55a. This test setup is repeated after adding physical obstacles in the surrounding environment (as shown in Figure 5-55b) to simulate the change in the environment and test how the proposed method self-calibrates its model to account for interferences caused by the surrounding environment. Experiments were conducted in a laboratory environment at Concordia University Construction Automation Lab. A total of 1062 data sets were collected covering 15 m2 of surface area. The test bed had 17 mobile tags, 4 reference tags and 3 fixed readers, with an average density of one reader per 5 m2 and one reference tag per 3.75 m2. In the first experimental step the setup in Figure 5-55 was used and dynamic signal propagation model, which has been initialized with initial values based on experimental measurements (A= -38.57 and B= -7.935) and tags' distances from the fixed readers were estimated accordingly. The system localization accuracy was measured at 87%.

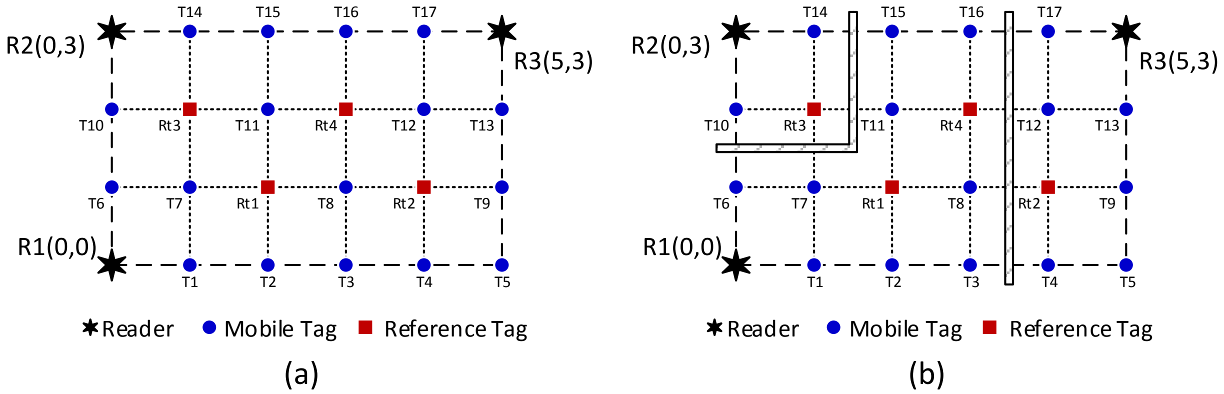


Figure 5-55: Grid Formation Test Bed (Exp. 1)

After adding physical obstacles as shown in Figure 5-55, the system location accuracy fall below 80% and the distance error were increased. Figure 5-56 shows a graphical display of the actual locations verses the estimated tag's locations after adding the obstacles, the orange triangles represent the actual tag's locations, and the black crosses represent the calculated tag's location.

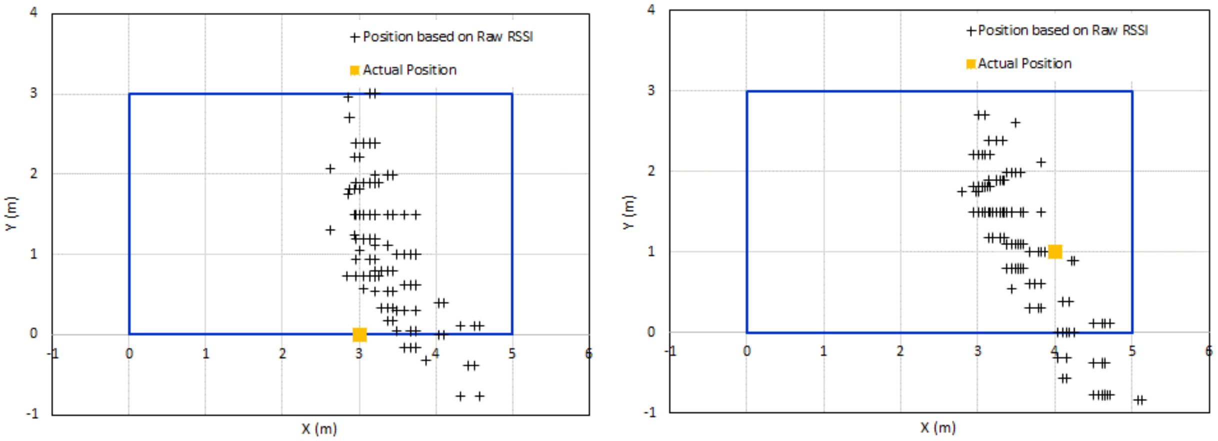


Figure 5-56: Actual vs Estimated Tag's Locations before Calibration Stage (Exp. 1)

The calibration stage was initiated based on encountering low system accuracy. Each reader signal propagation model is optimized using PSO algorithm and the RSSI from reference tags, the enhancement in the location estimation is clearly identified in Figure 5-57, which shows a graphical display of actual tag's locations verses the estimated tag's location after calibration.

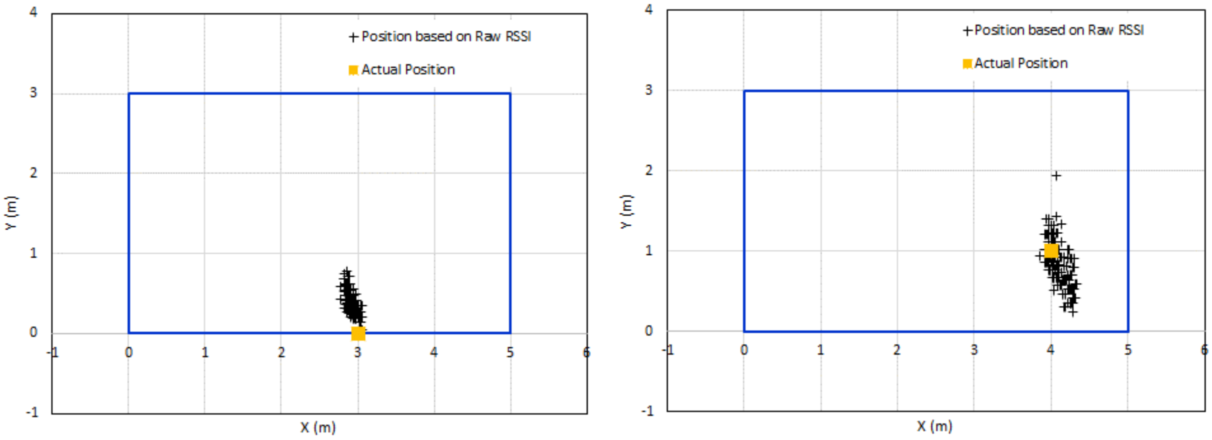


Figure 5-57: Actual vs Estimated Tag's Locations after Calibration Stage (Exp. 1)

Figure 5-58 shows the CDF of estimated distance errors for reader 1 before and after the calibration stage, where the mean error in the distance shows a decreasing trend. The mean absolute percentage error before calibration was 37.69%, while it was 14.96% after calibration. The SC-WSN method decreased the mean absolute percentage error by 60%.

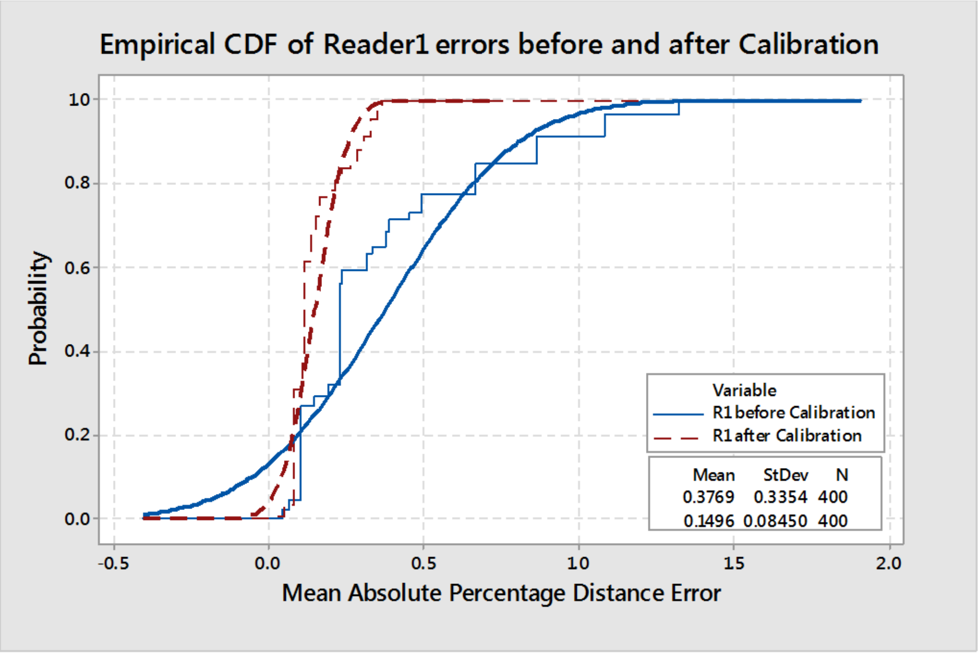
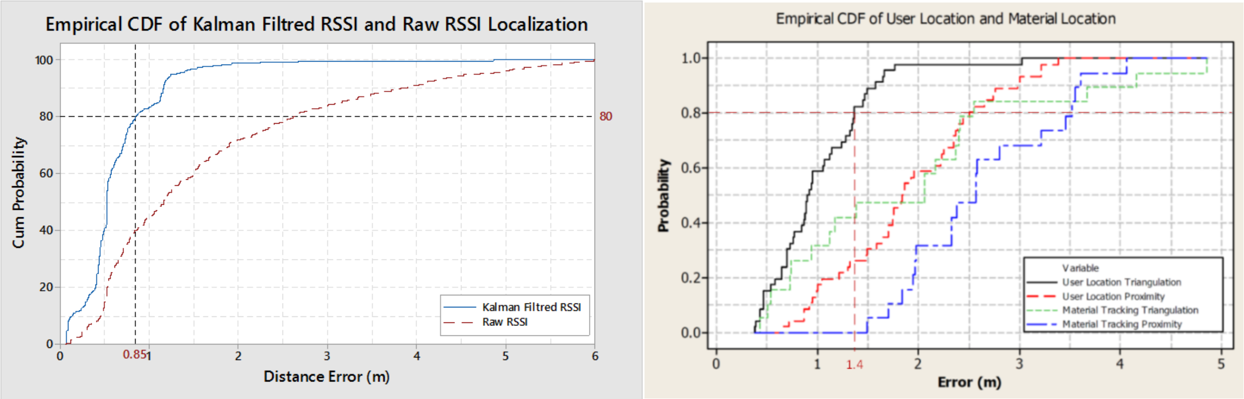


Figure 5-58: CDF of Estimated Distance Errors for Reader 1 before and after Calibration (Exp.1)

5.9.3 Discussion of Results

The cumulative probability density functions (CDF) of the distance error are usually used for measuring the precision of a system. To compare two positioning techniques with respect to accuracies and precision, the technique whose CDF graph reaches high probability values faster is more preferable, because its distance error is more concentrated in small values. In order to compare the proposed localization technique to those developed by others, the distance error CDF of the proposed technique is compared to the CDF graph for the system developed by Montaser and Moselhi (2014). The proposed Kalman filtered RSSI localization technique has a location precision of 90% within 1.16 m (the CDF of distance error of 1.16 m is 0.9) and 80% within 0.85 m, while the raw RSSI localization technique has a location precision of 90% within 3.70m and 80% within 2.60m as shown in Figure 5-59.



(a) (b)

Figure 5-59: Proposed Localization Error CDF vs Montaser and Moselhi (2014)

Moreover, the system developed by Montaser and Moselhi (2014) has a location precision of 90% within 1.60m and 80% within 1.40m as shown in Figure 5-59. The developed localization yields 68.6% and 27.5% enhancement over that based on unfiltered RSSI and that of Montaser (2014), respectively. In addition, the computational time required for that of Montaser (2014) is three times the time required for the proposed method due to the three location reading required in Montaser (2014), which presents higher advantage for the proposed method in real time localization applications.

To evaluate the performance of proposed method, several indoor experiments were conducted in different environments; lab environment, apartment building, indoor parking and warehouse environment. The proposed method produced location estimates with an average error of 0.66m in comparison with 1.67m using the raw RSSI signal. And with a likelihood of 80% the localization error of the proposed method is 0.85m in comparison to 2.60m using the raw RSSI signal. Moreover the performance of the proposed method was compared to previously developed system (Montaser and Moselhi 2014) using the cumulative distribution function (CDF) of the localization error. It was found that the proposed method has 27.5% less location absolute error than their method with a likelihood of 90%. The developed method would potentially improve indoor localization applications in construction such as automated project control and jobsite safety. The summary of the results obtained in this study are presented in Table 5-7.

Table 5-7: Localization Errors Summary

	Raw Signal Localization (Exp.1)	**Montaser and Moselhi 2014**	**SC-WSN** (Exp.1)
Mean Absolute Error	1.67	1.01	0.66
Standard Deviation	1.77	0.67	0.58
90 %Percentile	3.70	1.80	1.16
80% Percentile	2.60	1.60	0.85

5.10 Summary

Despite recent advances in wireless sensor technologies, mobile computing, and tracking techniques, indoor localization remains a technically challenging problem. Modeling indoor radio frequency signal propagation is not a simple task, especially in harsh and dynamic environments such as construction jobsites. This research presented an efficient localization prototype utilizing low cost radio frequency hardware modules for indoor localization based on RSSI signal smoothing and filtering. The proposed signal smoothing technique, which utilizes Kalman filter, not only increases the certainty in the estimated locations, but also enhanced the localization accuracy by over 68.6% of that based on the use of unfiltered RSSI. Such enhancement can be attributed to the filtering of the uncorrelated signal noise.

To evaluate the performance of proposed method, several indoor experiments were conducted in lab environment. The proposed method produced location estimates with an average error of 0.66m in comparison to 1.67m using unfiltered RSSI signals. And with a likelihood of 80% the localization error of the proposed method is 0.85m in comparison to 2.60m using unfiltered RSSI signals. Moreover the performance of the proposed method was also compared to that previously developed by Montaser and Moselhi (2014) using the cumulative distribution function (CDF) of localization errors. It was found that the proposed method outperformed their method by 27.5% with a likelihood of 90%. The developed method is expected to improve indoor localization applications in construction such as automated project control and onsite safety.

This paper presented a newly developed method for indoor localization on dynamic construction jobsites utilizing a self-calibrated wireless sensor network (SC-WSN). The developed SC-WSN hardware consists of fixed gateway unites mounted at predefined locations, reference tags and mobile tags mounted on tracked objects. The developed method consists of a prediction stage and calibration stage. The prediction stage estimates the tag’s location based on its measured signal strength (RSSI), which in turn is converted to the corresponding distance from fixed readers by a dynamic signal propagation model. The calibration stage is executed whenever the system accuracy falls below 80%, where the dynamic propagation model parameters are optimized to minimize the distance estimation errors of the reference tags. A particle swarm optimization (PSO) algorithm is used to find a near optimum solution for this non-linear problem. The PSO is not only able to find a solution effectively, but also has significantly better computational efficiency. Experimental results illustrated the significant accuracy improvement in estimating locations on construction jobsites, where the mean absolute percentage error before calibration was 37.69% while it was 14.96% after calibration. The SC-WSN method decreased the mean absolute percentage error by 60%.

Chapter 6 : CONCLUSIONS AND FUTURE WORK

6.1 Summary and Conclusions

This research aimed to study, design, configure and develop fully customized automated site data acquisition models, with a special focus on improving the accuracy of automated tracking and control of construction operations. A framework for automated progress tracking and control was developed encompassing two newly developed prototypes. The first is sensor aided GPS (SA-GPS) for tracking of outdoor construction operations and the second is self-calibrated wireless sensor Network (SC-WSN) for indoor localization.

Rapid prototyping technique was used as a vehicle for the hardware and software development in the two prototypes. The utilization of rapid prototyping in this research, allowed for faster development using virtual simulation environment and laboratory experiments. The hardware functions were tested and validated using simulation and lab experiments, then the prototype design was fine-tuned based on the results from these experiments. In this way, an economy of time and material are obtained. A modular hardware designing and configuring approach was utilized to allow for speedy redesign, increased ability to reuse some of the hardware modules and hence cut the development cost.

The SA-GPS prototype was designed with a special focus on earthmoving operations. This prototype consists of a microcontroller equipped with GPS module as well as a number of sensors such as accelerometer, barometric pressure sensor, Bluetooth proximity and strain gauges. The developed prototype is able to overcome standalone GPS limitations through data fusion of sensor data with GPS data, which enhances the progress assessment and productivity analysis. A fuzzy reasoning rule-based engine is designed to fuse data from various sensors to provide a comprehensive status of ongoing construction operations. A productivity analysis algorithm is developed to flag early operational bottlenecks, which provides project management personnel with a room to take necessary corrective actions in a timely manner. The SA-GPS prototype performance was compared to the standalone GPS through a case study. The results indicate superiority of the developed prototype over the standalone GPS, with mean absolute percentage error being 3% compared to 12% for the two methods respectively. The second prototype is the self-calibrated wireless sensor network (SC-WSN), which is designed for indoor localization and tracking of construction resources (labor, materials and equipment) inside buildings. This prototype enhanced the indoor localization accuracy by consistently adapting its parameters to cope with the changing construction environment. The SC-WSN prototype utilized a Kalman filter to filter and smooth the RSSI signal, which not only increased the certainty in the estimated location, but also reduced the localization error by 68.4% in comparison to that resulting from using the raw RSSI. Such enhancement can be attributed to the filtering of the uncorrelated white signal noise which enhanced the signal to noise ratio (SNR) by 72%.

The performance of the developed SC-WSN was compared to the system of previously developed by Montaser and Moselhi (2014) using the cumulative distribution function (CDF) of the localization error. The average absolute localization error dropped by 33%. The self-calibration feature was also tested on noisy simulated environment to measure the percentage of location improvement before and after the calibration. Accuracy improved by approximately 60% was achieved.

The captured data of the two prototypes is used to estimate the actual progress and hence enabled accurate and timely earned value analysis (EVA)

6.2 Research Contributions

The contributions of this research are expected to circumvent a number of limitations and challenges associated with current practice in tracking and progress reporting, and in existing automated site data acquisition methods. Specifically, the research contributions are:

- Design and conduct of a fundamental experimental investigation for design of customized automated data acquisition prototypes for efficient tracking and control of construction projects utilizing latest innovations in sensory and wireless technologies.
- Validation of concurrent design and testing of the developments made for automated site data acquisition using rapid prototyping techniques.
- Development of efficient data management scheme that utilizes integrated on-sensor node data processing and in-network data processing to transform raw data into high level useful and actionable information. In this context, data aggregation and processing render much faster near real-time progress measurements.
- Utilization of low-cost microcontrollers, sensors and wireless modules to design fully customized and flexible automated data acquisition prototypes for applications in construction. The adopted flexibility is expected to facilitate a wider scope of applications in construction safety, condition monitoring of civil infrastructure and energy management in built facilities.
- Development of fuzzy reasoning data fusion and knowledge extraction algorithm to provide a higher level of system redundancy in case of sensors malfunctions.

- Development of SA-GPS prototype to improve the accuracy of estimating earthmoving progress and related productivity in comparison to standalone GPS use.
- Development of SC-WSN to improve indoor localization accuracy while dynamically adapting its parameters to cope with the noisy, dynamic and continually changing construction environment. As well the utilization of Kalman filter to remove white uncorrelated noise from the RSSI.

6.3 Limitations

The following are the limitations of the developments made in this research:

- The developed prototypes were validated using simulation and laboratory experiments and not applied to real construction projects. The developed SA-GPS prototype was designed for earthmoving operations and not generic for outdoor construction. It, however, can be easily configured for applications such as concreting and landscape.
- The link between BIM 360 Field and the developed framework was proved in concept by only developing the necessary functions for importing and exporting data. A web-based software development with user graphical interface is needed.
- The self-calibrating algorithm was tested only on the Synapse SNAP wireless protocol. Further testing on other protocols and network configurations is needed.

6.4 Future Work

The following are some recommendations for future work areas:

- Development of a web-based software to link BIM 360 Field to the developed framework, and experiment with the self-calibrating algorithm on other wireless technologies.

- Deployment of the developed prototypes on a real construction projects to assess their performance in real construction environment.
- Expand on the productivity analysis algorithm to include more types of construction operations.

References

Acebes, F., Pajares, J., Galán, J. M., & López-Paredes, A. (2014). A new approach for project control under uncertainty. Going back to the basics. *International Journal of Project Management*, *32*(3), 423–434. http://doi.org/10.1016/j.ijproman.2013.08.003

Akhavian, R., & Behzadan, A. H. (2013). Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining. *Journal of Construction Engineering Management*, *139*(11), 04013021. http://doi.org/10.1061/(ASCE)CO.1943-7862.0000775

Akinci, B., Patton, M., & Ergen, E. (2002). Utilizing Radio Frequency Identification on Precast Concrete Components - Supplier's Perspective. In *Proc., the Nineteenth International Symposium on Automation and Robotics in Construction (ISARC 2002)* (pp. 381–386). Washington, DC USA.

Aliverdi, R., Moslemi Naeni, L., & Salehipour, A. (2013). Monitoring project duration and cost in a construction project by applying statistical quality control charts. *International Journal of Project Management*, *31*(3), 411–423. http://doi.org/10.1016/j.ijproman.2012.08.005

Alshibani, A. (1999). *A computerized cost and schedule control system for construction projects*. Master's thesis, Concordia Univ., Montreal, Canada.

Alshibani, A., & Moselhi, O. (2007). Tracking and forecasting performance of earthmoving operations using gps data. In *CME 25 Conference Construction Management and Economics* (pp. 1377–1388). Reading, UK.

Alshibani, A., & Moselhi, O. (2010). Automated System for Productivity Assessement of Earthmoving Operations, (Isarc), 684–693.

Alshibani, A., & Moselhi, O. (2012). Stochastic Method for Forecasting Project Time and Cost. *Construction Research Congress 2012*, 545–555. http://doi.org/10.1061/9780784412329.055

Anderson, B. D., & Moore, J. B. (2012). *Optimal filtering*. Courier Dover Publications.

Aryan, A., Shahi, A., West, J., Haas, C., Haas, R., & Caldwell, G. (2011). Evaluation of Ultra-Wideband Technology for Position Location and Progress Tracking in Indoor Construction Environments. In *3rd International/9th Construction Specialty Conference*. Ottawa, Ontario, Canada.

Bahl, P., & Padmanabhan, V. (2000). RADAR: An in-building RF-based user location and tracking system. In *Conf. of the IEEE Computer and Communications Societies* (pp. 775–784). New York: IEEE.

Baldwin, A. N., Thorpe, A., & Alkaabi, J. A. (1994). Improved material management through bar-code: Results and implications of a feasibility study. In *Proceedings of the Institution of Civil Engineers* (pp. 156–162).

Barraza, G., Back, W., & Mata, F. (2004). Probabilistic Forecasting of Project Performance Using Stochastic S Curves. *Journal of Construction Engineering and Management*, *130*(1), 25–32. http://doi.org/10.1061/(ASCE)0733-9364(2004)130:1(25)

Bosché, F. (2008). *Automated Recognition of 3D CAD Model Objects in Dense Laser Range Point Clouds*. University of Waterloo.

Bosche, F., Haas, C., & Murray, P. (2008). Performance of Automated Project Progress Tracking With 3D Data Fusion, 1–10.

Brilakis, I., Park, M. W., & Jog, G. (2011). Automated vision tracking of project related entities. *Advanced Engineering Informatics*, *25*(4), 713–724. http://doi.org/10.1016/j.aei.2011.01.003

Bulusu, N., Heidemann, J., & Estrin, D. (2000). GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, *7*(5), 28–34. http://doi.org/10.1109/98.878533

Caldas, C., Torrent, D., & Haas, C. (2006). Using Global Positioning System to Improve Materials-Locating Processes on Industrial Projects. *Journal of Construction Engineering and Management*, *132*(7), 741–749. http://doi.org/10.1061/(ASCE)0733-9364(2006)132:7(741)

Caron, F., Razavi, S., Song, J., Vanheeghe, P., Duflos, E., Caldas, C., & Haas, C. (2007). Locating sensor nodes on construction projects. *Autonomous Robots*, *22*(3), 255–263. http://doi.org/10.1007/s10514-006-9720-1

Caron, F., Ruggeri, F., & Merli, A. (2013). A Bayesian Approach to Improve Estimate at Completion in Earned Value Management. *Proj Mgmt Jrnl*, *44*, 3–16.

Chau, K. W., Anson, M., & Zhang, J. P. (2005). 4D dynamic construction management and visualization software: 1. Development. In *Automation in Construction* (Vol. 14, pp. 512–524). http://doi.org/10.1016/j.autcon.2004.11.002

Cheng, M. Y., & Chen, J. C. (2002). Integrating bar code and GIS for monitoring construction progress. *Journal of Automation in Construction*, *11*(3), 23–33.

Cheng, T., Migliaccio, G. C., Teizer, J., & Gatti, U. C. (2012). Data fusion of Real-time Location Sensing and Physiological Status Monitoring for Ergonomics Analysis of Construction Workers. *Journal of Computing in Civil Engineering*, *27*(3), 320–335. http://doi.org/10.1061/(ASCE)CP.1943-5487.0000222.

Cheok, G. ., Stone, W. C., Lipman, R. R., & Witzgall, C. (2000). Ladars for construction assessment and update. *Journal of Automation in Construction*, *9*(5-6), 463–477.

Chi, S., & Caldas, C. H. (2011). Automated Object Identification Using Optical Video Cameras on Construction Sites. *Computer-Aided Civil and Infrastructure Engineering*, *26*(5), 368–380. http://doi.org/10.1111/j.1467-8667.2010.00690.x

Cho, C.-Y., Kwon, S., Shin, T.-H., Chin, S., & Kim, Y.-S. (2011). A development of next generation intelligent construction liftcar toolkit for vertical material movement management. *Automation in Construction*, *20*(1), 14–27. http://doi.org/http://dx.doi.org/10.1016/j.autcon.2010.07.008

Christensen, D. S. (1993). The estimate at completion problem: a review of three studies. *Project Management Journal*, 35–42.

Deasy, T. P., & Scanlon, W. G. (2004). Stepwise algorithms for improving the accuracy of both deterministic and probabilistic methods in WLAN-based indoor user localization. In *Int. J. Wireless Inf. Networks* (pp. 207–216).

Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2008). *BIM Handbook: A Guide to Building Information Modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons, Inc.

Elnahrawy, E., Xiaoyan, L., & Martin, R. P. (2004). The limits of locali- zation using signal strength: A comparative study. In *First Annual IEEE Communications Society Conf. on Sensor and Ad Hoc Communications and Networks* (pp. 406–414). New York: IEEE.

El-Omari, S. (2008). *Automated data acquisition for tracking and control of construction projects*. Ph.D. Thesis presented to Building, Civil and Environmental Engineering Department, Concordia University, Montreal, Canada.

Elzarka, H., & Bell, L. (1997). Development of Pen-Based Computer Field Applications. *Journal of Computing in Civil Engineering*, *11*(2), 140–143. http://doi.org/10.1061/(ASCE)0887-3801(1997)11:2(140)

Ergen, E., & Akinci, B. (2007). An overview of approaches for utilizing RFID in construction industry. In *Proceedings of the 1st Annual RFID Eurasia Conference* (pp. 7–11).

Fard, M. G., & Peña-Mora, F. (2007). Application of Visualization Techniques for Construction Progress Monitoring. In *Computing in Civil Engineering (2007)* (pp. 216–223). American Society of Civil Engineers. http://doi.org/10.1061/40937(261)27

Fayek, A., AbouRizk, S., & Boyd, B. (1998). Implementation of automated site data collection with a medium-size contractor. In *Proceedings of ASCE Computing in Civil Engineering* (pp. 454–456). Boston, MA.

Fazeen, M., Gozick, B., Dantu, R., Bhukhiya, M., & González, M. C. (2012). Safe Driving Using Mobile Phones. *IEEE Transactions on Intelligent Transportation Systems*, *13*(3), 1462 – 1468. http://doi.org/10.1109/TITS.2012.2187640

Feng, M. W., Wen, S. L., Tsai, K. C., Liu, Y. C., & Lai, H. R. (2008). Wireless sensor network and sensor fusion technology for ubiquitous smart living space applications. In *Proceedings of the 2nd International Symposium on Universal Communication, ISUC 2008* (pp. 295–302). http://doi.org/10.1109/ISUC.2008.87

Ferdaus, S. N., Vardy, A., Mann, G., & Gosine, R. (2008). Comparing global measures of image similarity for use in topological localization of mobile robots. In *Canadian Conference on Electrical and Computer Engineering* (pp. 913–918). http://doi.org/10.1109/CCECE.2008.4564669

Finley, D. R. (2007). Point-In-Polygon Algorithm. Retrieved from http://alienryderflex.com/polygon/

Fleming, Q. W., & Koppelman, J. M. (2000). *Earned Value Project Management* (2nd ed.). Newtown Square, PA.: Project Management Institute Inc.

Fu, Q., & Retscher, G. (2009). Active RFID Trilateration and Location Fingerprinting Based on RSSI for Pedestrian Navigation. *The Journal of Navigation*, *62*(2), 323–340. http://doi.org/http://dx.doi.org/10.1017/S0373463308005195

Gelb, A. (1974). *Applied optimal estimation*. Cambridge, MA.: MIT Press.

Glanzer, G., Bernoulli, T., Wiessflecker, T., & Walder, U. (2009). Semi-autonomous indoor positioning using MEMS-based inertial measurement units and building information. In *2009 6th Workshop on Positioning, Navigation and Communication*. http://doi.org/10.1109/WPNC.2009.4907816

Golparvar-Fard, M., Heydarian, A., & Niebles, J. (2013). Vision-Based Action Recognition of Earthmoving Equipment Using Spatio-Temporal Features and Support Vector Machine Classifiers. *Advanced Engineering Informatics, 27,* 652–663.

Gonçalo, G., & Helena, S. (2009). A novel approach to indoor location systems using propagation models in WSNs. *International Journal on Advances in Networks and Services, 2*(4), 251–260.

Goodrum, P., McLaren, M., & Durfee, A. (2006). The application of active active radio frequency identification technology for tool tracking on construction job sites. *Automation in Construction*, *15*(3), 292–302.

Grindvoll, H., Vermesan, O., Crosbie, T., Bahr, R., Dawood, N., & Revel, G. M. (2012). A Wireless Sensor Network for Intelligent Building Energy Managemnt Based on Multi Communication Standards–A Case Study, *17*(December 2010), 43–62.

Hartmann, T., Gao, J., & Fischer, M. (2008). Areas of Application for 3D and 4D Models on Construction Projects. *Journal of Construction Engineering and Management*, *134*(10), 776–785. http://doi.org/10.1061/(ASCE)0733-9364(2008)134:10(776)

Hassan, R., Cohanim, B., de Weck, O. L., & Venter, G. (2005). A Comparison of Particle Swarm Optimization and the Genetic Algorithm. In *Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference* (pp. 18–21). Austin, TX.

Hassanein, A., & Moselhi, O. (2003). Tracking and control of linear infrastructure projects. In *Proceedings of 5th Construction Specialty Conference of the Canadian Society for Civil Engineering (CSCE 2003)*. New Brunswick, Canada.

Hassanien, A. (2002). *Planning and scheduling highway construction using GIS and dynamic programming*. PhD Thesis, Department of Building, Civil and Environmental Engineering, Concordia University, Montreal, Quebec.

Hassanpour, H. (2008). A time–frequency approach for noise reduction. *Digital Signal Processing*, *18*(5), 728–738.

Hightower, J., & Borriello, G. (2001). Location systems for ubiquitous computing. *Computer*. http://doi.org/10.1109/2.940014

Hildreth, J., Vorster, M., & Martinez, J. (2005). Reduction of Short-Interval GPS Data for Construction Operations Analysis. *Journal of Construction Engineering and Management*, *131*(8), 920–927. http://doi.org/10.1061/(ASCE)0733-9364(2005)131:8(920)

Huang, Q., Li, X., Shaurette, M., & Cox, R. (2011). Novel Sensor Network Architecture for Intelligent Building Environment Monitoring and Management. In *Computing in Civil Engineering (2011)* (pp. 347–354). American Society of Civil Engineers. http://doi.org/doi:10.1061/41182(416)43

Huang, Y. K., Hsiu, P. C., Chu, W. N., Hung, K. C., Pang, A. C., Kuo, T. W., … Fang, H. W. (2008). An integrated deployment tool for ZigBee-based wireless sensor networks. In *Proceedings of The 5th International Conference on Embedded and Ubiquitous Computing, EUC 2008* (Vol. 1, pp. 309–315). http://doi.org/10.1109/EUC.2008.87

Ibrahim, M., Germain, C., Guevremont, M., Forcier, M., & Moselhi, O. (2013). Automated development of construction schedules using onsite data acquisition. In *30th International Symposium on Automation and Robotics in Construction and Mining, ISARC* (pp. 840–848).

Ibrahim, M., & Moselhi, O. (2013a). Onsite data acquisition using low cost open source microcontroller. In *30th International Symposium on Automation and Robotics in Construction and Mining, ISARC* (pp. 404–416).

Ibrahim, M., & Moselhi, O. (2013b). Self Adaptive Forecasting for Construction Projects. In *Creative Construction Conference*. Budapest, Hungary.

Ibrahim, M., & Moselhi, O. (2014a). Automated Productivity Assessment of Earthmoving Operations. *Journal of Information Technology in Construction (ITcon)*, *19*, 169–184.

Ibrahim, M., & Moselhi, O. (2014b). Automated Productivity Assessment of Loader-Trucks Earthmoving Operations. In *Canadian Society of Civil Engineering (CSCE) Annual Conference*.

Ibrahim, M., & Moselhi, O. (2014c). Discussion of "Knowledge-Based Simulation Modeling of Construction Fleet Operations Using Multimodal-Process Data Mining" by Reza Akhavian and Amir H. Behzadan. *Journal of Construction Engineering and Management*, *140*(10), 7014001. http://doi.org/10.1061/(ASCE)CO.1943-7862.0000924

Ibrahim, M., & Moselhi, O. (2014d). Experimental study of wireless sensor networks for indoor construction operations. In *31st International Symposium on Automation and Robotics in Construction and Mining, ISARC* (pp. 805–812).

Ibrahim, M., & Moselhi, O. (2014e). Wireless Sensor Networks Configurations for Applications in Construction. *Procedia Engineering*, *85*, 260–273. http://doi.org/10.1016/j.proeng.2014.10.551

Ibrahim, M., & Moselhi, O. (2015a). Enhanced Localization for Indoor Construction. In *Creative Construction Conference (CCC)*. Krakow, Poland.

Ibrahim, M., & Moselhi, O. (2015b). IMU-Based Indoor Localization for Construction Applications. In *The 32th International Symposium on Automation and Robotics in Construction (ISARC)*. Oulu, Finland.

Ibrahim, M., & Moselhi, O. (2015c). Self- Calibrated WSN for Indoor Tracking and Control of Construction Operations. In *CSCE International Construction Specialty Conference*. Vancouver, British Columbia.

Iranmanesh, S. H., & Mokhtari, Z. (2008). Application of Data Mining Tools to Predicate Completion Time of a Project, 204–209.

Jang, W., & Skibniewski, M. J. (2007). Wireless Network-Based Tracking and Monitoring on Project Sites of Construction Materials. *Proceedings of the 9th International Conference on Modern Building Material, Structures and Techniques*, 8.

Jang, W.-S., Healy, W. M., & Skibniewski, M. J. (2008). Wireless sensor networks as part of a web-based building environmental monitoring system. *Automation in Construction*. http://doi.org/10.1016/j.autcon.2008.02.001

Jaselskis, E., Anderson, M. R., Jahren, C. T., Rodriguez, Y., & Njos, S. (1995). Radio- frequency identification applications in construction industry. *Journal of Construction Engineering and Management, 121*(2), 189– 196.

JBKnowledge. (2014). *THE 3rd CONSTRUCTION TECHNOLOGY REPORT*.

Jog, G. M., Park, M.-W., & Brilakis, I. (2011). Truck-face recognition using semantic texton forests. In *Proc., 3rd Int./9th Construction Specialty Conf., CSCE*. Ottawa, ON, Canada.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, *82*(1), 35–45.

Kang, J., Anderson, S., & Clayton, M. (2007). Empirical Study on the Merit of Web-Based 4D Visualization in Collaborative Construction Planning and Scheduling. *Journal of Construction Engineering and Management*, *133*(6), 447–461. http://doi.org/10.1061/(ASCE)0733-9364(2007)133:6(447)

Karl, H., & Willig, A. (2007). *Protocols and architectures for wireless sensor networks*. New York: Wiley- Interscience.

Khaleghi, B., Khamis, A., Karray, F. O., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, *14*(1), 28–44. http://doi.org/10.1016/j.inffus.2011.08.001

Khosrowpoura, A., Nieblesb, J. C., & Golparvar-Fard, M. (2014). Vision-based workface assessment using depth images for activity analysis of interior construction operations. *Automation in Construction*, *48*, 74–87. http://doi.org/10.1016/j.autcon.2014.08.003

Khoury, H., & Kamat, V. (2007). WLAN Based User Position Tracking for Contextual Information Access in Indoor Construction Environments. In *Computing in Civil Engineering (2007)* (pp. 838–845). American Society of Civil Engineers. http://doi.org/doi:10.1061/40937(261)99

Khoury, H. M., & Kamat, V. R. (2009). Evaluation of position tracking technologies for user localization in indoor construction environments. *Automation in Construction*, *18*(4), 444–457. http://doi.org/10.1016/j.autcon.2008.10.011

Kim, C., Park, T., Lim, H., & Kim, H. (2013). On-site construction management using mobile computing technology. *Automation in Construction*, *35*, 415–423. http://doi.org/10.1016/j.autcon.2013.05.027

Kim, S., Bai, Y., Jung, Y.-K., & Yun, D. (2011). Highway Construction Productivity Measurement with a Wireless Real-Time Productivity Measurement System. *Journal of the Transportation Research Board*, *2228*(1), 26–33.

Kimoto, K., Endo, K., Iwashita, S., & Fujiwara, M. (2005). The application of PDA as mobile computing system on construction management. *Automation in Construction*, *14*(2005), 500–511.

Kintner-Meyer, M. (2005). Opportunities of Wireless Sensors and Controls for Building Operation. *Energy Engineering*, *102*(5), 27–48. http://doi.org/10.1080/01998590509509441

Kouche, A. El, Hassanein, H. S., & Obaia, K. (2014). WSN platform Plug-and-Play (PnP) customization. In *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)* (pp. 1–6). http://doi.org/10.1109/ISSNIP.2014.6827642

Lamsweerde, A. van. (2001). Goal-oriented requirements engineering: a guided tour. In *Proceedings Fifth IEEE International Symposium on Requirements Engineering*. http://doi.org/10.1109/ISRE.2001.948567

Ledin, J. (2001). *Simulation Engineering: Build Better Embedded Systems Faster*. CMPBooks, USA.

Lee, W., Song, J., Kwon, S., Chin, S., Choi, C., & Kim, Y. (2008). A gate sensor for construction logistics. In *Proceedings of the 25th International Symposium on Automation and Robotics in Construction (ISARC 2008)* (pp. 100–105).

Li, H., Chan, G., Huang, T., Skitmore, M., Tao, T. Y. E., Luo, E., … Li, Y. F. (2015). Chirp-spread-spectrum-based real time location system for construction safety management: A case study. *Automation in Construction*, *55*, 58–65. http://doi.org/10.1016/j.autcon.2015.03.024

Li, J. (2004). *Web-based integrated project control*. Ph.D. Thesis presented to Building, Civil and Environmental Engineering Department, Concordia University, Montreal, Canada.

Li, J., Moselhi, O., & Alkass, S. (2006). Forecasting Project Status by Using Fuzzy Logic. *Journal of Construction Engineering and Management*, *132*(11), 1193–1202. http://doi.org/10.1061/(ASCE)0733-9364(2006)132:11(1193)

Li, N., & Becerik-Gerber, B. (2011). Performance-based evaluation of RFID-based indoor location sensing solutions for the built environment. *Advanced Engineering Informatics*, *25*(3), 535–546. http://doi.org/10.1016/j.aei.2011.02.004

Luo, J., Ying, K., & Bai, J. (2005). Savitzky–Golay smoothing and differentiation filter for even number data. *Signal Processing*, *85*(7), 1429–1434.

Lytle, A. M. (2011). *A framework for object recognition in construction using building information modeling and high frame rate 3D imaging*. Ph.D. Thesis presented to the Faculty of the Virginia Polytechnic Institute and State University.

Mahalik, N. P. (2007). *Sensor networks and configuration: fundamentals, standards, platforms, and applications*. Berlin: New York : Springer.

Malatras, A., Asgari, A., & Baugé, T. (2008). Web enabled wireless sensor networks for facilities management. *IEEE Systems Journal*, *2*(4), 500–512. http://doi.org/10.1109/JSYST.2008.2007815

Mazuelas, S., Bahillo, A., Lorenzo, R. ., Fernandez, P., Lago, F. ., & Garcia, E. (2009). Robust indoor positioning provided by real- time RSSI values in unmodified WLAN networks. *IEEE Journal of Selected Topics in Signal Processing*, *3*(5), 821–831.

McCullouch, B. (1997). Automating Field Data Collection on Construction Organizations. In *5th Construction Congress: Managing Engineered Construction in Expanding Global Markets* (pp. 957–963). Minneapolis.

Montaser, A. (2013). *Automated Site Data Acquisition for Effective Project Control*. Ph.D. Thesis presented to Building, Civil and Environmental Engineering Department, Concordia University, Montreal, Canada.

Montaser, A., Bakry, I., Alshibani, A., & Moselhi, O. (2012). Estimating productivity of earthmoving operations using spatial technologies. *Canadian Journal of Civil Engineering*. http://doi.org/10.1139/l2012-059

Montaser, A., & Moselhi, O. (2012a). 4D and tablet PC for progress reporting. In *Proceedings of International Symposium on Automation and Robotics in Construction (ISARC 2012)*. Eindhoven, Netherlands.

Montaser, A., & Moselhi, O. (2012b). RFID + for tracking earthmoving operations. In *Proceedings of Construction Research Congress* (pp. 1011–1020).

Montaser, A., & Moselhi, O. (2014). RFID indoor location identification for construction projects. *Automation in Construction*, *39*, 167–179. http://doi.org/10.1016/j.autcon.2013.06.012

Moon, S., & Yang, B. (2009). Effective Monitoring of the Concrete Pouring Operation in an RFID-Based Environment. *Journal of Computing in Civil Engineering*, *24*(1), 108–116. http://doi.org/10.1061/(ASCE)CP.1943-5487.0000004

Moselhi, O. (1993). Applied earned value for control. In *International Symposium of CIB W- 65*. St. Augustine, Trinidad and Tobago.

Moselhi, O. (2011). The use of earned value in forecasting project duration. In *28th International Symposium on Automation and Robotics in Construction, ISARCInternational Symposium on Automation and Robotics in Construction (ISARC)*. Seoul, KOREA.

Moselhi, O., & El-Omari, S. (2007). Hybrid Methodology for automated Collection of Data from Construction Sites.

Moselhi, O., Li, J., & Alkass, S. (2004). Web-based integrated project control system. *Journal of Construction Management and Economics*, *22*(1), 35–46.

Motamedi, A., & Hammad, A. (2009). Lifecycle management of facilities components using radio frequency identification and building information model. *Journal of Information Technology in Construction (ITCON)*, *14*, 238–262.

Naeni, L. M., Shadrokh, S., & Salehipour, A. (2011). A fuzzy approach for the earned value management. *International Journal of Project Management*, *29*(6), 764–772. http://doi.org/10.1016/j.ijproman.2010.07.012

Narbaev, T., & De Marco, A. (2014). An Earned Schedule-based regression model to improve cost estimate at completion. *International Journal of Project Management*, *32*(6), 1007–1018. http://doi.org/10.1016/j.ijproman.2013.12.005

Narbaev, T., & Marco, A. De. (2011). Cost Estimate at Completion Methods in Construction Projects, *15*, 32–36.

Navon, R., & Shpatnitsky, Y. (2005). A model for automated monitoring of road construction. *Construction Management and Economics*, *23*(9), 941–951.

Osterlind, F., Pramsten, E., Roberthson, D., Eriksson, J., Finne, N., & Voigt, T. (2007). Integrating building automation systems and wireless sensor networks. *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*. http://doi.org/10.1109/EFTA.2007.4416941

Park, M. W., Makhmalbaf, A., & Brilakis, I. (2011). Comparative study of vision tracking methods for tracking of construction site resources. *Automation in Construction*, *20*(7), 905–915. http://doi.org/10.1016/j.autcon.2011.03.007

Penã-Mora, F., & Dwivedi, G. (2002). Multiple Device Collaborative and Real Time Analysis System for Project Management in Civil Engineering. *Journal of Computing in Civil Engineering*, *16*(1), 23–38. http://doi.org/10.1061/(ASCE)0887-3801(2002)16:1(23)

Pertuz, A. (1968). *Adaptive Time Series Forecasting*.

Randell, C., Djiallis, C., & Muller, H. (2003). Personal position measurement using dead reckoning. In *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings*. http://doi.org/10.1109/ISWC.2003.1241408

Rappaport, T. S. (1996). *Wireless Communications: Principles and Practice*. Prentice-Hall, IEEE.

Razavi, S. N. (2010). *Data Fusion for Materials Location Estimation in Construction*. University of Waterloo.

Razavi, S. N., & Moselhi, O. (2012). GPS-less indoor construction location sensing. *Automation in Construction*, *28*, 128–136. http://doi.org/10.1016/j.autcon.2012.05.015

Razavi, S. N., Young, D. A., Nasir, H., Haas, C., Caldas, C., Goodrum, P., & Murray, P. (2008). Field trial of automated material tracking in construction. In *Annual Conference of the Canadian Society for Civil Engineering* (pp. 1503–1511). Quebec City, QC.

Rensfelt, O. (2012). *Experimental Challenges in Wireless Sensor Networks – Environment , Mobility , and Interference*.

Rezazadeh Azar, E., Dickinson, S., & McCabe, B. (2013). Server-Customer Interaction Tracker: Computer Vision–Based System to Estimate Dirt-Loading Cycles. *Journal of Construction Engineering and Management*, *139*(7), 785–794. http://doi.org/10.1061/(ASCE)CO.1943-7862.0000652

Rezazadeh Azar, E., & McCabe, B. (2012). Automated Visual Recognition of Dump Trucks in Construction Videos. *Journal of Computing in Civil Engineering*, *26*(6), 769–781. http://doi.org/10.1061/(ASCE)CP.1943-5487.0000179

Rueppel, U., & Stuebbe, K. . (2008). BIM-based indoor-emergency-navigation-system for complex buildings. *Tsinghua Science Technology*, *13*(1), 362–367.

Sacks, R., Navon, R., Brodetskaia, I., & Shapira, A. (2005). Feasibility of Automated Monitoring of Lifting Equipment in Support of Project Control. *Journal of Construction Engineering and Management*, *131*(5), 604–614. http://doi.org/10.1061/(ASCE)0733-9364(2005)131:5(604)

Sadeghioon, A., Metje, N., Chapman, D., & Anthony, C. (2014). SmartPipes: Smart Wireless Sensor Networks for Leak Detection in Water Pipelines. *Journal of Sensor and Actuator Networks*, *3*(1), 64–78. http://doi.org/10.3390/jsan3010064

Saidi, K., Haas, C., & Balli, N. (2002). The value of handheld computers in construction. In *Proc., 19th International Symp. on Automation and Robotics in Construction, (ISARC)* (pp. 557–562). Washington, DC, USA.

Savitzky, A., & Golay, M. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, *36*(8), 1627–1639.

Shah, R., Dawood, N., & Castro, S. (2008). Automatic generation of progress profiles for earthwork operations using 4D visualisation model. *Electronic Journal of Information Technology in Construction*, *13*, 491–506.

Shahandashti, S. M., Akinci, B., Garrett, J., & Soibelman, L. (2010). IDENTIFICATION OF INFORMATION REQUIREMENTS USING SIMULATION FOR SUPPORITING CONSTRUCTION PRODUCTIVITY ASSESSMENT. In *Proceedings of the 2010 Winter Simulation Conference* (pp. 3076–3087).

Shahandashti, S., Razavi, S., Soibelman, L., Berges, M., Caldas, C., Brilakis, I., … Zhu, Z. (2011). Data-Fusion Approaches and Applications for Construction Engineering. *Journal of Construction Engineering and Management*, *137*(10), 863–869. http://doi.org/10.1061/(ASCE)CO.1943-7862.0000287

Shahi, A. (2012). *Activity-Based Data Fusion for Automated Progress Tracking of Construction Projects*. University of Waterloo.

Shahi, A., Safa, M., Haas, C., & West, J. (2014). Data Fusion Process Management for Automated Construction Progress Estimation. *Journal of Computing in Civil Engineering*, 1–9. http://doi.org/10.1061/(ASCE)CP.1943-5487.0000436.

Shahi, A., West, J., & Haas, C. (2013). Onsite 3D marking for construction activity tracking. *Automation in Construction*, *30*, 136–143. http://doi.org/10.1016/j.autcon.2012.11.027

Shen, X. ., Chen, W., & Lu, M. (2008). Wireless sensor networks for resources tracking at building construction sites. *Tsinghua Science and Technology*, *13*(S1), 78–83. http://doi.org/10.1016/S1007-0214(08) 70130-5

Shen, X., & Lu, M. (2012). This paper is one of a selection of papers in this Special Issue on Construction Engineering and Management. *Canadian Journal of Civil Engineering*, *39*(9), 1083–1088. http://doi.org/10.1139/l2012-094

Shin, S. Y., Park, H. S., & Kwon, W. H. (2007). Mutual interference analysis of IEEE 802.15.4 and IEEE 802.11b. *Computer Networks*, *51*(12), 3338–3353. http://doi.org/10.1016/j.comnet.2007.01.034

Soibelman, L., Wu, J., Caldas, C., Brilakis, I., & Lin, K. (2008). Management and analysis of unstructured construction data types. *Advanced Engineering Informatics*, *22*(1), 15–27.

Soltani, M. M., Motamedi, A., & Hammad, A. (2013). Enhancing Cluster-based RFID Tag Localization Using Artificial Neural Networks and Virtual Reference Tags. *Automation in Construction*, *54*(October), 452–461. http://doi.org/10.1109/IPIN.2013.6817886

Song, J., Haas, C., & Caldas, C. (2006). Tracking the Location of Materials on Construction Job Sites. *Journal of Construction Engineering and Management*, *132*(9), 911–918. http://doi.org/10.1061/(ASCE)0733-9364(2006)132:9(911)

Song, J., Haas, C., Caldas, C., Ergen, E., & Akinci, B. (2006). Automating the task of tracking the delivery and receipt of fabricated pipe spools in industrial projects. *Automation in Construction*, *15*(2), 166–177. http://doi.org/10.1016/j.autcon.2005.03.001

Stüber, G. L., & Caffrey, J. J. (1999). *Radiolocation Techniques. The Mobile Com- munications Handbook* (2nd Editio). CRC Press.

Styliadis, A. D. (2007). Digital documentation of historical buildings with 3d modelling functionality. *Automation in Construction*, *16*, 498–510.

Sunkpho, J., & Garrett, J. (2003). Java Inspection Framework: Developing Field Inspection Support Systems for Civil Systems Inspection. *Journal of Computing in Civil Engineering*, *17*(4), 209–218. http://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(209)

Taneja, S., Akcamete, A., Akinci, B., Garrett, J. H., Soibelman, L., & East, E. W. (2012). Analysis of Three Indoor Localization Technologies for Supporting Operations and Maintenance Field Tasks. *Journal of Computing in Civil Engineering*, *26*(6), 708–719. http://doi.org/10.1061/(ASCE)CP.1943-5487.0000177

Teizer, J., Lao, D., & Sofer, M. (2007). Rapid automated monitoring of construction site activities using ultra-wideband. *24th International Symposium on Automation and Robotics in Construction, ISARC 2007,* (2), 23–28.

Tserng, H. P., Dzeng, R. J., Lin, Y. C., & Lin, S. T. (2005). Mobile construction supply chain management using PDA and Bar Codes. *Computer-Aided Civil and Infrastructure Engineering*, *20*(4), 242–264. http://doi.org/10.1111/j.1467-8667.2005.00391

Turkan, Y. (2012). *Automated Construction Progress Tracking using 3D Sensing Technologies*. Phd thesis presented to the University of Waterloo.

Turkan, Y., Bosché, F., Haas, C., & Haas, R. (2013). Toward Automated Earned Value Tracking Using 3D Imaging Tools. *Journal of Construction Engineering and Management*, *139*(4), 423–433. http://doi.org/10.1061/(ASCE)CO.1943-7862.0000629.

Vahdatikhaki, F., & Hammad, A. (2014). Framework for near real-time simulation of earthmoving projects using location tracking technologies. *Automation in Construction*, *42*, 50–67. http://doi.org/10.1016/j.autcon.2014.02.018

Ward, M. J., Thorpe, A., & Price, A. D. F. (2003). SHERPA: mobile wireless data capture for piling works. *Computer-Aided Civil and Infrastructure Engineering*, *13*, 299–314.

Widrow, B. (1966). *Adaptive Filters I: Fundamentals*.

Woo, S., Jeong, S., Mok, E., Xia, L., Choi, C., Pyeon, M., & Heo, J. (2011). Application of WiFi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR. *Automation in Construction*, *20*(1), 3–13. http://doi.org/10.1016/j.autcon.2010.07.009

Yang, S. K., Liu, T. S., & Cheng, Y. C. (2008). Automatic measurement of payload for heavy vehicles using strain gages. *Measurement*, *41*(5), 491–502. http://doi.org/10.1016/j.measurement.2007.07.003

Zhu, Z., German, S., & Brilakis, I. (2010). Detection of large-scale concrete columns for automated bridge inspection. *Automation in Construction*, *19*(8), 1047–1055.

Zwikael, O., Globerson, S., & Raz, T. (2000). Evaluation of models for forecasting the final cost of a project. *Project Management Journal*, *31*(1), 53–57.

Appendix A: Weather Station

The Libelium weather station can monitor 6 parameters related to the weather, such as ambient temperature and humidity, atmospheric pressure, precipitation and wind speed and direction. The kit comprises of 6 sensors: temperature sensor, humidity sensor, barometric pressure sensor, wind gauge, anemometer and wind vane, as shown in figure (A-1).

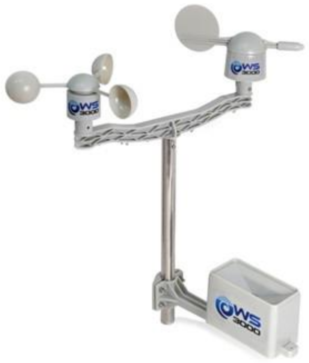


Figure A-1: Libelium Weather Station

Example of data collected with the weather station is shown in Figure (A-2).

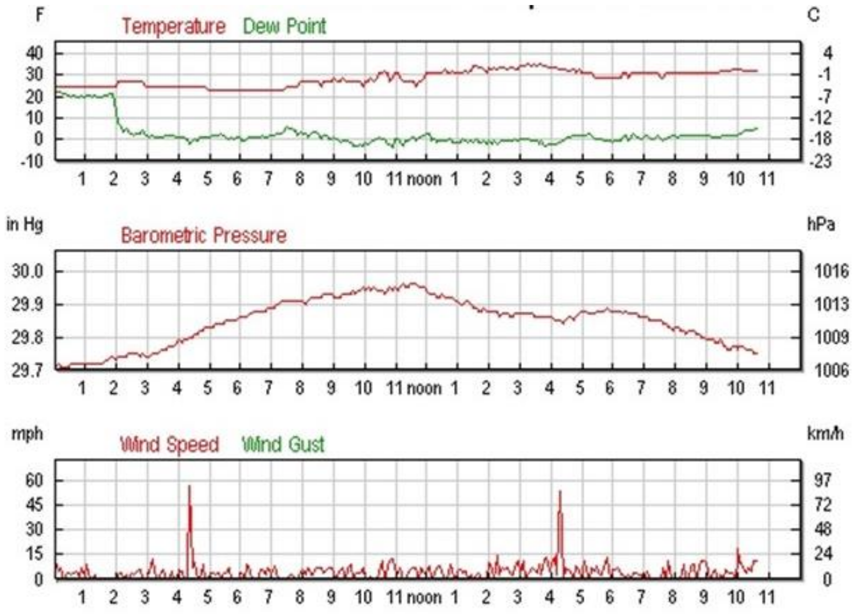


Figure A-2: Sample Weather Data

Appendix B: BIM360 FIELD API CODE

Code Snippet for connecting to the BIM360 server

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Autodesk.BIM360Field.APIService;
using Autodesk.BIM360Field.APIService.Models;
using Autodesk.BIM360Field.APIService.Support;

namespace APIExample
{
    class APIExample
    {
        static void Main(string[] args)
        {
            string username = "m_omari@encs.concordia.ca";
            string password = "password";
            string server = "https://api.velasystems.com";

            switch (args.Length)
            {
                case 0:
                case 1:
                    Console.WriteLine("You must specify a username, password
and optionally the server to connect to");
                    Console.WriteLine(string.Format("APIExample.exe {0} {1}
{2}", username, password, server));
                    Environment.Exit(1);
                    break;
                case 2:
                    username = args[0];
                    password = args[1];
                    break;
                case 3:
                    username = args[0];
                    password = args[1];
                    server = args[2];
                    break;
            }

            Console.WriteLine(string.Format("Connecting to BIM 360 Field API
service on {0} as {1}", server, username));


            try
            {
                API api = new API(server);

                api.authenticate(username, password);

                Console.WriteLine("Authenticated successfully. Retrieving
project list.\n\n");
```

```
                List<Project> projects = api.getProjects();

                Console.WriteLine("Project ID\t\t\t\tProject Name");
                Console.WriteLine("----------\t\t\t\t------------");
                foreach (Project project in projects)
                {
                    Console.WriteLine(string.Format("{0}\t{1}",
project.project_id, project.name));
                }

                Console.WriteLine("Retrieving list of Checklists for first
project\n\n");


                api.DefaultProject = projects[0];

                List<Checklist> checklists = api.getChecklists(null, null,
0, 10000); // Defaults to 25
                Console.WriteLine(string.Format("The project {0} has {1}
checklist(s)", api.DefaultProject.name, checklists.Count));

                if (checklists.Count > 0)
                {
                    Console.WriteLine("Checklist ID\t\t\t\tName");
                    Console.WriteLine("------------\t\t\t\t----");

                    foreach (Checklist checklist in checklists)
                    {
                        Console.WriteLine(string.Format("{0}\t{1}",
checklist.id, checklist.name));
                    }

                    Console.WriteLine("\n\n");

                    Checklist firstChecklist =
api.getChecklist(checklists[0].id);
                    Console.WriteLine(string.Format("The checklist with ID
{0} has {1} sections. Please inspect this object to see what else is
available!", firstChecklist.id, firstChecklist.sections.Count));
                }
            }
            catch (BIM360FieldAPIException ex)
            {
                Console.WriteLine(string.Format("API service threw an
exception: {0} {1}", ex.Code, ex.Message));
            }
            catch (UnauthorizedAccessException ua)
            {
                Console.WriteLine("Failed to authenticate with the supplied
credentials.");
            }
        }
    }
}
```

Code Snippet for extracting 4D model information

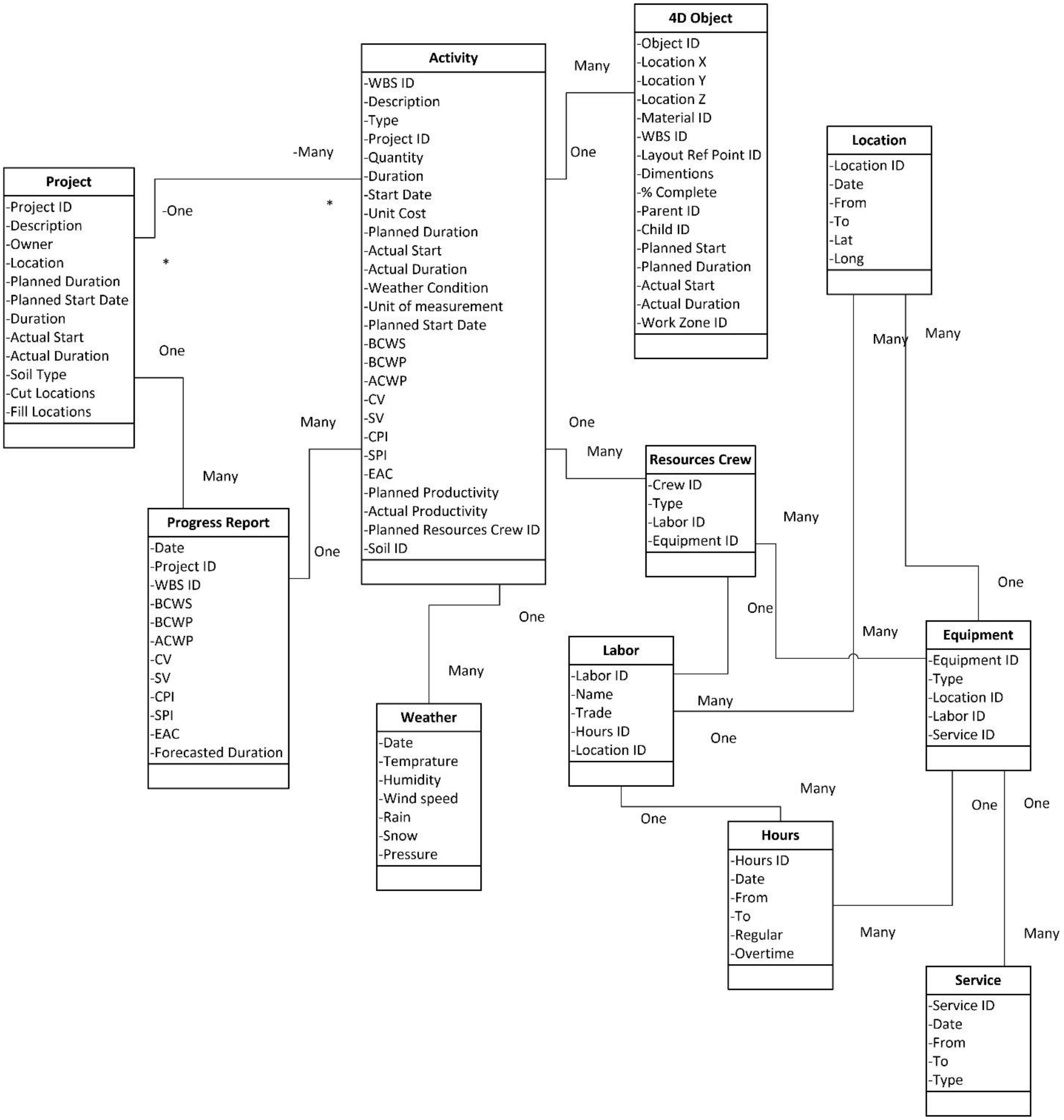
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Services;
using System.IO;
using System.Text;

namespace DataExtraction
{
  public partial class project : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      BIM360WebServiceAPI apiObj = new BIM360WebServiceAPI(Request);

      // If no logged on, just redirect to the home page
      if (!apiObj.userLoggedIn)
      {
        string redirURL = BIM360WebServiceAPI.GetBaseURL() +
"/default.aspx";
        Response.Redirect(redirURL);
      }

      // Get the Project ID from the URL.. if not valid, just display
message
      string projectID = Request.Params["id"];
      if (projectID == "")
      {
        this.page_header.InnerHtml = "<h1>Invalid Project ID</b>: [" +
projectID + "]</h1>";
        return;
      }
      project_info_response_v1 tProj = apiObj.getProjectInfo(projectID);
      if (tProj != null)
      {
        string tHead = "<h1>Project: " +
HttpUtility.UrlDecode(tProj.project_name) + " [ID=" + projectID + "]</h1>";
        this.page_header.InnerHtml = tHead;

        string tHead2 = "<b>Project Created: </b>" + tProj.created_date + "
<b>Roster Count: </b>" + tProj.project_roster.Count();
        tHead2 += " <a class=\"roster_link\" id=\"roster_link\"
href=\"javascript:void();\" onClick=\"viewProjectRoster('" + projectID +
"');\">(View Roster)</a>";
        this.page_sub_header.InnerHtml = tHead2;
      }
      else
      {
        this.page_header.InnerHtml = "<h1>Project: [ID=" + projectID +
"]</h1>";
      }
      string tJS = "";
      tJS += "<script>";
      tJS += "loadProjectTree(\"" + projectID + "\");";
      tJS += "</script>";
      this.page_contents.InnerHtml = tJS;
    }

    [WebMethod]
    static public string ajax_GetProjectTree()
    {
      string buildHTML = "";
      HttpRequest aRequest = HttpContext.Current.Request;
      BIM360WebServiceAPI apiObj = new BIM360WebServiceAPI(aRequest);

      // If no logged on, just redirect to the home page
      if (!apiObj.userLoggedIn)
      {
        return "[]";
      }

      // Get the ID
      string projectID = aRequest.Params["id"];
      HttpContext.Current.Response.ContentType = "application/json;
charset=UTF-8;";

      // Build project list...
      buildHTML += apiObj.getProjectTreeView(projectID);
      return buildHTML;
    }

    [WebMethod]
    static public string ajax_GetModelInfo()
    {
      string buildHTML = "";
      HttpRequest aRequest = HttpContext.Current.Request;
      BIM360WebServiceAPI apiObj = new BIM360WebServiceAPI(aRequest);

      // If no logged on, just redirect to the home page
      if (!apiObj.userLoggedIn)
      {
        return "<b>Unauthorized: Please login to continue</b>";
      }

      // Get the ID
      string modelID = aRequest.Params["id"];

      // Get the model info...
      model_info_response_v1 tModel = apiObj.getModelInfo(modelID);
      if (tModel == null)
      {
        return "<b>Model Not Found</b>";
      }

      buildHTML += "<center>";
      buildHTML += "<table width=500 style=\"border: 1px solid #CCCCCC;\">";
      buildHTML += "<tr bgcolor=\"#CCCCCC\">";
      buildHTML += "<td><b>Attribute</b></td>";
      buildHTML += "<td><b>Value</b></td>";
      buildHTML += "</tr>";
      buildHTML += addRow("company_id", tModel.company_id);
      buildHTML += addRow("project_id", tModel.project_id);
      buildHTML += addRow("model_name", tModel.model_name);
      buildHTML += addRow("model_id", tModel.model_id);
      buildHTML += addRow("model_version", tModel.model_version.ToString());
      buildHTML += addRow("model_version_id", tModel.model_version_id);
      buildHTML += addRow("is_merged_model",
tModel.is_merged_model.ToString());
      buildHTML += addRow("action_id", tModel.action_id);
      buildHTML += addRow("created_by", tModel.created_by);
      buildHTML += addRow("created_date", tModel.created_date);
      buildHTML += addRow("modified_by", tModel.modified_by);
      buildHTML += addRow("modified_date", tModel.modified_date);
      buildHTML += addRow("parent_folder_id", tModel.parent_folder_id);
      buildHTML += addRow("file_parsed_status",
tModel.file_parsed_status.ToString());

      // Build the URL to view the model
      string timestamp =
BIM360WebServiceAPI.getUNIXEpochTimestamp().ToString();
      string tURL = "";
      tURL += BIM360SDKDeveloperConfig.GLUE_VIEWER_BASE_URL;

      // Add question mark if needed
      if (tURL.Substring(tURL.Length - 1) != "?")
      {
        tURL += "?";
      }

      // Parameters for viewer 2

      tURL += "<br/>api_key=" +
BIM360SDKDeveloperConfig.BIM360GLUESDK_API_KEY;
      tURL += "<br/>&amp;timestamp=" + timestamp;
      tURL += "<br/>&amp;sig=" +
BIM360WebServiceAPI.generateAPISignature(timestamp);
      tURL += "<br/>&company_id=" +
BIM360SDKDeveloperConfig.BIM360GLUESDK_COMPANY_ID;
      tURL += "<br/>&amp;auth_token=" + apiObj.auth_token;
      tURL += "<br/>&runner=embedded/#" +
BIM360SDKDeveloperConfig.BIM360GLUESDK_COMPANY_ID
        + "/action" + "/" + tModel.action_id;

      buildHTML += addRow("View URL", tURL);

      buildHTML += "</table>";
      return buildHTML;
    }

    static string addRow(string aField, string aVal)
    {
      string rHTML = "";
      rHTML += "<tr style=\"border-bottom: 1px solid #CCCCCC\">";
      rHTML += "<td style=\"border-right: 1px solid #CCCCCC\">";
        rHTML += "<b>" + aField + "</b>";
      rHTML += "</td>";

      rHTML += "<td>";
      rHTML += aVal;
      rHTML += "</td>";
     rHTML += "</tr>";
     return rHTML;
    }

    [WebMethod]
    static public string ajax_GetProjectRoster()
    {
     string buildHTML = "";
     HttpRequest aRequest = HttpContext.Current.Request;
     BIM360WebServiceAPI apiObj = new BIM360WebServiceAPI(aRequest);

     // If no logged on, just redirect to the home page
     if (!apiObj.userLoggedIn)
     {
      return "<b>Unauthorized: Please login to continue</b>";
     }

     // Get the ID
     string projectID = aRequest.Params["id"];

     // Get the model info...
     project_info_response_v1 tProj = apiObj.getProjectInfo(projectID);
     if ((tProj == null) || (tProj.project_roster == null))
     {
      return "<b>Roster Not Found</b>";
     }

     buildHTML += "<center>";
     buildHTML += "<table width=500 style=\"border: 1px solid #CCCCCC;\">";
     buildHTML += "<tr bgcolor=\"#CCCCCC\">";
     buildHTML += "<td><b>Login Name</b></td>";
     buildHTML += "<td><b>Date Added</b></td>";
     buildHTML += "</tr>";

     foreach (user_info_response_v1 tUser in tProj.project_roster)
     {
      buildHTML += addRow(tUser.login_name, tUser.created_date);
     }

     buildHTML += "</table>";

     return buildHTML;
    }

    [WebMethod]
    static public string ajax_GetModelViews()
    {
     string buildHTML = "";
     HttpRequest aRequest = HttpContext.Current.Request;
     BIM360WebServiceAPI apiObj = new BIM360WebServiceAPI(aRequest);

     // If no logged on, just redirect to the home page
```

```
      if (!apiObj.userLoggedIn)
      {
        return "<b>Unauthorized: Please login to continue</b>";
      }

      // Get the ID
      string modelID = aRequest.Params["id"];

      // Get the model info...
      model_info_response_v1 tModel = apiObj.getModelInfo(modelID);
      if (tModel == null)
      {
        return "<b>Model Not Found</b>";
      }

      if (tModel.view_tree == null)
      {
        return "<div class=\"message notice\" style=\"margin: 0px 16px 6px
16px;\"><b>This model does not contain any Views.</b></div>";
      }

      buildHTML += "<center>";
      buildHTML += "<table width=500 style=\"border: 1px solid #CCCCCC;\">";
      buildHTML += "<tr bgcolor=\"#CCCCCC\">";
      buildHTML += "<td><b>Name</b></td>";
      buildHTML += "<td><b>Create Date</b></td>";
      buildHTML += "<td><b>Creator</b></td>";
      buildHTML += "</tr>";

      foreach (model_view_node tView in tModel.view_tree)
      {
        if (tView.type != "VIEW")
        {
          continue;
        }

        buildHTML += "<tr style=\"border-bottom: 1px solid #CCCCCC\">";

        buildHTML += "<td style=\"border-right: 1px solid #CCCCCC\">";

          // Build the URL to view the model
          string timestamp =
BIM360WebServiceAPI.getUNIXEpochTimestamp().ToString();
          string tURL = "";
          tURL += BIM360SDKDeveloperConfig.GLUE_VIEWER_BASE_URL;

          // Add question mark if needed
          if (tURL.Substring(tURL.Length - 1) != "?")
          {
            tURL += "?";
          }

          // Set parameters for viewer 2

          tURL += "api_key=" +
BIM360SDKDeveloperConfig.BIM360GLUESDK_API_KEY;
          tURL += "&timestamp=" + timestamp;
```

```
     tURL += "&sig=" +
BIM360WebServiceAPI.generateAPISignature(timestamp);
     tURL += "&company_id=" +
BIM360SDKDeveloperConfig.BIM360GLUESDK_COMPANY_ID;
     tURL += "&auth_token=" + apiObj.auth_token;
     tURL += "&runner=embedded/#" +
BIM360SDKDeveloperConfig.BIM360GLUESDK_COMPANY_ID
      + "/action" + "/" + tView.action_id;

     buildHTML += "<a href=\"javascript:void(0);\"
onClick=\"loadModel('" + tURL + "');\">";
     buildHTML += HttpUtility.UrlDecode(tView.name);
     buildHTML += "</a>";

     buildHTML += "</td>";

     buildHTML += "<td style=\"border-right: 1px solid #CCCCCC\">";
     buildHTML += tView.created_date;
     buildHTML += "</td>";

     buildHTML += "<td>";
     buildHTML += tView.created_by;
     buildHTML += "</td>";
    buildHTML += "</tr>";
   }

   buildHTML += "</table>";
   return buildHTML;
  }

  [WebMethod]
  static public string ajax_GetModelMarkups()
  {
   string buildHTML = "";
   HttpRequest aRequest = HttpContext.Current.Request;
   BIM360WebServiceAPI apiObj = new BIM360WebServiceAPI(aRequest);

   // If no logged on, just redirect to the home page
   if (!apiObj.userLoggedIn)
   {
    return "<b>Unauthorized: Please login to continue</b>";
   }

   // Get the ID
   string modelID = aRequest.Params["id"];

   // Get the model info...
   model_markup[] tMarkups = apiObj.getAllModelMarkups(modelID);
   if (tMarkups == null)
   {
    return "<div class=\"message notice\" style=\"margin: 0px 16px 6px
16px;\"><b>This model does not contain any Markups.</b></div>";
   }

   buildHTML += "<center>";
   buildHTML += "<table width=500 style=\"border: 1px solid #CCCCCC;\">";
   buildHTML += "<tr bgcolor=\"#CCCCCC\">";
```

```
      buildHTML += "<td><b>Name</b></td>";
      buildHTML += "<td><b>Create Date</b></td>";
      buildHTML += "<td><b>Creator</b></td>";
      buildHTML += "</tr>";

      // Show the markups
      foreach (model_markup aMarkup in tMarkups)
      {
        buildHTML += "<tr style=\"border-bottom: 1px solid #CCCCCC\">";

        buildHTML += "<td style=\"border-right: 1px solid #CCCCCC\">";

        // Build the URL to view the model
        string timestamp =
BIM360WebServiceAPI.getUNIXEpochTimestamp().ToString();
        string tURL = "";
        tURL += BIM360SDKDeveloperConfig.GLUE_VIEWER_BASE_URL;

        // Add question mark if needed
        if (tURL.Substring(tURL.Length - 1) != "?")
        {
          tURL += "?";
        }

        // Set parameters for viewer 2

        tURL += "api_key=" + BIM360SDKDeveloperConfig.BIM360GLUESDK_API_KEY;
        tURL += "&timestamp=" + timestamp;
        tURL += "&sig=" +
BIM360WebServiceAPI.generateAPISignature(timestamp);
        tURL += "&company_id=" +
BIM360SDKDeveloperConfig.BIM360GLUESDK_COMPANY_ID;
        tURL += "&auth_token=" + apiObj.auth_token;
        tURL += "&runner=embedded/#" +
BIM360SDKDeveloperConfig.BIM360GLUESDK_COMPANY_ID
          + "/action" + "/" + aMarkup.action_id;
        buildHTML += "<a href=\"javascript:void(0);\" onClick=\"loadModel('"
+ tURL + "');\">";
        buildHTML += HttpUtility.UrlDecode(aMarkup.name);
        buildHTML += "</a>";

        buildHTML += "</td>";

        buildHTML += "<td style=\"border-right: 1px solid #CCCCCC\">";
        buildHTML += aMarkup.created_date;
        buildHTML += "</td>";

        buildHTML += "<td>";
        buildHTML += aMarkup.created_by;
        buildHTML += "</td>";
        buildHTML += "</tr>";
      }
      buildHTML += "</table>";
      return buildHTML;
    }
  }
}
```

Appendix C: DATABASE STRUCTURE



Appendix D: AS-BUILT SCHEDULES GENERATION

The as-built schedules generation algorithm was developed in collaboration with a major utility company in Quebec (Ibrahim et al., 2013). This appendix describe in details the automated generation of as-built schedules. The work on sites was monitored daily with the inspectors and using the iPad portable computers. These electronic inspection reports are collected and stored in a SQL database. The goal of this project is to develop a program that able to link the progress data from SQL database with the Primavera project as planned and generate the as built schedule. Two major developments were achieved in this project. First, a revised version of the SQL database was produced. Second, a computer program tool for generating the as built schedule was developed. In order to establish the link between the inspection reports database and the primavera schedule, two new fields were added to the SQL database. The two new fields were the activity identification as per the primavera schedule, and The work breakdown structure code based on the primavera schedule.

The new added fields facilitated the linking and the data exchange between the SQL database and the primavera schedule. The addition of these fields is done automatically by the developed software, using the following SQL queries:

```sql
if not exists (select column_name from HQ2.INFORMATION_SCHEMA.COLUMNS
where table_name = 'tbl_Rapport_Journalier' and column_name = 'WBS')
  alter table [HQ].[dbo].[tbl_Rapport_Journalier] add WBS varchar(50)

if not exists (select column_name from HQ2.INFORMATION_SCHEMA.COLUMNS
where table_name = 'tbl_Rapport_Journalier' and column_name = 'ACT_ID')
  alter table [HQ].[dbo].[tbl_Rapport_Journalier] add ACT_ID varchar(50)
```

Each inspection report in the database includes the following information:

- Type of the report (NoRapport),
- Location of the work (NoLocalisation) , and
- Task description (NoActivite).

These three pieces of information were used to execute a three-phase scanning algorithm as shown in Figure D-1.

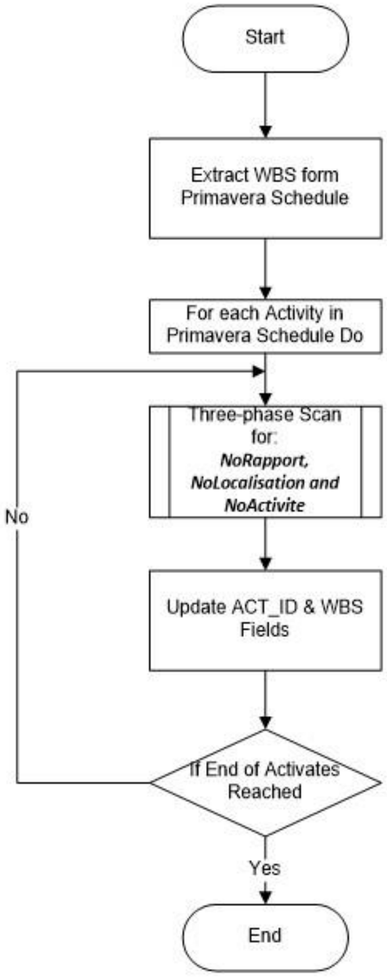

Start
Extract WBS form Primavera Schedule
For each Activity in Primavera Schedule Do
Three-phase Scan for: NoRapport, NoLocalisation and NoActivite
Update ACT_ID & WBS Fields
If End of Activates Reached
No
Yes
End

Figure D-1: Database Update Algorithm Flowchart

The Update of the database Fields is executed by the SQL query:

```
UPDATE HQ.dbo.tbl_Rapport_Journalier
SET tbl_Rapport_Journalier.WBS = '1LS.6.6802.1200.211',
tbl_Rapport_Journalier.ACT_ID = 'D180'
WHERE NoActivite='E356F399-7AE0-445B-8143-CDD1CFB66018'
and NoLocalisation='D4705505-2BFD-43B6-ADB6-7FAC46FB11EC'
and NoRapport='2BC0F785-13B0-4557-9425-7D29BD4B667C'
```

The Automated schedule generation is based on the actual data stored in the SQL database. After implementing the linking fields in the database as explained in the previous section, the process of extracting the actual data from the database is initiated.

The extracted data are:

- The Actual start and finish dates for the activities.
- The Actual assigned resources for each activity.
- The Actual labor and equipment working hours.
- The Actual placed concrete quantities.

The data extraction algorithm based on aggregation process, which is applied on the records in the database. As the database is a relational database, the required data is extracted by joining multiple tables using SQL queries.

Actual labor resources assignments query:

```
select tbl_Rapport_Journalier.ACT_ID, tbl_Code_Main_Oeuvre.CorpsMetier,
min(tbl_Rapport_Journalier.DateRapport) as Actual_start,
max(tbl_Rapport_Journalier.DateRapport) as Actual_Finish,
sum(tbl_Main_Oeuvre.Nombre * tbl_Main_Oeuvre.NbreHreRegulier +
tbl_Main_Oeuvre.Nombre * tbl_Main_Oeuvre.NbreHreSup * 1.5)  as Total_ManHrs
from HQ.dbo.tbl_Rapport_Journalier
left join HQ.dbo.tbl_Activite on
tbl_Rapport_Journalier.NoActivite=tbl_Activite.NoActivite
left join HQ.dbo.tbl_Main_Oeuvre on
tbl_Rapport_Journalier.NoFormulaire=tbl_Main_Oeuvre.NoFormulaire
left join HQ.dbo.tbl_Code_Main_Oeuvre on tbl_Main_Oeuvre.NoCodeMainOeuvre =
tbl_Code_Main_Oeuvre.NoCodeMainOeuvre
where(tbl_Rapport_Journalier.ACT_ID Is Not NULL) And
(tbl_Code_Main_Oeuvre.CorpsMetier Is Not NULL)
group by tbl_Rapport_Journalier.ACT_ID, tbl_Code_Main_Oeuvre.CorpsMetier
order by tbl_Rapport_Journalier.ACT_ID
```

Actual equipment resources assignments query:

```
select tbl_Rapport_Journalier.ACT_ID, tbl_Code_Equipement.[Description],
min(tbl_Rapport_Journalier.DateRapport) as Actual_start,
max(tbl_Rapport_Journalier.DateRapport) as Actual_Finish,
sum(HreTravaille)  as Total_ManHrs from HQ2.dbo.tbl_Rapport_Journalier
left join HQ2.dbo.tbl_Equipement on tbl_Rapport_Journalier.NoFormulaire =
tbl_Equipement.NoFormulaire
left join HQ2.dbo.tbl_Code_Equipement on
tbl_Equipement.NoCodeEquipement=tbl_Code_Equipement.NoCodeEquipement
left join HQ2.dbo.tbl_Type_Equipement on
tbl_Code_Equipement.NoTypeEquipement=tbl_Type_Equipement.NoTypeEquipement
where (tbl_Rapport_Journalier.ACT_ID Is Not NULL) And
(tbl_Code_Equipement.[Description] Is Not NULL)
group by tbl_Rapport_Journalier.ACT_ID, tbl_Code_Equipement.[Description],
tbl_Code_Equipement.NoEquipement
order by tbl_Rapport_Journalier.ACT_ID
```

Actual Placed Concrete Quantities Query:

```
select ACT_ID, 'Beton' as Resource ,MIN(tbl_Rapport_Journalier.DateRapport)
as Actual_Start ,MAX(tbl_Rapport_Journalier.DateRapport) as Actual_Finish
,SUM(tbl_Beton_Place.Quantite) as Total_Qty
from HQ2.dbo.tbl_Rapport_Journalier
left join HQ2.dbo.tbl_Beton on
tbl_Rapport_Journalier.NoFormulaire=tbl_Beton.NoFormulaire
left join HQ2.dbo.tbl_Beton_Place on
tbl_Beton.NoBeton=tbl_Beton_Place.NoBeton
where tbl_Beton_Place.Quantite > 0
group by ACT_ID
```

Actual Activity Start and Finish Dates Query:

```
select tbl_Rapport_Journalier.ACT_ID,
max(tbl_Rapport_Journalier.DateRapport) as Actual_Finish,
min(tbl_Rapport_Journalier.DateRapport) as Actual_Start
from HQ2.dbo.tbl_Rapport_Journalier
where tbl_Rapport_Journalier.ACT_ID Is Not NULL
group by tbl_Rapport_Journalier.ACT_ID
order by tbl_Rapport_Journalier.ACT_ID
```

The resources Cross check algorithm is responsible for identifying any mismatch between the allocated resources to the planned schedule and the actual resources utilized in the actual inspection reports. If the resources were not assigned in the planned schedule, the algorithm generates a list of the new resources, and creates them in order to assign them in the as-built schedule as per the actual data. The actual resources list is extracted from the database using the following queries:

```
Select tbl_Code_Main_Oeuvre.CorpsMetier
from HQ2.dbo.tbl_Rapport_Journalier
left join HQ2.dbo.tbl_Main_Oeuvre on
tbl_Rapport_Journalier.NoFormulaire=tbl_Main_Oeuvre.NoFormulaire
left join HQ2.dbo.tbl_Code_Main_Oeuvre on tbl_Main_Oeuvre.NoCodeMainOeuvre =
tbl_Code_Main_Oeuvre.NoCodeMainOeuvre
where ACT_ID Is Not null and tbl_Code_Main_Oeuvre.CorpsMetier is not null
group by tbl_Code_Main_Oeuvre.CorpsMetier

select tbl_Code_Equipement.Description
from HQ2.dbo.tbl_Rapport_Journalier
left join HQ2.dbo.tbl_Equipement on tbl_Rapport_Journalier.NoFormulaire =
tbl_Equipement.NoFormulaire
left join HQ2.dbo.tbl_Code_Equipement on tbl_Equipement.NoCodeEquipement =
tbl_Code_Equipement.NoCodeEquipement
left join HQ2.dbo.tbl_Type_Equipement on
tbl_Code_Equipement.NoTypeEquipement = tbl_Type_Equipement.NoTypeEquipement
where ACT_ID Is Not null and tbl_Code_Equipement.Description is not null
group by tbl_Code_Equipement.Description
```

The as-built schedule is then generated according to the actual data from the inspection reports in the SQL database. The schedule generation algorithm update the planned schedule with the actual progress from the extracted data in the previous section as illustrated in Figure D-2.

The software was developed in Microsoft Visual Basic.net environment. The developed tool generates the as-built schedule in primavera .Xer format. The Program requires two main inputs: as Planned Schedule in Primavera .xer format, and site inspection reports database (SQL format).

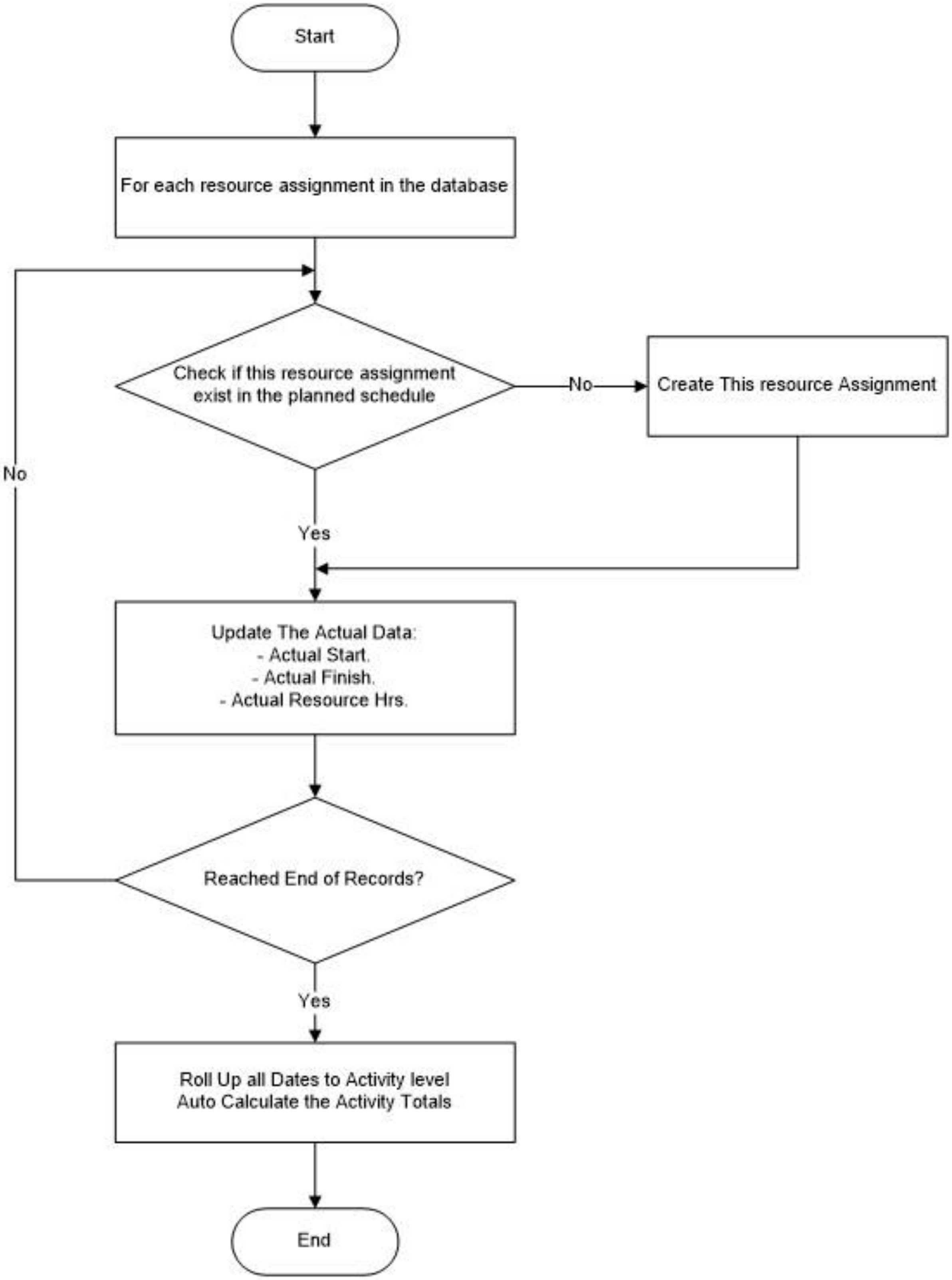

Start
For each resource assignment in the database
Check if this resource assignment exist in the planned schedule
No
Create This resource Assignment
No
Yes
Update The Actual Data:
- Actual Start.
- Actual Finish.
- Actual Resource Hrs.
Reached End of Records?
Yes
Roll Up all Dates to Activity level
Auto Calculate the Activity Totals
End

Figure D-2: As-built Schedule Generation Algorithm

Appendix E: SELF-ADAPTIVE FORECASTING

Adaptive filtering is a technique, which had been widely used in telecommunications to forecast and to build mathematical models of unknown dynamic systems, where the parameters of the filter are updated continuously to achieve better estimates for output signals. Figure (E-1) depicts the general filter setup with its inputs and outputs. The filter self-leaning process is defined by adjusting the filter parameters to reduce the error between the output signal y(t) and the estimated signal Y'(t). As the input data x(t) and output data y(t) changes, the filter adapts to the new values by generating a new values for its parameters. As a result, when the estimation error e(t) become smaller, the adaptive filter output converges to the unknown system performance. A one-step look ahead forecast is estimated based on the weighted sum of past observations. In general terms, this approach is represented as:

$$Y'_{t+1} = \sum_{i=1}^{n} w_i x_i \qquad \text{(E-1)}$$

Where Y'_{t+1} is the forecast at the end of period t + 1; w_i, the weight assigned to observation i; x_i, the ith observed value, and n, the number of periods.

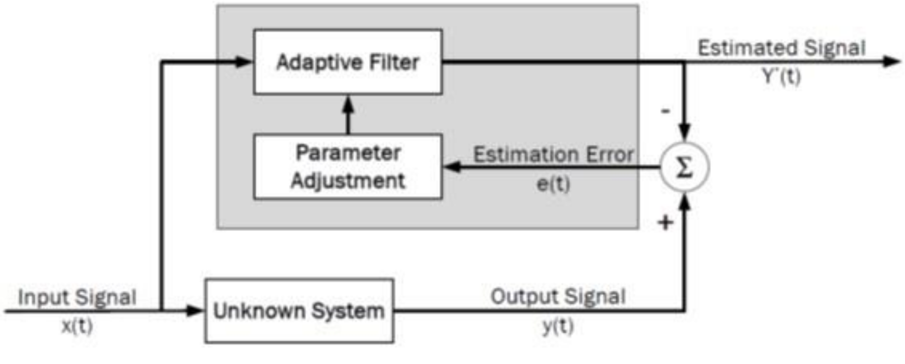


Figure E-1: Self-Adaptive Filtering General Model

The forecasting process starts by obtaining a series of n past observations of the variable to be forecasted. Then, an initial value for each of the weights is assumed to be equal, and a forecast, Y'_{t+1}, is made using Equation (E-1). The forecasted value is compared later (at the next reporting period) with the actual value recorded at that time period, and hence the forecast error is computed to adjust the weights subsequently and hence reduce the error of the next forecast. This process is repeated for each of the following time intervals. This technique is reliable in forecasting even in situations where relatively small amount of data is available. The fundamentals of adaptive filtering are not only technically sound, but also can be explained in an intuitively appealing manner to management. The original work on adaptive filters is attributed to (Pertuz, 1968; Widrow, 1966).

The self-adaptive filtering, explained in the previous section, is utilized in the developed forecasting method using periodic actual and planned costs data as inputs for the iterative self-learning process. The block diagram presented in Figure (E-2) illustrates the developed self-adaptive cost forecasting method.

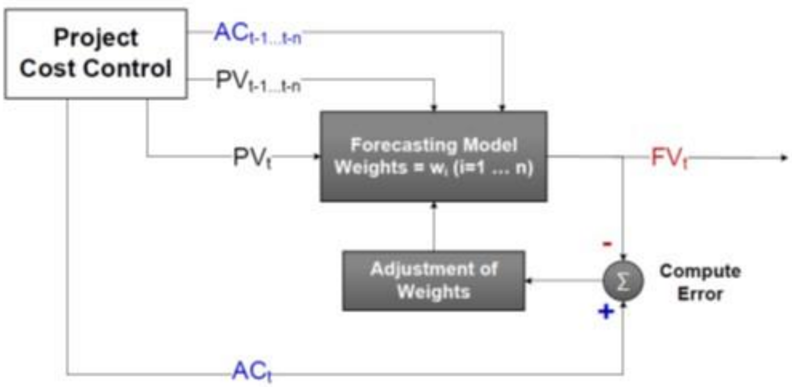


Figure E-2: Developed Self-Adaptive Filtering Cost Forecasting Method

In this method, a one-step-ahead cost forecast is calculated as weighted sum of the n most recent observations values, using the following equation:

$$FV_t = \sum_{i=0}^{n} w_i\,(\mathrm{k}_{\alpha\mathrm{i}} \times \alpha_\mathrm{i} \times \mathrm{PV}_{\mathrm{t-i}} + \mathrm{k}_{\beta\mathrm{i}} \times \beta_\mathrm{i} \times \mathrm{FV}_{\mathrm{t-1}}) \qquad \text{(E-2)}$$

Where FV_t is the forecasted cost of work performed at the end of period t, W_i is the weight for the i^{th} forecast, $\mathrm{k}_{\alpha\mathrm{i}}$ and $\mathrm{k}_{\beta\mathrm{i}}$ are adjustment coefficients for the i^{th} forecast, α_i is the forecasting factor at the i^{th} previous observation, β_i is the forecasted cost correction factor for the i^{th} forecast and $\mathrm{PV}_{\mathrm{t\text{-}i}}$ is Planned Cost of work scheduled for periods t to t-n.

Figure (E-3) to Figure (E-6) illustrate the developed iterative mechanism for calculating the weights and coefficients for the developed self-adaptive filter form the last n observations. The sequence of the developed iterative mechanism shown in Figure (E-3), is explained in the following four steps:

Step 1: The process start at the end of reporting period t-1, by computing a forecast factor α for each past observation as shown in Figure (E-3), and forecasting the actual cost at the following time step t using n equal weights.

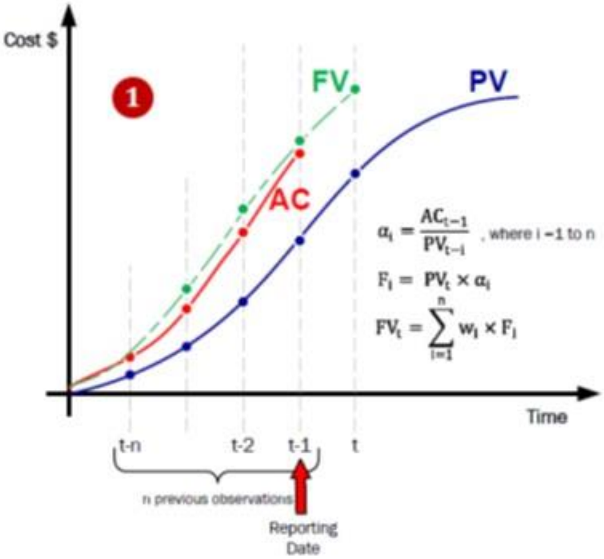

Cost $
1
FV
PV
AC
$\alpha_i = \frac{AC_{t-1}}{PV_{t-i}}$, where i –1 to n
$F_i = PV_t \times \alpha_i$
$FV_t = \sum_{i=1}^{n} w_i \times F_i$
Time
t-n
t-2
t-1
t
n previous observations
Reporting Date

Figure E-3: Step 1 of the Iterative Mechanism

At the end of time period t-1 calculate the forecasting factor:

$$\alpha_i = \frac{AC_{t-1}}{PV_{t-i}} \tag{E-3}$$

Where AC_{t-1} = Actual cost of work performed in period (t-1); PV_{t-i} = Planned cost of work scheduled for periods t-1 to t-n; n = number of previous observation used in the forecast; i = 1 to n, and α_i = the forecast factor at the ith previous observation.

Calculate the forecasted cost at end of period t (next observation):

$$F_i = PV_t \times \alpha_i \tag{E-4}$$

$$FV_t = \sum_{i=1}^{n} w_i \times F_i \tag{E-5}$$

Where PV_t = planned cost of work scheduled for next period; α_i= the forecast factor at the ith previous observation; F_i = the ith forecasted cost of work performed based on the ith observation; W_i = the weight for the ith Forecast (equal to 1/n for the first cycle), and FV_t = forecasted cost of work performed in period t.

Step 2: At the end of reporting period t, and after the actual cost become know, the relative weights for next forecast are calculated based on previous forecast error as shown in Figure (E-4). The key to the adaptive filtering effectiveness is the rule to adapt the weights at the end of each cycle based on the measured error in the forecasted value.

By definition, the error as the difference between the actual cost and the forecasted cost, the weights are calculated as the reciprocal of the forecast error divided by the sum of the reciprocals of all forecast errors.

$$e_i = |AC_t - F_i| \tag{E-6}$$

$$w_i = \frac{\frac{1}{e_i}}{\sum_{i=1}^{n}\frac{1}{e_i}} \tag{E-7}$$

Where AC_t = actual cost of work performed in period t, and e_i = i^{th} forecast error at period t.

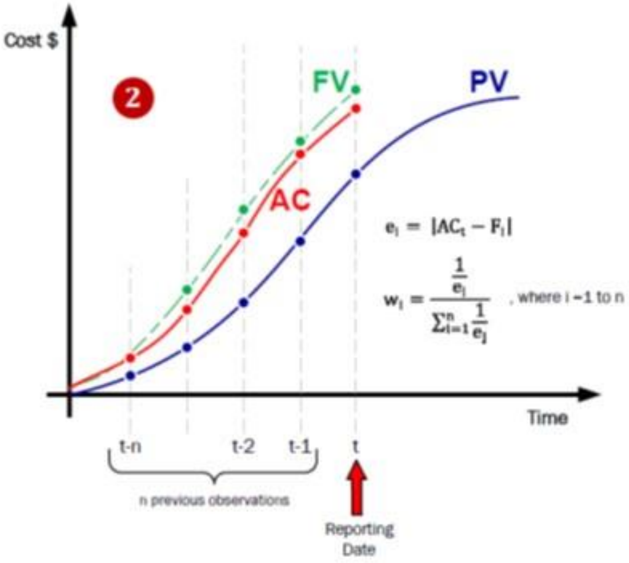


Figure E-4: Step 2 of the Iterative Mechanism

Step 3: The forecasted cost FV_{t+1} using Equation (2) is a function of planned value PV_t during the next time period and the forecasted cost F_t at the end of current time period. Therefor two adjustment coefficients k_α and k_β are calculated to add more weight on the closest value of the F_t and PV_t with respect to the AC. The Accounting Cost Variance (ACV) and the forecast error (FE) are used to calculate the two adjustment coefficients k_α and k_β as shown in Figure (E-5).

These two coefficients are calculated by:

$$k_{\alpha i} = \frac{|AC_t - F_i|}{(|AC_t - PV_t| + |AC_t - F_i|)} \tag{E-8}$$

$$k_{\beta i} = \frac{|AC_t - PV_t|}{(|AC_t - PV_t| + |AC_t - F_i|)} \qquad \text{(E-9)}$$

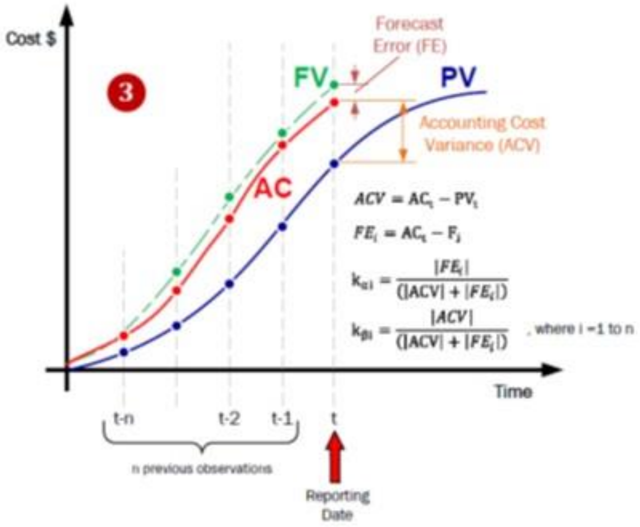


Figure E-5: Step 3 of the Iterative Mechanism

For example, considering a project at the end of period 4 has its AC_4 equals to $900.00, its budgeted cost of work scheduled PV_4 equals $380.00, its predicted cost F_1 using the data of the first period at this reporting date equals $1,300.00. Therefore, coefficient $K_{\alpha 1}$ can be calculated as = |$900.00-$1300.00|/ (|$900.00-$380.00| + |$900.00-$1300.00|) = 0.43. coefficient $K_{\beta 1}$ can be calculated as = |$900.00-$380.00|/(|$900.00-$380.00| + |$900.00-$1300.00|) = 0.57, which means that the next period forecast will yields an adjusted forecast 43% on the next planned value and 57% based on actual cost of last period.

Step 4: In order to account for a situation when the underlying trend in project performance is changing over time, a correction factor β is calculated to periodically compensate for cost variances due to unusual events in a reporting period which is not likely to happen in another period as shown in Figure (E-6). This forecast correction factor is calculated by:

$$\beta_i = \frac{AC_t}{F_i} \qquad \text{(E-10)}$$

Then the forecasting factor for next period is recalculated:

$$\alpha_i = \frac{AC_t}{PV_{t-i+1}} \qquad \text{(E-11)}$$

The forecasted cost for period t+1 is calculated as:

$$F_i = k_{\alpha i} \times \alpha_i \times PV_{t+1} + k_{\beta i} \times \beta_i \times FV_t \qquad \text{(E-12)}$$

$$FV_{t+1} = \sum\nolimits_{i=1}^{n} w_i \times F_i \qquad \text{(E-13)}$$

Where PV_{t+1} = Planned cost of work scheduled at next period; W_i = the weight for the i^{th} Forecast from step 2, and FV_{t+1} = Forecasted cost at next period.

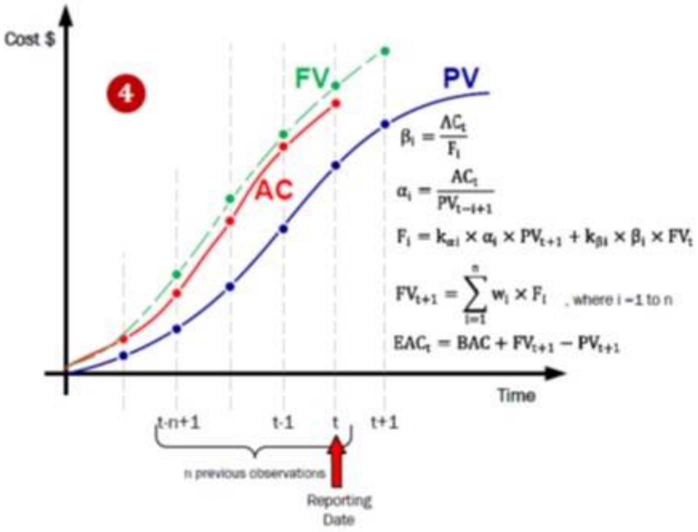


Figure E-6: Step 4 of the Iterative Mechanism

Finally the Project Cost at Completion is then forecasted by:

$$EAC_t = BAC + FV_{t+1} - PV_{t+1} \qquad \text{(E-14)}$$

Where BAC = Planned budget at completion; FV_{t+1} = Forecasted cost at the end of next period; PV_{t+1} = Planned cost of work scheduled in next period, and EAC_t = Estimate at completion at the end of period t. Steps from 2 to 4 are repeated until the end of the project.

Overall, the suitability of the self-adaptive filtering method for project cost forecasting is validated using twelve projects of different type, size and performance. The proposed method was able to forecast the project cost at-completion and at intermediate periods with higher accuracy than methods based on index (EVM) and regression. The enhanced accuracy can be attributed to iterative nature of the proposed method, which calibrates the forecasting factor based on actual and future planned costs. As a result, the methodology developed in this paper contributes to the forecasting techniques used for project control body-of-knowledge.

The proposed method is a simple and accurate, and can be easily integrated in any cost control system. Unlike the index based methods of EVM, the proposed method does not require the collection of periodical progress data (EV and % complete), which in turn saves substantial effort for project management personnel. Also, it can effectively be applied to short-duration project because it only require as little as one data point to forecast project's cost with acceptable accuracy. The proposed method, however requires the availability of project's planned cost data, and accurate actual cost accounting updates.

Appendix F: MATLAB SIMULATION CODE

```
% Localization Raw RSSI
%****************************
x_limit=[-20,20];
y_limit=[-20,20];
num_points=100;
p=zeros(num_points,2);
estimated_p=zeros(num_points,2);
d=zeros(num_points,3);
rssi=zeros(num_points,3);
estimated_d=zeros(num_points,3);
d_error=zeros(num_points,3);
loc_error=zeros(num_points);
r1=[-10,-10];
r2=[-10,10];
r3=[10,10];
error_distance = 0;
error_loc= 0;

for i=1:num_points
    p(i,1)=(x_limit(1)+abs(x_limit(2)-x_limit(1))*rand());
    p(i,2)=(y_limit(1)+abs(y_limit(2)-y_limit(1))*rand());
    d(i,1)= sqrt(((p(i,1)-r1(1))^2)+((p(i,2)-r1(2))^2));
    d(i,2)=sqrt(((p(i,1)-r2(1))^2)+((p(i,2)-r2(2))^2));
    d(i,3)=sqrt(((p(i,1)-r3(1))^2)+((p(i,2)-r3(2))^2));
end

se = RandStream('mt19937ar','Seed',1);
RandStream.setGlobalStream(se);
for i=1:num_points
    for j=1:3
        e = -0.5658*log(d(i,j))+4.0113;
        rssi_error = e*rand(1,100);
        rssi(i,j)= -7.635*log(d(i,j))-40.635+mean(rssi_error);
        estimated_d(i,j) = exp((rssi(i,j)+40.635)/-7.635);
    end
end

for i=1:num_points
   va=((estimated_d(i,2)^2-estimated_d(i,3)^2)-(r2(1)^2-r3(1)^2)-(r2(2)^2-
r3(2)^2))/2;
   vb=((estimated_d(i,2)^2-estimated_d(i,1)^2)-(r2(1)^2-r1(1)^2)-(r2(2)^2-
r1(2)^2))/2;
   estimated_p(i,2)=(vb*(r3(1)-r2(1))-va*(r1(1)-r2(1)))/((r1(2)-
r2(2))*(r3(1)-r2(1))-(r3(2)-r2(2))*(r1(1)-r2(1)));
   estimated_p(i,1)=(va-estimated_p(i,2)*(r3(2)-r2(2)))/(r3(1)-r2(1));

end

for i=1:num_points
   for j=1:3
       d_error(i,j)=abs(d(i,j)-estimated_d(i,j));
       error_distance=d_error(i,j)+error_distance;
   end
   loc_error(i)=sqrt((p(i,1)-estimated_p(i,1))^2+(p(i,2)-
estimated_p(i,2))^2);
   error_loc=loc_error(i)+error_loc;
end

error_distance = error_distance/(num_points*3)
error_loc= error_loc/num_points

figure
plot(p(:,1),p(:,2),'o')
hold;
plot(estimated_p(:,1),estimated_p(:,2),'k+')
plot(r1(1),r1(2),'r*')
plot(r2(1),r2(2),'r*')
plot(r3(1),r3(2),'r*')
legend('Actual Tag Location','Estimated Tag Location','Reader Location');
xlabel('X (m)') % x-axis label
ylabel('Y (m)') % y-axis label
```

```
% RSSI Filtering
% *************************
m=200;
rssi=zeros(m);
x=zeros(m);
d=15;
E = -0.5658*log(d)+4.0113;
PL = -7.635*log(d)-40.635;

R=0.15;
Q=0.00001;
x1=0;
p=1000;

for i=1:m
    rssi(i)= PL+E*rand();
    p1=p+Q;
    y=rssi(i);
    k=p1/(p1+R);
    x(i)=x1+k*(y-x1);
    p=(1-k)*p1;
    x1=x(i);
end

window_size = 9;
simple = tsmovavg(rssi,'s',window_size,1);

j =linspace(0,m,m);
figure
plot(j,rssi(:,1),'b.');

axis([0 m min(rssi(:,1))-1 max(rssi(:,1))+1]);

%axis([0 m -54 -49]);
hold on
```

```
plot(j,x(:,1),'r');
plot(j,simple(:,1),'k');

legend('Raw RSSI','Kalman Filter', 'MVA');
xlabel('Sample') % x-axis label
ylabel('dBm') % y-axis label
```

```
% Localization Kalman Filtered RSSI
%***********************************
x_limit=[-20,20];
y_limit=[-20,20];
num_points=100;
p=zeros(num_points,2);
estimated_p=zeros(num_points,2);
d=zeros(num_points,3);
rssi=zeros(num_points,3);
estimated_d=zeros(num_points,3);
d_error=zeros(num_points,3);
loc_error=zeros(num_points);
r1=[-10,-10];
r2=[-10,10];
r3=[10,10];
error_distance = 0;
error_loc= 0;

P_rssi=zeros(100,1);
x=zeros(100,1);
R=0.15;
Q=0.00001;
x1=0;
pp=1000;
temp_d=zeros(100,1);
k=0;

for i=1:num_points
    p(i,1)=(x_limit(1)+abs(x_limit(2)-x_limit(1))*rand());
    p(i,2)=(y_limit(1)+abs(y_limit(2)-y_limit(1))*rand());
    d(i,1)= sqrt(((p(i,1)-r1(1))^2)+((p(i,2)-r1(2))^2));
    d(i,2)=sqrt(((p(i,1)-r2(1))^2)+((p(i,2)-r2(2))^2));
    d(i,3)=sqrt(((p(i,1)-r3(1))^2)+((p(i,2)-r3(2))^2));
end

se = RandStream('mt19937ar','Seed',1);
RandStream.setGlobalStream(se);
for i=1:num_points
    for j=1:3
        d1=d(i,j);
        E = -0.5658*log(d1)+4.0113;
        PL = -7.635*log(d1)-40.635;
        for n=1:100
            P_rssi(n,1)= PL+E*rand();
            p1=pp+Q;
            y=P_rssi(n,1);
            k=p1/(p1+R);
            x(n,1)=x1+k*(y-x1);
            pp=(1-k)*p1;
            x1=x(n,1);
            temp_d(n,1) = exp((x(n,1)+38.57)/-7.935);
        end
        x1=0;
        pp=1000;
        rssi(i,j)= mean(P_rssi);
        estimated_d(i,j) = mean(temp_d);
    end
end

for i=1:num_points
   va=((estimated_d(i,2)^2-estimated_d(i,3)^2)-(r2(1)^2-r3(1)^2)-(r2(2)^2-
r3(2)^2))/2;
   vb=((estimated_d(i,2)^2-estimated_d(i,1)^2)-(r2(1)^2-r1(1)^2)-(r2(2)^2-
r1(2)^2))/2;
   estimated_p(i,2)=(vb*(r3(1)-r2(1))-va*(r1(1)-r2(1)))/((r1(2)-
r2(2))*(r3(1)-r2(1))-(r3(2)-r2(2))*(r1(1)-r2(1)));
   estimated_p(i,1)=(va-estimated_p(i,2)*(r3(2)-r2(2)))/(r3(1)-r2(1));

end

for i=1:num_points
   for j=1:3
       d_error(i,j)=abs(d(i,j)-estimated_d(i,j));
       error_distance=d_error(i,j)+error_distance;
   end
   loc_error(i)=sqrt((p(i,1)-estimated_p(i,1))^2+(p(i,2)-
estimated_p(i,2))^2);
   error_loc=loc_error(i)+error_loc;
end

error_distance = error_distance/(num_points*3)
error_loc= error_loc/num_points

figure
plot(p(:,1),p(:,2),'o')
hold;
plot(estimated_p(:,1),estimated_p(:,2),'k+')
plot(r1(1),r1(2),'r*')
plot(r2(1),r2(2),'r*')
plot(r3(1),r3(2),'r*')
legend('Actual Tag Location','Estimated Tag Location','Reader Location');
xlabel('X (m)') % x-axis label
ylabel('Y (m)') % y-axis label
```

```
% Localization Kalman Filtered RSSI High Noise Environment
%************************************************************
x_limit=[-20,20];
y_limit=[-20,20];
num_points=100;
p=zeros(num_points,2);
estimated_p=zeros(num_points,2);
d=zeros(num_points,3);
rssi=zeros(num_points,3);
estimated_d=zeros(num_points,3);
d_error=zeros(num_points,3);
loc_error=zeros(num_points);
r1=[-10,-10];
r2=[-10,10];
r3=[10,10];
error_distance = 0;
error_loc= 0;

P_rssi=zeros(100,1);
x=zeros(100,1);
R=0.15;
Q=0.00001;
x1=0;
pp=1000;
temp_d=zeros(100,1);
k=0;

for i=1:num_points
    p(i,1)=(x_limit(1)+abs(x_limit(2)-x_limit(1))*rand());
    p(i,2)=(y_limit(1)+abs(y_limit(2)-y_limit(1))*rand());
    d(i,1)= sqrt(((p(i,1)-r1(1))^2)+((p(i,2)-r1(2))^2));
    d(i,2)=sqrt(((p(i,1)-r2(1))^2)+((p(i,2)-r2(2))^2));
    d(i,3)=sqrt(((p(i,1)-r3(1))^2)+((p(i,2)-r3(2))^2));
end

se = RandStream('mt19937ar','Seed',1);
RandStream.setGlobalStream(se);
for i=1:num_points
    for j=1:3
        d1=d(i,j);
        E = -0.5658*log(d1)+6.0113;
        PL = -7.635*log(d1)-40.635;
        for n=1:100
            P_rssi(n,1)= PL+E*rand();
            p1=pp+Q;
            y=P_rssi(n,1);
            k=p1/(p1+R);
            x(n,1)=x1+k*(y-x1);
            pp=(1-k)*p1;
            x1=x(n,1);
            temp_d(n,1) = exp((x(n,1)+37.6061)/-7.9361);
        end
        x1=0;
        pp=1000;
        rssi(i,j)= mean(P_rssi);
        estimated_d(i,j) = mean(temp_d);
    end
```

```
end

for i=1:num_points
   va=((estimated_d(i,2)^2-estimated_d(i,3)^2)-(r2(1)^2-r3(1)^2)-(r2(2)^2-
r3(2)^2))/2;
   vb=((estimated_d(i,2)^2-estimated_d(i,1)^2)-(r2(1)^2-r1(1)^2)-(r2(2)^2-
r1(2)^2))/2;
   estimated_p(i,2)=(vb*(r3(1)-r2(1))-va*(r1(1)-r2(1)))/((r1(2)-
r2(2))*(r3(1)-r2(1))-(r3(2)-r2(2))*(r1(1)-r2(1)));
   estimated_p(i,1)=(va-estimated_p(i,2)*(r3(2)-r2(2)))/(r3(1)-r2(1));

end

for i=1:num_points
   for j=1:3
       d_error(i,j)=abs(d(i,j)-estimated_d(i,j));
       error_distance=d_error(i,j)+error_distance;
   end
   loc_error(i)=sqrt((p(i,1)-estimated_p(i,1))^2+(p(i,2)-
estimated_p(i,2))^2);
   error_loc=loc_error(i)+error_loc;
end

error_distance = error_distance/(num_points*3)
error_loc= error_loc/num_points

figure
plot(p(:,1),p(:,2),'o')
hold;
plot(estimated_p(:,1),estimated_p(:,2),'k+')
plot(r1(1),r1(2),'r*')
plot(r2(1),r2(2),'r*')
plot(r3(1),r3(2),'r*')
legend('Actual Tag Location','Estimated Tag Location','Reader Location');
xlabel('X (m)') % x-axis label
ylabel('Y (m)') % y-axis label
```

```
% Self-Calibrating Path-Loss
%****************************
population_size= 50;
k_max= 500;
opt_error=zeros(k_max,1);
temp_error=zeros(population_size,1);
w_max=0.9;
w_min=0.4;
c1=2;
c2=2;
old_gen=zeros(population_size,5);
bBest=zeros(population_size,5);
a_max = -30;
a_min = -46;
b_max = -6.35;
b_min =-9.52;
va=0;
```

```
vb=0;
intial_bBest = 1000;
gBest=zeros(1,5);

%intialize population
rand_gen1=randn(population_size,1);
rand_gen2=randn(population_size,1);
old_gen(1,1)= a_min+(abs(a_max-a_min))*rand_gen1(1,1);
old_gen(1,2)= b_min+(abs(b_max-b_min))*rand_gen2(1,1);
old_gen(1,3)= va;
old_gen(1,4)= vb;
old_gen(1,5)= find_error(p,old_gen(1,1),old_gen(1,2));


gBest(1,1)=old_gen(1,1);
gBest(1,2)=old_gen(1,2);
gBest(1,3)=old_gen(1,3);
gBest(1,4)=old_gen(1,4);
gBest(1,5)=old_gen(1,5);


for i=2:population_size

    old_gen(i,1)= a_min+(abs(a_max-a_min))*rand_gen1(i,1);
    old_gen(i,2)= b_min+(abs(b_max-b_min))*rand_gen2(i,1);
    old_gen(i,3)= va;
    old_gen(i,4)= vb;
    old_gen(i,5)= find_error(p,old_gen(i,1),old_gen(i,2));

    if old_gen(i,5)< gBest(1,5)
        gBest(1,1)=old_gen(i,1);
        gBest(1,2)=old_gen(i,2);
        gBest(1,3)=old_gen(i,3);
        gBest(1,4)=old_gen(i,4);
        gBest(1,5)=old_gen(i,5);
    end

end

bBest=old_gen;

for k=1:k_max

   w= w_max-((w_max-w_min)*(k/k_max));

   for i=1:population_size
      va=w*old_gen(i,3)+c1*rand*(bBest(i,1)-
old_gen(i,1))+c2*rand*(gBest(1,1)-old_gen(i,1));
      vb=w*old_gen(i,4)+c1*rand*(bBest(1,2)-
old_gen(i,2))+c2*rand*(gBest(1,2)-old_gen(i,2));
      old_gen(i,1)=old_gen(i,1)+va;
      old_gen(i,2)=old_gen(i,2)+vb;
      old_gen(i,3)= va;
      old_gen(i,4)= vb;
      old_gen(i,5)= find_error(p,old_gen(i,1),old_gen(i,2));
      if old_gen(i,5)< gBest(1,5)
          gBest(1,1)=old_gen(i,1);
          gBest(1,2)=old_gen(i,2);
          gBest(1,3)=old_gen(i,3);
          gBest(1,4)=old_gen(i,4);
          gBest(1,5)=old_gen(i,5);
      end
      if old_gen(i,5)< bBest(i,5)
          bBest(i,1)=old_gen(i,1);
          bBest(i,2)=old_gen(i,2);
          bBest(i,3)=old_gen(i,3);
          bBest(i,4)=old_gen(i,4);
          bBest(i,5)=old_gen(i,5);
      end

   end

   %temp_error=old_gen(:,5);
   opt_error(k,1)=gBest(1,5);

end
```

```
% PSO fitness function
%****************************
function location_error= find_error(ref_points,par_A,par_B)

num_points=length(ref_points);
rssi=zeros(num_points,3);
P_rssi=zeros(100,1);
estimated_d=zeros(num_points,3);
estimated_p=zeros(num_points,2);
loc_error=zeros(num_points);
x=zeros(100,1);
R=0.15;
Q=0.00001;
x1=0;
pp=1000;
temp_d=zeros(100,1);
k=0;
r1=[-10,-10];
r2=[-10,10];
r3=[10,10];
d=zeros(num_points,3);
location_error=0;

for i=1:num_points
    d(i,1)= sqrt(((ref_points(i,1)-r1(1))^2)+((ref_points(i,2)-r1(2))^2));
    d(i,2)=sqrt(((ref_points(i,1)-r2(1))^2)+((ref_points(i,2)-r2(2))^2));
    d(i,3)=sqrt(((ref_points(i,1)-r3(1))^2)+((ref_points(i,2)-r3(2))^2));
end

se = RandStream('mt19937ar','Seed',1);
RandStream.setGlobalStream(se);
    for i=1:num_points
        for j=1:3
```

```
            d1=d(i,j);
            E = -0.5658*log(d1)+6.0113;
            PL = -7.635*log(d1)-40.635;
            for n=1:100
                P_rssi(n,1)= PL+E*rand();
                p1=pp+Q;
                y=P_rssi(n,1);
                k=p1/(p1+R);
                x(n,1)=x1+k*(y-x1);
                pp=(1-k)*p1;
                x1=x(n,1);
                temp_d(n,1) = exp((x(n,1)-par_A)/par_B);
            end
            x1=0;
            pp=1000;
            rssi(i,j)= mean(P_rssi);
            estimated_d(i,j) = mean(temp_d);
        end
    end
    for i=1:num_points
       va=((estimated_d(i,2)^2-estimated_d(i,3)^2)-(r2(1)^2-r3(1)^2)-
(r2(2)^2-r3(2)^2))/2;
       vb=((estimated_d(i,2)^2-estimated_d(i,1)^2)-(r2(1)^2-r1(1)^2)-
(r2(2)^2-r1(2)^2))/2;
       estimated_p(i,2)=(vb*(r3(1)-r2(1))-va*(r1(1)-r2(1)))/((r1(2)-
r2(2))*(r3(1)-r2(1))-(r3(2)-r2(2))*(r1(1)-r2(1)));
       estimated_p(i,1)=(va-estimated_p(i,2)*(r3(2)-r2(2)))/(r3(1)-r2(1));
    end
    for i=1:num_points
       loc_error(i)=sqrt((ref_points(i,1)-
estimated_p(i,1))^2+(ref_points(i,2)-estimated_p(i,2))^2);
       location_error=loc_error(i)+location_error;
    end

    location_error= location_error/num_points;

end
```

Appendix G: IF-Then Rules

Queue Load:

```
entry /
        if ((Return finshed) and (Truck in proximity) and (Location = Cut) and (Truck
        speed = 0) and (Truck weight =0))
        {
                Queue Loading Start = True
        }

do /    Start Counting Queue Loading Duration

exit /
        if ((Queue Loading Start) and (Location = Cut) and (Loader in proximity) and
        (Truck speed =0)
        {
                Queue Loading Finished = True
        }
```

Load:

```
entry /
        if ((Queue Loading finished or Return finshed) and (Loader in proximity) and
        (Loader cycles are detected) and (Location = Cut) and (Truck speed = 0) and
        (Truck weight increasing))
        {
                Loading Start = True
        }

do /    Start Counting Loading Duration

exit /
        if ((Loading Start) and (Location = Road)    and (No Entities in proximity) and
(Truck speed >0)
        {
                Loading Finished = True
        }
```

Travel:

```
entry /
        if (Loading Finished) and (Location = Road) and (Truck speed > 0) and (Truck weight >0))
        {
                Travel Start = True
        }

do /    Start Counting Travel Duration

exit /
        if (Travel Start) and (Location = Fill) and    (Truck or Spotter in proximity) and (Truck speed=0)
        {
                Travel Finished = True
        }
```

Queue Dump:

```
entry /
        if (Travel Finshed) and (Location =Fill) and (Truck speed = 0) and (Truck weight >0)) and (Truck in Proximity) and (Tilt angle =0)
        {
                Queue Dump Start = True
        }

do /    Start Counting Queue Dump Duration

exit /
        if (Queue Dump Start) and (Tilt angle >0) and (Location = Fill) and (Truck weight>0)
        {
                Queue Dump Finished = True
        }
```

Dump:

```
entry /
        if (Travel Finshed or Dump Queue Finished) and (Location =Fill) and (Truck speed = 0) and (Truck weight >0)) and (Spotter in Proximity) and (Tilt angle >0)
        {
                Dump Start = True
        }

do /    Start Counting Dump Duration

exit /
        if (Dump Start) and (Tilt angle =0) and (Location = Fill) and (Truck weight=0)
        {
                Dump Finished = True
        }
```

Return:

```
entry /
        if (Dump Finished) and (Location = Road) and (Truck speed > 0) and (Truck weight >0))
        {
                Return Start = True
        }

do /    Start Counting Return Duration

exit /
        if (Return Start) and (Location = Cut) and (Truck or Loader in proximity) and (Truck speed=0)
        {
                Return Finished = True
        }
```

Out of Service:

```
entry /
        if ((Location = Service or Any) and (Truck speed = 0) and (Time since last state >
                Average Cycle Time)
        {
                Service Start = True
        }

do /    Start Counting Service Duration

exit /
        if (State changed)
        {
                Service Finished = True
        }
```

Appendix H: SOURCE CODE

SA-GPS Code Snippet for Arduino Uno

```
#include <SD.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
// Include the Wire library for I2C access.
#include <Wire.h>
// Include the Electronics BMP180 library.
#include <BMP180.h>

// Store an instance of the BMP180 sensor.
BMP180 barometer;
// Store the current sea level pressure at your location in Pascals.
float seaLevelPressure = 101325;

TinyGPS gps;
SoftwareSerial ss(6,5);
bool firstData = false;
float flat1, flon1;
const int chipSelect = 10;
const int writeLED = 3;
const int removeLED = 4;
const int startLogPin = 2;
int logState = 0;

void setup()
{

  ss.begin(38400);
  pinMode(chipSelect, OUTPUT);
  pinMode(writeLED, OUTPUT);
  pinMode(removeLED, OUTPUT);
  pinMode(startLogPin, INPUT);

  // We start the I2C on the Arduino for communication with the BMP180 sensor.
  Wire.begin();
  // We create an instance of our BMP180 sensor.
  barometer = BMP180();
  if(barometer.EnsureConnected())
  {
    // When we have connected, we reset the device to ensure a clean start.
    barometer.SoftReset();
    // Now we initialize the sensor and pull the calibration data.
    barometer.Initialize();   }
  else
  {
    Serial.println("Could not connect to BMP180.");
    digitalWrite(indicatorLed, LOW); // Set our LED.
  }

  digitalWrite(writeLED, LOW);
  digitalWrite(removeLED, LOW);

  String headerStr = "Date;Time;Lat;Long;Alt;Course;Speed;Sat;Age;Distance;Prec";
  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    //Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  //Serial.println("card initialized.");
  if (SD.exists("datalog.csv")){
    SD.remove("datalog.csv");
  }
  Serial.begin(115200);
  File dataFile = SD.open("datalog.csv", O_CREAT | O_WRITE);
  digitalWrite(writeLED, HIGH);
  if (dataFile) {
    dataFile.print(headerStr);
    dataFile.println();
    dataFile.close();
    digitalWrite(writeLED, LOW);
  }
  // if the file isn't open, pop up an error:
  else {
    Serial.println("error opening datalog.csv");
  }

}

String printDigits(int digits){
  // utility function for digital clock display: prints preceding colon and leading 0
  String str;
  if(digits < 10)
    str = "0" + String(digits);
  else
    str = String(digits);
  return str;
    //Serial.print('0');
  //Serial.print(digits);
}


void loop()
{
  logState = digitalRead(startLogPin);
  if (logState == HIGH)
  {
    SD.begin(chipSelect);
    digitalWrite(removeLED,LOW);
    bool newData = false;


    for (unsigned long start = millis(); millis()-start <1000;)
    {
      while (ss.available())
      {
        char c = ss.read();
        if (gps.encode(c))
        {
          newData = true;
          if (!firstData)
          {
            unsigned long fix_age;
            gps.f_get_position(&flat1, &flon1, &fix_age);
            Serial.println("Intial Data");
            Serial.print("Lat = "); Serial.println(flat1);
            Serial.print("Long = "); Serial.println(flon1);
            Serial.println();
          }
          firstData = true;
        }
      }
    }

    if (newData)
    {
      float flat2, flon2, falt, fc, fkmph, distance;
      unsigned long fix_age, prec;
      unsigned short sat;
      int year;
      byte month, day, hour, minute, second, hundredths;
      String timeStr, dateStr;
      if(barometer.IsConnected)
      {
       // Retrive the current pressure in Pascals.
       long currentPressure = barometer.GetPressure();
      // Retrive the current altitude (in meters).
      float altitude = barometer.GetAltitude(seaLevelPressure);

     }

     gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths, &fix_age);
      timeStr = printDigits((int)hour) + ":" + printDigits((int)minute) + ":" +
printDigits((int)second);
      dateStr = printDigits((int)day) + "/" + printDigits((int)month) + "/" +
printDigits(year);
      gps.f_get_position(&flat2, &flon2, &fix_age);
      falt = gps.f_altitude(); // +/- altitude in meters
      fc = gps.f_course(); // course in degrees
      fkmph = gps.f_speed_kmph(); // speed in km/hr
      sat = gps.satellites();
      distance = gps.distance_between (flat1, flon1, flat2, flon2);
      prec = gps.hdop();

      Serial.print("Date = "); Serial.println(dateStr);
      Serial.print("Time = "); Serial.println(timeStr);
      Serial.print("Lat = "); Serial.println(flat2,6);
      Serial.print("Long = "); Serial.println(flon2,6);
      Serial.print("Alt = "); Serial.println(falt);
      Serial.print("Course = "); Serial.println(fc);
      Serial.print("Speed = "); Serial.println(fkmph);
      Serial.print("Sat = "); Serial.println(sat);
      Serial.print("Age = "); Serial.println(fix_age);
      Serial.print("Distance = "); Serial.println(distance);
      Serial.print("Prec = "); Serial.println(prec);
      Serial.print("Pressure = "); Serial.println(currentPressure,4);
      Serial.print("Altitude = "); Serial.println(altitude,4);
      Serial.println();

      File dataFile = SD.open("datalog.csv", O_CREAT | O_APPEND | O_WRITE);
      if (dataFile) {
        digitalWrite(writeLED, HIGH);
        //dataString = flat & ";" & flon & ";" & gps.satellites() & ";" & gps.hdop();
        dataFile.print(dateStr); dataFile.print(";");
        dataFile.print(timeStr); dataFile.print(";");
        dataFile.print(flat2,6); dataFile.print(";");
        dataFile.print(flon2,6); dataFile.print(";");
        dataFile.print(falt,2); dataFile.print(";");
        dataFile.print(fc,2); dataFile.print(";");
        dataFile.print(fkmph,2); dataFile.print(";");
        dataFile.print(sat); dataFile.print(";");
        dataFile.print(fix_age); dataFile.print(";");
        dataFile.print(distance,2); dataFile.print(";");
        dataFile.print(prec); dataFile.print(";");
        dataFile.print(currentPressure,4); dataFile.print(";");
        dataFile.print(altitude,4); dataFile.print(";");
        dataFile.println();
        dataFile.flush();
        dataFile.close();
        digitalWrite(writeLED, LOW);
      }
      else {
        Serial.println("error opening datalog.csv");
        digitalWrite(removeLED,HIGH);
        digitalWrite(writeLED,HIGH);
        return;
      }
      flat1 = flat2;
      flon1 = flon2;
    }
  }
  else
    digitalWrite(removeLED,HIGH);

}
```

SC-WSN (Tag) Code Snippet

```
# Use Synapse Pin definitions
from synapse.nvparams import *
from synapse.switchboard import *
from synapse.platforms import *

BMP180_ADDRESS = 0x77
seaLevelPressure = 101325
numRegs = 0x34

def intialize():

    #Set transmit power level
    txPwr(17)

    #Disable UART 0
    initUart(0, 0)
    flowControl(0, False)

    #Disable UART 1
    initUart(1, 0)
    flowControl(1, False)

    #Set all GPIOs to inputs with pullup to cut down on current use
    setPinDir(0, 0)
    setPinPullup(0, 1)

    setPinDir(1, 0)
    setPinPullup(1, 1)

    setPinDir(2, 0)
    setPinPullup(2, 1)

    setPinDir(3, 0)
    setPinPullup(3, 1)

    setPinDir(4, 0)
    setPinPullup(4, 1)

    setPinDir(5, 0)
    setPinPullup(5, 1)

    setPinDir(6, 0)
    setPinPullup(6, 1)

    setPinDir(7, 0)
    setPinPullup(7, 1)


    setPinDir(8, 0)
    setPinPullup(8, 1)

    setPinDir(9, 0)
    setPinPullup(9, 1)
    setPinDir(10, 0)
    setPinPullup(10, 1)

    setPinDir(11, 0)
    setPinPullup(11, 1)

    setPinDir(12, 0)
    setPinPullup(12, 1)

    setPinDir(13, 0)
    setPinPullup(13, 1)

    setPinDir(14, 0)
    setPinPullup(14, 1)

    setPinDir(15, 0)
    setPinPullup(15, 1)

    setPinDir(16, 0)
    setPinPullup(16, 1)

    setPinDir(17, 0)
    setPinPullup(17, 1)

    setPinDir(18, 0)
    setPinPullup(18, 1)

    #Disconnect and do not use UARTs
    crossConnect(DS_NULL, DS_PACKET_SERIAL)


    #Turn off node relaying packets for others
    saveNvParam(NV_MESH_OVERRIDE_ID, 1)

def poll():

    #init I2C
    i2cInit(True)
    #Read pressure value
    cmd = chr( BMP180_ADDRESS | 1 )
    result = i2cRead(cmd, numRegs, 1 , False)
    #Calculate altitude
    altitude = 44330*(1-(result/seaLevelPressure)^(0.19029496))
    #Send Ping broadcast
    mcastRpc(1, 1, "tag_ping", localAddr(), altitude)
    #Go back to sleep
    sleep(0, 10)


#Install event handlers
snappyGen.setHook(SnapConstants.HOOK_STARTUP, intialize)
snappyGen.setHook(SnapConstants.HOOK_100MS, poll)
```