Spring 2020

Accurate Range-based Indoor Localization Using PSO-Kalman Filter Fusion

Paul Bupe Jr

ACCURATE RANGE - BASED INDOOR LOCALIZATION USING PSO - KALMAN FILTER FUSION

by

PAUL BUPE JR

(Under the Direction of Pradipta De)

ABSTRACT

Accurate indoor localization often depends on infrastructure support for distance estimation in range-based techniques. One can also trade off accuracy to reduce infrastructure investment by using relative positions of other nodes, as in range-free localization. Even for range-based methods where accurate Ultra-WideBand (UWB) signals are used, non line-of-sight (NLOS) conditions pose significant difficulty in accurate indoor localization. Existing solutions rely on additional measurements from sensors and typically correct the noise using a Kalman filter (KF). Solutions can also be customized to specific environments through extensive profiling. In this work, a range-based indoor localization algorithm called PSO - Kalman Filter Fusion (PKFF) is proposed that minimizes the effects of NLOS on localization error without using additional sensors or profiling. Location estimates from a windowed Particle Swarm Optimization (PSO) and a dynamically adjusted KF are fused based on a weighted variance factor. PKFF achieved a 40 % lower 90-percentile root-mean-square localization error (RMSE) over the standard least squares trilateration algorithm at 61 cm compared to 102 cm.

INDEX WORDS: Indoor localization, Particle swarm optimization, Kalman filter, NLOS, Sensor fusion

ACCURATE RANGE - BASED INDOOR LOCALIZATION USING PSO - KALMAN FILTER FUSION

by

PAUL BUPE JR

B.S., Georgia Southern University, 2013

M.S., Georgia Southern University, 2015

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in

Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ACCURATE RANGE - BASED INDOOR LOCALIZATION USING PSO - KALMAN FILTER FUSION

by

PAUL BUPE JR

Major Professor: Pradipta De
Committee: Andrew Allen
Muralidhar Medidi

Electronic Version Approved:
May 2020

DEDICATION

This thesis is dedicated to my family, starting with my parents Dr. Paul and Catherine Bupe, and my siblings Faith, Hope, and David who have always been a constant source of encouragement. To my nephews Timothy and Josiah, who have been such a joy and source of much laughter and entertainment. And to my friends Elise, Logan, Lauren, Sara, and Jo, who have been a true blessing during my time in Statesboro and so graciously put up with my many ramblings about my thesis.

ACKNOWLEDGMENTS

I would like to thank Dr. Pradipta De for taking on the sometimes arduous task of keeping me on track as I worked through this thesis. His knowledge and insight was invaluable and I could not have completed this work without him. I would also like to acknowledge and thank Dr. Muralidhar Medidi and Dr. Andrew Allen, from whom I've learned so much through the course of my studies. Finally I would like to thank Gene and the rest of the CRI family for being so good to me during my years at Georgia Southern University. And as in all things, I give thanks to God for giving me the knowledge and ability to successfully complete this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Localization is the process of determining the location of an entity, commonly referred to as a node, within an environment. The most widely used method of localization is the global positioning system (GPS). GPS is used in a wide array of applications such as navigation, surveying and mapping, and even the tracking of wild animals for scientific research or conservation. The recent rise of location-aware applications and services has raised the need for accurate localization in indoor environments. There are many indoor applications that greatly benefit from location-awareness including locating assets in factories and warehouses, tracking patrons for indoor guided tours, indoor robotic navigation, and even newer applications like augmented reality. Even with its widespread application and use, GPS is not suitable for indoor localization because it largely relies on line-of-sight (LOS) communication to at least four GPS satellites. In indoor environments these signals are greatly attenuated by walls, roofs, and other such structures which means that GPS often does not work indoors. Even in the case where signals from at least four satellites are received indoors, the localization accuracy is too low to be useful indoors.

In lieu of GPS, a large number of indoor localization techniques and real-time locating systems (RTLS) have been developed. Indoor localization techniques can be grouped into two broad categories: radio-frequency (RF) based, and non-RF based. Non-RF based techniques generally use cameras and sensors like Inertial Measurement Units (IMU) and laser distance finders for localization (Yassin et al., Secondquarter 2017). Cameras can use markers or extract visual features from an environment to perform localization using various techniques including the very common technique known as simultaneous localization and mapping (SLAM). This technique involves building a map of an environment while at the same time localizing in that environment (Bailey and Durrant-Whyte 2006). Data from an IMU and laser scans can also be fused with the visual data to improve accuracy. These techniques are very computationally heavy, require specialized hardware, and are typically used in robotics or autonomous navigation applications.

RF-based techniques, generally referred to as wireless, are the most common in indoor localization and utilize technologies such as Wi-Fi, radio-frequency identification (RFID), Bluetooth, and ultra-wideband (UWB). These technologies are useful for indoor localization because the technologies are already found in existing infrastructure and can be set up at very low cost. Wireless localization techniques can be classified as range-based, or range free. Range-based (or distance-based) techniques are the most accurate and involve measuring the distance from the unknown node being localized to some fixed nodes with a known locations, typically called anchors. Range-free techniques on the other hand use the relative positions of existing nodes, connectivity information, or detecting the proximity of the unknown node to fixed anchors with known locations (using RFID or Bluetooth) to localize. Range-free techniques are simpler, cheaper, and more energy efficient than range-based algorithms but at the cost of having low localization accuracy. Because of this, range-free algorithms are generally only useful when coarse locations are desired.

1.1 Motivation

The main issue that arises in range-based indoor localization is that the various techniques used for measuring distances are greatly affected by obstacles such as walls, furniture, and shelving. As the radio signals travel in an indoor environment, they often have to travel through walls or other obstacles, which are non-line of sight (NLOS) conditions. Radio signals can also bounce off walls and other surfaces before reaching the intended destination, creating reflections that can take multiple paths to the destination and are referred to as multipath conditions. These occurrences can cause errors in the measurements (referred to as measurement noise) which then translate to localization errors, that is the estimated position is different from the true position. There is therefore a need to mitigate the effects of these conditions in order to achieve more accurate indoor localization.

This is not a novel problem and there have been many solutions proposed to address it. Many of the existing solutions to this problem rely on a tight coupling to hardware such as refining

the localization accuracy with sensors like an IMU using a technique called sensor fusion. Another technique employed uses low-level RF data like the impulse or phase response in Ultra-wideband (UWB) localization systems to determine whether a measurement is LOS or NLOS. This requires actually having access to that information on a hardware level in the first place which increases the cost and complexity of the localization system. Most solutions also employ the use of a Kalman filter (KF) for fusing data or correcting noise. While typically accurate, these methods are as a result very application-specific and do not generalize well. Lastly, since these are range-based techniques another solution is to simply add a lot more fixed anchors to the system so that there are never any NLOS conditions – this solution greatly increases system costs and still can't guarantee LOS conditions in a dynamic environment.

1.2 Original Contribution

The aforementioned techniques for range-based localization in indoor environments have their place in specific applications but all have additional cost-adding components or increased complexity that is tied to specific hardware. There is therefore a need for a more general range-based indoor localization solution that has no tight-coupling to hardware and can work with a minimal anchor count while being accurate enough for use in indoor situations. The primary contributions of the proposed work are:

1. An efficient range-based indoor localization algorithm called PSO-Kalman Filter Fusion (PKFF) that is able to produce accurate results in dynamic LOS / NLOS environments with varying measurement noise levels. This is achieved with no tight coupling to or reliance on specific hardware, no need for profiling of the environment, and without using additional sensors.
2. A formulation of the Particle Swarm Optimization (PSO) and Kalman-filtered least-squares (LSQ) trilateration algorithms that provides enough variability in their estimates from the same input source to make them suitable for use in a data fusion scheme.

3. A state vector data fusion technique that is able to achieve a reduction in localization error by using the Inverse Distance Weighting (IDW) interpolation technique to fuse two position estimates dynamically based on the variance of the distance measurements.

CHAPTER 2

INDOOR LOCALIZATION TECHNIQUES

Indoor localization techniques can be broadly classified as range-based and range-free (Kumar and Hegde 2017). As the name implies, range-based techniques rely on range (distance) measurements between the unknown node and an anchor while range free methods rely on connectivity information, fingerprinting, or proximity. Figure 2.1 illustrates this classification as well the associated measurement techniques. As earlier mentioned, range-free techniques are not as accurate as range-based ones and are generally used for course positioning while range-based techniques can be used for very accurate localization, even down to millimeter-level accuracy.

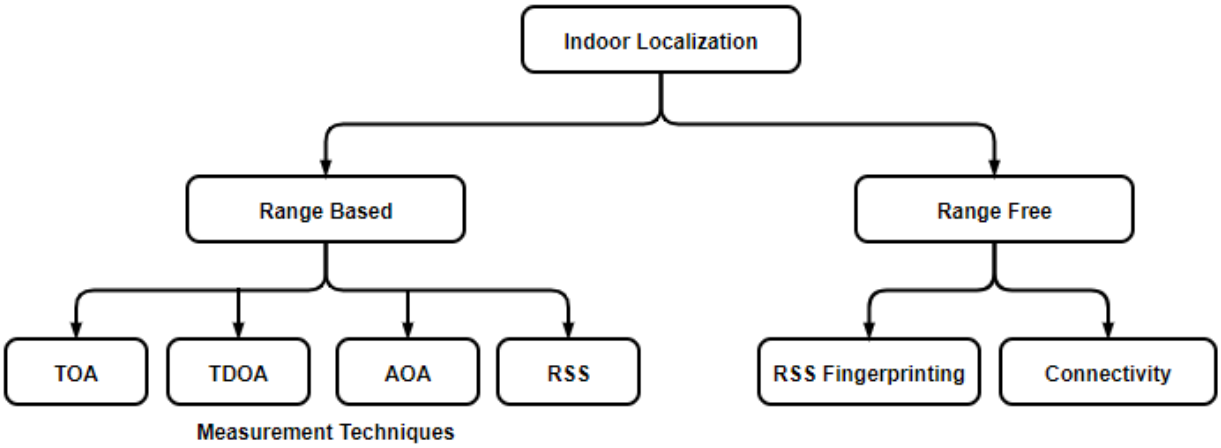


Figure 2.1: Localization Techniques

2.1 Range-Based Localization

Most range-based localization algorithms utilize one of the common measurement techniques which fall into two categories: angle-based and distance-based. The most common angle-based measurement technique is Angle of Arrival (AoA) and the main distance measurement techniques are Time of Arrival (ToA), Time Difference of Arrival (TDoA) and Received Signal Strength (RSS) (Safavi et al. 2018).

2.1.1 Angle-based Measurement

Angle of Arrival (AOA)

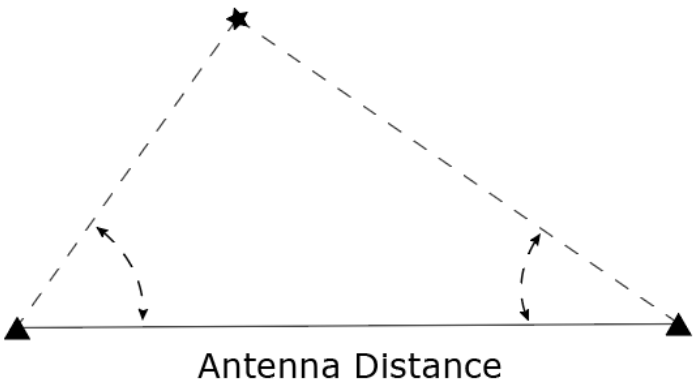


Figure 2.2: Angle of Arrival Localization

AoA measurement techniques calculate the angle (or bearing) between the node to be localized and a fixed anchor with a known location. These measurements are obtained using two main techniques that include (1) measuring the amplitude response of the receiving antenna and (2) measuring the phase response of the receiving antenna. The location of the unknown node is a line having a certain angle from an anchor node – this requires at least two nodes to calculate the position as shown in Figure 2.2 (Paul and Sato 2017). The accuracy of these measurements is affected mainly by the directivity of the antenna as well as the effects of NLOS and multipath in the environment. Since AoA measures angles, it requires direct LOS between the receiver and transmitter because a reflected signal arriving at the receiver can be interpreted as coming from a completely different direction, which can result in very large errors in the measurement (Mao and Fidan 2009).

2.1.2 Distance-based Measurement

Time of Arrival

ToA is a technique that calculates distance based on the measured time of arrival of a signal from a transmitting node to a receiving one. This is more formally referred to as a one way propagation time measurement. The primary drawback to this technique is that it requires perfect time synchronization between the clocks of the transmitter and receiver nodes; any difference between the two clocks can become a large error in the distance calculation. Assuming normal conditions (air as the medium and radio waves traveling at the speed of light) a small clock synchronization error of 1 ns will relate to a distance measurement error of 0.3 m (Paul and Sato 2017).

One way of overcoming this issue is by measuring the round trip propagation time instead of the one way propagation. The first node sends a signal to a second node which in turn immediately sends that signal back to the first node and the distance is calculated using the round trip time. This removes the need for the transmitter and receiver to have synchronized clocks. The primary issue with this round trip method is the processing delay accrued from the second node receiving the signal then sending it back in turn. This delay is usually known and specified by the manufacturer (or during a calibration process) so it can be subtracted from the measurement at the first node.

Time Difference of Arrival

TDoA is another technique that measures propagation time but in this case the difference between the arrival time of a signal at two different fixed receivers is measured. This requires that the location of the two receivers are known and those two receivers also have synchronized clocks. Unlike ToA one way ranging there is no need for the clocks between the transmitter and receivers to be in perfect sync.

Received Signal Strength

There are two main methods of estimating distance using RSS: using the path loss log-normal shadowing model and RSS fingerprinting (Yassin et al., Secondquarter 2017). Distance estimation using the path loss model is accomplished by measuring the signal attenuation as it propagates from the transmitting node to the receiving node. The relationship between distance and signal attenuation is heavily dependent on channel characteristics and as a result requires a very accurate propagation model in order to have acceptable results. The standard log-normal model used in this technique is as follows:

$$P_r(d)[dBm] = P_0(d_0)[dBm] - 10n_p \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \tag{2.1}$$

with $P_0(d_0)[dBm]$ being the reference power at distance d_0 from the transmitter, n_p being the path loss exponent measuring the rate at which the RSS decreases with distance, and X_σ being a zero mean Gaussian random variable with standard deviation σ which accounts for random showing effects (Paul and Sato 2017).

2.1.3 Multilateration

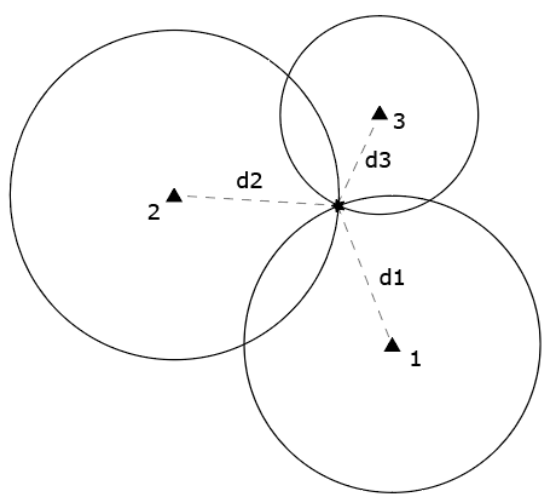


Figure 2.3: Trilateration in Ideal Conditions

Multilateration is a core technique for estimating the location of a node using the measured distances from the node to multiple anchors. Traditionally, this is achieved using 3 anchors (for 2D localization) and is referred to as trilateration (Sadowski and Spachos 2018). The locations of the anchors is assumed to be known and the location of the unknown node is the intersection of the three circles with the center at the location of each anchor and radius equal to the measured distance to the unknown node.

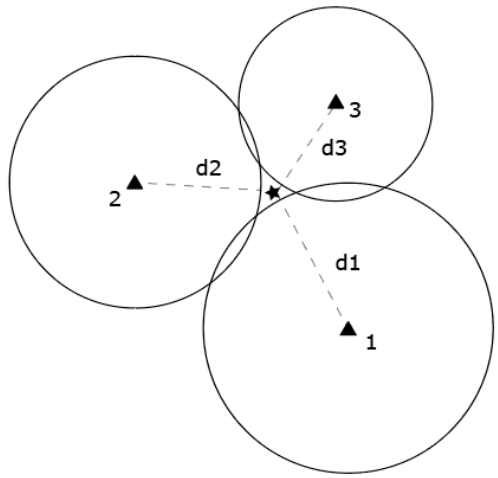


Figure 2.4: Trilateration with Measurement Uncertainty

An example of the relationship between the node and anchors is shown in Figure 2.3. In practice the measured distances are not accurate, as earlier discussed, so the circles don't intersect at a single point as shown in Figure 2.4. In this case with the locations of the anchors and the estimated distances between the anchors and node known, trilateration then becomes a problem of solving three nonlinear circle equations

$$
\begin{aligned}
(x - x_1)^2 + (y - y_1)^2 &= r_1^2 \\
(x - x_2)^2 + (y - y_2)^2 &= r_2^2 \\
(x - x_3)^2 + (y - y_3)^2 &= r_3^2
\end{aligned}
\tag{2.2}
$$

where (x_1, y_1), (x_2, y_2), and (x_3, y_3), are the coordinates of anchors 1, 2, and 3. These equations can be linearized into the form

$$Ax = b \tag{2.3}$$

with

$$\mathbf{A} = \begin{bmatrix} 2(x_1 - x_3) & 2(y_1 - y_3) \\ 2(x_2 - x_3) & 2(y_2 - y_3) \end{bmatrix} \tag{2.4}$$

$$\mathbf{b} = \begin{bmatrix} r_1^2 - r_3^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2 \\ r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \end{bmatrix} \tag{2.5}$$

$$\mathbf{x} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \tag{2.6}$$

which can be solved using the least squares method:

$$x = (A^T A)^{-1}(A^T b) \tag{2.7}$$

Multilateration is not to be confused with triangulation which uses knowledge of the angles between the node and anchors to find the node-to-anchor distances using the law of sines (Safavi et al. 2018).

2.2 Range Free Localization

2.2.1 Connectivity Based

Connectivity based localization works fundamentally by checking if a node is connected to another node. If each node is aware of all the nodes it is connected to then coarse locations can be determined by counting the number of hops between nodes using algorithms such as DV-Hop and Centroid (Paul and Sato 2017). Coarse localization can also be achieved by simply measuring the presence of a node near a fixed anchor with a known position, typically using RFID or Bluetooth.

2.2.2 Profiling / Fingerprinting

Another type of range free localization is the use of RSS profiling or fingerprinting in order to overcome the inaccuracies of RSS-based distance measurements. Fingerprinting involves taking RSS measurements at many points in an environment and building a map of those measurements and their position. Localization then becomes a matter of matching RSS measurements, not distances, to what is stored in the map in order to figure out the location of the node. This type of localization is very well suited for machine learning.

2.3 Particle Swarm Optimization (PSO) Localization

PSO is an optimization algorithm that overcomes the issue of being trapped in local minima and is suitable for use in dynamic environments. It is an evolutionary computation technique belonging to the class of stochastic global algorithms which simulate the social behavior of bird flocking and is computationally efficient (Kennedy and Eberhart 1995; Gopakumar and Jacob 2008). PSO uses a set of particles that are initially randomly populated within a search space. These particles are then moved in the search space following rules that are inspired by bird flocking (swarming). Each particle is moved toward a randomly weighted average of the best position found by the particle (called **pbest**) and also the best position found by the entire particle population (called **gbest**) (Noel, Joshi, and Jannett 2006).

In an N dimensional search space, the position of the ith particle in the swarm is represented by $\mathbf{x}_i = [x_{i1}, x_{i2}, ...x_{iN}]$ and the velocity of the ith particle is represented by $[v_{i1}, v_{i2}, ...v_{iN}]$. $gbest = [g_{i1}, g_{i2}, ...g_{iN}]$ represents the best particle in the swarm while $pbest = [p_{i1}, p_{i2}, ...p_{iN}]$ holds the position of the personal best for that particle (Gopakumar and Jacob 2008). An objective function is used to determine the fitness of a particle as it moves through the search space. The fitness of a particle closer to the global solution will be higher than that of a particle that is farther away and PSO tries to minimize or maximize that objective function (Ramesh et al. 2012). Particles update their location and velocity based on the equations

$$v_{id} = \omega v_{id} + c_i r_1 (p_i - x_i) + c_2 r_2 (g_d - x_{id}) \tag{2.8}$$

$$x_{id} = x_{id} + v_{id} \tag{2.9}$$

Where ω is the inertial weight, c_1 and c_2 are the cognitive and social constants, and r_1 and r_2 are uniformly distributed numbers between 0 and 1. $d = 1, 2, ..., N$ and $i = 1, 2, ..., K$ with K being the swarm size. For each particle, c_1 determines how much its best solution influences the particle and c_2 is how the rest of the swarm influences the particle. The inertia weight controls the exploration of the search space (local and global) (Gopakumar and Jacob 2008). The PSO algorithm is illustrated in Figure 2.5.

2.3.1 PSO Localization

In order for PSO to be used for localization, the objective function can be framed as the mean squared error (MSE) between the particle and each anchor node, achieved by subtracting the calculated distance between the node and anchor from the noisy measured distance $\hat{d}$.

$$f(x, y) = \frac{1}{M} \sum_{i=1}^{M} \left(\sqrt{(x - x_i)^2 + (y - y_i)^2} - \hat{d}_i \right)^2 \tag{2.10}$$

with $M >= 3$ being the number of anchors, (x_i, y_i) the coordinates of the ith anchor, and (x, y) coordinates of the node. PSO tries to minimize this objective function until either a set threshold is reached or the set number of iterations are reached.

2.4 Data Fusion in Localization

Data fusion can be used to improve localization accuracy by combining any number of localization algorithms and techniques. Data fusion is the integration and extraction of information from data obtained by two or more sensors or data sources. This is often done as a way of obtain-

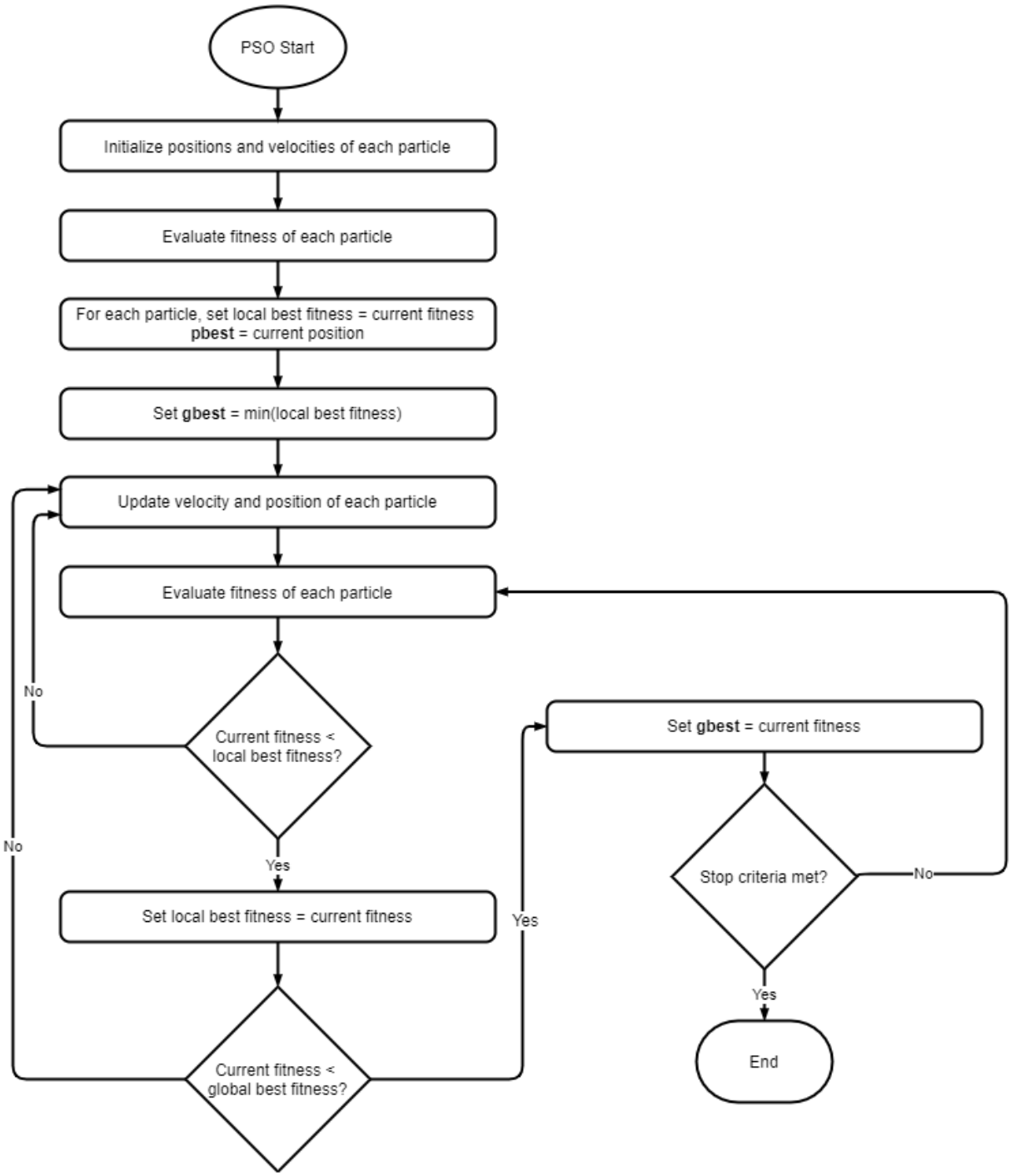


Figure 2.5: Particle Swarm Optimization Algorithm

ing more accurate and complete information about a particular operation or to improve decision making (Mosallaei, Salahshoor, and Bayat 2007). Data fusion can be found in a multitude of ap-

plications including multi-sensor data fusion (MSDF), the combination of multiple databases, and robot navigation to name a few. The term ”Data fusion” itself was coined by the Joint Working Group Directors of Laboratories (JDL) of the US Defense Department and was formally defined as ”a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identify estimates, and complete timely assessments of situations and threats, and their significance. The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results” (Ben Ayed, Trichili, and Alimi 2015).

Data fusion can, broadly speaking, improve the performance of a system through representation, certainty, accuracy, and completeness (Mitchell 2012). An improvement through representation means the the information that comes out of data fusion has a higher granularity or abstract level than the input data. A certainty improvement means there is a gain in certainty such that the *a priori* probability of the fused output data is greater than that of the input data. An accuracy improvement is simply a reduction in standard deviation between input data and the fused output data. Lastly, completeness allows for a more complete view through the bringing in of new information to the current knowledge (Mitchell 2012).

2.4.1 Fusion Strategies

Boudjemaa and Forbes (2004) have generalized various sensor fusion strategies into four types based on which part of the system is used in the fusion:

1. **Fusion across sensors**: Various sensors nominally measure the same quantity or property.
2. **Fusion across attributes**: Various sensors sensors measure different quantities or properties associated with the same situation.

3. **Fusion across domains**: Various sensors measure the same attribute over a number of different ranges or domains.
4. **Fusion across time**: Current measurements are fused with historical information.

Similarly, Durrant-Whyte (1988) categorized sensor data fusion based on sensor configuration into three categories:

1. **Complementary fusion**: The sensors are not directly dependent on each other but can be combined in order to give a more complete image. This type of fusion helps reduce incompleteness.
2. **Competitive fusion**: The sensors each deliver an independent measurement of the same property. This type of fusion helps to reduce the effects of uncertain and erroneous measurements.
3. **Cooperative fusion**: Independent sensors are used to derive information that would not be available from each individual sensor (Mitchell 2012).

2.4.2 Kalman Filter-based Data Fusion

Of the various techniques utilized for data fusion, Kalman filtering is one of the most popular since it is an efficient recursive algorithm that is suitable for real-time applications (Mosallaei, Salahshoor, and Bayat 2007). The KF utilizes a mathematical model for filtering signals - it is an iterative process that uses prior knowledge of system and measurement noise characteristics to account for and filter out the noise (Shareef and Zhu 2009). Intuitively, the KF works by comparing its prior estimate to new measurements and then predicting a new estimate that is in between those two values based on whether it ”trusts” the measurements or its model more. The “trust level” is referred to as the Kalman gain. A thorough derivation and analysis of Kalman filters is presented by Brookner (1998) and Labbe (2018).

The primary benefits of using a KF for tracking as outlined by Brookner (1998) are that it:

- Provides a running measure of accuracy of predicted position.

- Permits optimum handling of measurements of accuracy that varies with time.
- Allows optimum use of a priori information if available.
- Permits target dynamics to be used directly to optimize filter parameters.
- Has the addition of random-velocity variable, which forces Kalman filter to be always stable (Brookner 1998).

Kalman Filter Equations

The performance of a KF is heavily contingent on selecting the proper model for the dynamics of the system in use. In the case of indoor localization, which is typically a low dynamic scenario, the position-velocity (PV) model, also called the Continuous Wiener Process Velocity (CWPV) model, is most useful (Alberto and Murillo 2011). In 2D space, this model tracks the (x, y) position as well as the velocities and represents the standard kinematics equation $x = x_0 + v_0 t$. This model assumes that the target being tracked moves at near constant velocity between measurement updates. The linear KF can be described using two linear equations:

$$\mathbf{x_k} = \mathbf{F_k x_{k-1}} + \mathbf{G_k u_{k-1}} + \mathbf{w_{k-1}} \tag{2.11}$$

$$\mathbf{z_k} = \mathbf{H_k x_k} + \mathbf{v_k} \tag{2.12}$$

In Equation(2.11) $\mathbf{x_k}$ is the state vector given by

$$\mathbf{x_k} = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T \tag{2.13}$$

in which x_k and y_k are the positional coordinates and $\dot{x}_k$ and $\dot{y}_k$ are the velocities in the x and y directions. **F** is the transition matrix given by

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t_k & 0 \\ 0 & 1 & 0 & \Delta t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

with $\Delta t = t_k - t_{k-1}$ being the sampling period. Matrix **G** is the control input given by

$$\mathbf{G} = \begin{bmatrix} \frac{\Delta t_k^2}{2} & 0 \\ 0 & \frac{\Delta t_k^2}{2} \\ \Delta t_k & 0 \\ 0 & \Delta t_k \end{bmatrix} \tag{2.15}$$

and defines the relationship between the input vector u_k which represents the independent random acceleration and is given by

$$\mathbf{u_k} = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{2.16}$$

and the the system state vector. W_k is the process noise with diagonal covariance matrix $\mathbf{Q} = \sigma_1^2\mathbf{I}$, with **I** being an identity matrix. The second linear equation (2.12) describes the noisy observations of the system where each element of the output vector $\mathbf{y_k}$ holds a sensor measurement, **H** is the observation matrix given by

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.17}$$

and $\mathbf{v_k}$ is the measurement noise modeled as Gaussian white noise with covariance matrix **R**.

$$\mathbf{R} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{2.18}$$

$\mathbf{z_k}$ is the observation vector of measurements which is defined as $\mathbf{z_k} = [x_k, y_k]^T$.

Kalman Filter Steps

The Kalman filter algorithm has two stages, state prediction and state update. In the **prediction** stage, the state vector is predicted using the following equation:

$$\hat{\mathbf{x}}_\mathbf{k} = \mathbf{F}\mathbf{x}_{\mathbf{k-1}} + \mathbf{G}\mathbf{u}_{\mathbf{k-1}} \tag{2.19}$$

Next, the state error covariance matrix **P** is predicted using:

$$\mathbf{P_k} = \mathbf{F}\mathbf{P}_{\mathbf{k-1}}\mathbf{F}^\mathbf{t} + \mathbf{Q} \tag{2.20}$$

For the state **update** stage the Kalman gain matrix, **K**, is calculated by:

$$\mathbf{K_k} = \mathbf{P_k}\mathbf{H}_\mathbf{k}^\mathbf{T}(\mathbf{H_k}\mathbf{P}_\mathbf{k}^\mathbf{T} + \mathbf{R})^{-\mathbf{1}}) \tag{2.21}$$

The state estimate can then be updated by:

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}_{\mathbf{k-1}} + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H_k}\hat{\mathbf{x}}_{\mathbf{k-1}} \tag{2.22}$$

Followed by the state error covariance being updated by:

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k}\mathbf{H_k})\mathbf{P_k} \tag{2.23}$$

with **I** being an identity matrix.

Methods of Kalman Filter-based Fusion

The two most common methods for data fusion using Kalman filters are state-vector (track-to-track) fusion and measurement fusion. As shown in Figure 2.6, state vector fuses the outputs (states) of individual KFs in order to get an improved joint state (Bhattacharya and Raj 2004). This type of fusion has lower computational cost than measurement fusion and provides more fault tolerance.

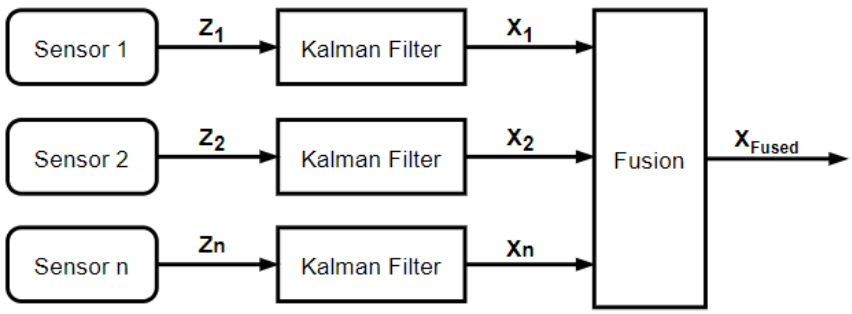


Figure 2.6: State Vector Fusion

Measurement fusion (Figure 2.7) uses a single KF to obtain a final state from individual measurements that have been already fused by either simply merging them or using minimum-mean-square-error estimation (Habtie, Abraham, and Midekso 2015). This type of fusion can be more accurate than state-vector fusion but requires that the sensors have identical measurement matrices (Bhattacharya and Raj 2004).

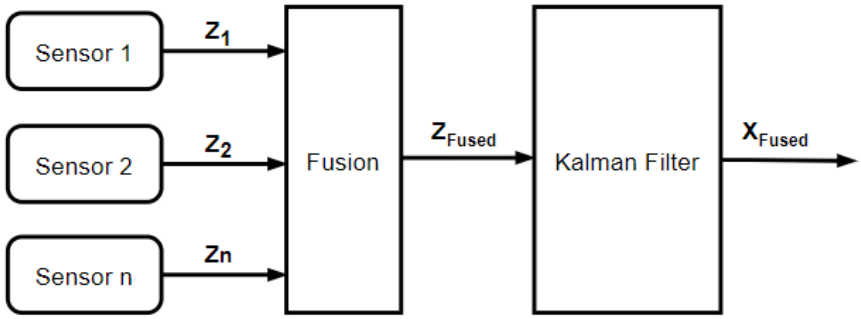


Figure 2.7: Measurement Fusion

CHAPTER 3

LITERATURE REVIEW

There are quite a number of approaches to the indoor localization problem available in literature. Since the biggest factor in improving localization accuracy is the mitigation of the effects of NLOS conditions, many approaches specifically target the identification of NLOS conditions. An even greater number of approaches seek to improve localization accuracy by fusing multiple sources of data to try and gain a more accurate location, typically involving sensors. Lastly, a number of machine learning techniques have also been employed, mostly relying on RSS fingerprinting data obtained from WiFi, RFID, or Bluetooth to train models.

3.1 NLOS Identification

There has been significant research interest in the identification and mitigation of NLOS conditions in indoor localization and tracking. Reducing localization and tracking errors due to NLOS conditions is typically achieved via two main strategies: hard and soft. The hard strategy involves first attempting to identify the NLOS condition and if successful then discarding or mitigating those measurements in some way, such as in Bregar and Mohorčič (2018). On the other hand, the soft strategy doesn't seek to explicitly classify measurements as LOS or NLOS and instead either use adaptive methods, as demonstrated by Zhang et al. (2013), to adjust the probabilities of each model or use sub-problem transformation to mitigate NLOS errors (Y. Wang et al. 2018).

Machine learning techniques are quite popular for NLOS detection since by nature it is a classification problem and is well suited for machine learning. Bregar and Mohorčič (2018) used raw channel impulse response (CIR) data from UWB ranging modules to classify measurements as LOS or NLOS using a convolutional neural networks (CNN). This predicted channel state and associated ranging error information was then used with least squares and weighted least squares (WLS) algorithms to perform localization. The CNN NLOS channel classification method slightly

outperformed the statistically derived CIR method while the use of CNN with WLS methods significantly improved the localization performance (Bregar and Mohorčič 2018).

A TDOA-based classifier was proposed by Wu et al. (2018). This classifier first identified correlations between measured distances and TDOA data. These correlations were then used to train a support vector machine (SVM) classifier. The NLOS measurements were simply discarded. Results showed a 70% improvement in accuracy when NLOS measurements were discarded (Wu et al. 2018).

Finally, Zhu et al. (2019) proposed an NLOS identification method using an AdaBoost classifier. Signal features from TOA data were first extracted and used to train weak learners and then a strong classifier was constructed by boosting the weak learners (Zhu et al. 2019). Their classifier had the lowest localization error compared to other classifiers like an SVM.

Switching from machine learning techniques, Xu, Wang, and Zekavat (2011) proposes a phase wrapping selection algorithm for calculating the phase difference variance across two-antenna elements. A relationship is maintained between the variance and Rician K-factor and then a hypothesis test is formed on the K-factor in order to detect NLOS conditions. Another similar technique presented by Decarli et al. (2010) uses features from the channel waveforms to estimate LOS and NLOS conditions via hypothesis test.

In Yu et al. (2017), a mean shift Kalman filter method is proposed. First a number of distance measurements are acquired for each anchor and the mean value determined prior to being used in the Kalman prediction stage. A hypothesis and an “alternative method” are then utilized to detect the NLOS condition. In the NLOS condition the mean shift method, which is used to approximate the probability density, is used to calculate the weighted means of the range measurements prior to being used in a data association algorithm before the Kalman update stage. The proposed solution had a 13.07% higher localization accuracy than the Maximum Likelihood (same as least squares) method (Yu et al. 2017).

Yi et al. (2014) presents a method based on individual measurement estimation and LOS detection. This method first calculates a pseudo-measured position by choosing the point along the

circle defined by a measurement which has the shortest distance to the predicted position of the moving target. This is then passed to a detector to be identified as LOS or NLOS. All the LOS pseudo-measured positions are then used as a new measurement for a KF to create a state update of the target.

Finally, an interacting multiple model (IMM) filtering algorithm is proposed by Zhang et al. (2013) meant to deal with frequent transitions between NLOS and LOS conditions. This algorithm first adaptively calculates the probabilities of NLOS and LOS models, which then interact automatically through the Markov chain. Finally, two parallel KFs are used (one for each model) and then combined based on the model probabilities. The cumulative distribution function (CDF) Results showed a 90-percentile localization error of 1.29 m (Zhang et al. 2013).

Table 3.1: NLOS Detection

Reference	Title	Localization Technique	NLOS Detection Technique
Zhu et al. (2019)	NLOS Identification via AdaBoost for Wireless Network Localization	Range-Based	Machine Learning (AdaBoost)
Wu et al. (2018)	TDOA Based Indoor Positioning with NLOS Identification by Machine Learning	Range-Based	Machine Learning (SVM)
Bregar and Mohorčič (2018)	Improving Indoor Localization Using Convolutional Neural Networks on Computationally Restricted Devices	Range-Based	Machine Learning (CNN)
Yu et al. (2017)	Mean Shift-Based Mobile Localization Method in Mixed LOS/NLOS Environments for Wireless Sensor Network	Range-Based	Hypothesis method with threshold
Yi et al. (2014)	Target Tracking in Mixed LOS/NLOS Environments Based on Individual Measurement Estimation and LOS Detection	Range-Based	Confiedence measure with KF
Xu, Wang, and Zekavat (2011)	Non-line-of-sight Identification via Phase Difference Statistics Across Two-Antenna Elements	Range-Based	Hypothesis method
Decarli et al. (2010)	LOS/NLOS Detecttion for UWB Signals: A Comparative Study Using Experimental Data	Range-Based	Hypothesis method

3.2 Data Fusion

Data fusion has also been heavily utilized as a technique to improve localization and tracking accuracy. The majority of the data fusion in localization research pairs RSSI information with data from another physical device like an IMU.

S. Wang et al. (2018) and Röbesaat et al. (2017) both proposed solutions that fuse RSSI and IMU data for localization. S. Wang et al. (2018) fused position estimates acquired from RSSI/-geomagnetic fingerprinting with data from a Strapdown Intertial Navigation System (SINS). An adaptive Unscented Kalman Filter (UKF) was then used to fuse all three data sources. Results showed an 88-percentile localization error of 1 m (S. Wang et al. 2018).

Another similar method was proposed by F. Wang et al. (2019) that relied on RFID tracking using an UKF and fused that data with trajectory information from the IMU only when the RFID sampling period was larger than a preset threshold (F. Wang et al. 2019). This method is effectively the same as the one proposed by S. Wang et al. (2018) including the use of a UKF for fusion. Kaya and Alkar (2018) fuses RSSI and accelerometer data sources.

Similarly, Röbesaat et al. (2017) proposed a mobile localization solution that used Kalman filter-based fusion to combine position information from RSSI range-based trilateration and dead reckoning. BLE beacons are placed at known locations and the RSSI readings to each beacon are used to obtain a course location. The dead reckoning is performed by an IMU tracking the number of steps as well as the direction using an android phone. A KF is then used to merge the two positions. This technique was shown to provide an average accuracy of under 1 m (Röbesaat et al. 2017).

Another example of fusing range-based localization with sensor data from an IMU is presented by Briese, Kunze, and Rose (2017). In this work a position estimate was also generated initially by trilateration from UWB modules in fixed locations. A dynamically adjusted covariance KF was then used to fuse the estimated position with absolute acceleration measurements from an IMU. The KF was adjusted by dynamically tuning the state covariance matrix using the accelera-

tion data from the IMU, which allowed the system to have comparable performance in both static and dynamic environments (Briese, Kunze, and Rose 2017).

A range-based fusion method that did not use an IMU or fingerprinting was proposed by Y. Wang et al. (2018). First the distance measurements were classified as LOS or NLOS and distance correction applied. Each measurement was then filtered using a Square Root Unscented Kalman Filter (SRUKF) and a particle filter (PF). After that the values were fused based on the LOS/NLOS probabilities. Finally convex optimization is performed and maximum likelihood estimation is used to estimate the final position. An average localization error of 1.96 m was achieved (Y. Wang et al. 2018).

Table 3.2: Fusion Based Techniques

Reference	Title	Localization Techniques	Fused Attributes	Fusion Method
Proposed Work	Accurate Range-Based Indoor Localization Using PSO-Kalman Filter Fusion	Range-Based	Position Estimate from PSO, Position Estimate from KF	KF
F. Wang et al. (2019)	Indoor Tracking by RFID Fusion with IMU Data	Fingerprinting, Sensor Dead Reckoning	RFID RSSI, IMU Dead Reckoning Data	Not specified
Miyagusuku, Yamashita, and Asama (2019)	Data Information Fusion From Multiple Access Points for WiFi-Based Self-localization	Fingerprinting	Weighted Individual Likelihoods for Position	Joint Likelihood using general Product of Experts (gPoE)
Gu et al. (2019)	Indoor Localization Fusion Algorithm Based on Signal Filtering optimization Of Multi-sensor	Fingerprinting	Position Estimates from Bluetooth, WiFi, and RFID Fingerprinting	Modified KF (KILA), Averaging
Y. Wang et al. (2018)	A Hierarchical Voting Based Mixed Filter Localization Method for Wireless Sensor Network in Mixed LOS/NLOS Environments	Range-Based	Position Estimate from KF, Position Sstimate from Particle Filter	Probabilistic
S. Wang et al. (2018)	A Data Fusion Method of Indoor Location Based on Adaptive UKF	Fingerprinting, Sensor Dead Reckoning	MEMS-SINS, RSSI, and Geomagnetic measurements	Adaptive Unscented KF (UKF)
Landolsi and Shubair (2018)	TOAI/AOA/RSS Maximum Likelihood Data Fusion for Efficient Localization in Wireless Networks	Range-Based, Fingerprinting	TOA, AOA, and RSS Measurements	Maximum likelihood (ML) Estimation
Kaya and Alkar (2018)	Indoor localization and tracking by multi sensor fusion in KF	Fingerprinting, Sensor Dead Reckoning	RSSI, Accelerometer data	KF
Röbesaat et al. (2017)	An Improved BLE Indoor Localization with Kalman-Based Fusion: An Experimental Study	Range based, Sensor Dead Reckoning	Trilateration position, dead reckoning position	KF
Briese, Kunze, and Rose (2017)	UWB localization using adaptive covariance KF based on sensor fusion	Range-Based, Sensor Dead Reckoning	Position Estimate, acceleration Data	KF
Belmonte-Hernández et al. (2017)	Adaptive Fingerprinting in Multi-Sensor Fusion for Accurate Indoor Tracking	Fingerprinting	Position Estimates	KF (Multisensor Fusion using Adaptive Fingerprint)

3.3 Machine Learning Techniques

The majority of machine learning-based indoor localization solutions use RSSI measurements from existing WiFi infrastructure. As previously mentioned, RSSI-based localization typically involves extensive fingerprinting of the environment, which is well suited for machine learning applications.

Many machine learning solutions use ensemble learning techniques. Ahmadi et al. (2017) and Nastac et al. (2017) both use classifier and regression methods for localization with WiFi RSS data. Ahmadi et al. (2017) proposes a method based on dividing the training input set into subset then applying an RT localization algorithm to each. K-Nearest Neighbor (KNN) classifier is used to select the best anchors. The proposed method had about a 7% higher accuracy than a Support Vector Machine (SVM) based method (Ahmadi et al. 2017).

Zhao and Wang (2017) pairs Kernel Direct Discrimination Analysis (KDDA) with Relevance Vector Regression (RVR) while Salamah et al. (2016) pairs a K-Nearest Neighbor (KNN) classifier with a Support Vector Machine (SVM). In Zhao and Wang (2017), the KDDA method is first used to extract non-linear localization features then the RSS localization information is reorganized to remove redundant positioning features prior to being used in RVR (Zhao and Wang 2017). This solution had an 86.2-percentile localization error of 1.5 m.

Fan et al. (2019) and Zou et al. (2016) take a different approach and use Neural Networks (NN). The approach proposed by Fan et al. (2019) doesn't use RSS data but instead uses channel state information (CSI) fingerprinting, which is much more accurate than RSS measurements. A Deep Neural Network (DNN) and Back Propagation Neural Network (BPNN) are used and achieved over 75% mean square error below 1 m (Fan et al. 2019). Zou et al. (2016) uses a Weighted Extreme Learning Machine (WELM) to perform localization. This technique is based on a generalized Single-hidden Layer Feedforward Neural Network (SLFN) and is a new way of handling device heterogeneity and environmental dynamics by standardizing RSS values (Zou et al. 2016). Experimental results showed a 6.35% improvement over the RSS-ELM method.

Table 3.3: Machine Learning Techniques

Reference	Title	Localization Technique	Attributes	Machine Learning Techniques
Fan et al. (2019)	A Machine Learning Approach for Hierarchical Localization Based on Multipath MIMO Fingerprints	Fingerprinting (MIMO Radio)	CSI	DNN, BPNN
Ahmadi et al. (2017)	Learning Ensemble Strategy for Static and Dynamic Localization in Wireless Sensor Networks	Fingerprinting (WiFi)	RSS	KNN, RT
Nastac et al. (2017)	Indoor Positioning WLAN based Fingerprinting as Supervised Machine Learning Problem	Fingerprinting (WiFi)	RSS	Regression, Classification
Zhao and Wang (2017)	WiFi indoor positioning algorithm based on machine learning	Fingerprinting (WiFi)	RSS	KDDA, RVR
Zou et al. (2016)	A Robust Indoor Positioning System Based on the Procrustes Analysis and Weighted Extreme Learning Machine	Fingerprinting (WiFi)	RSS	WELM
Salamah et al. (2016)	An enhanced WiFi indoor localization system based on machine learning	Fingerprinting (WiFi)	RSS	KNN, SVM

3.4 Research Comparison

The various techniques presented in literature for indoor localization all have their place in different applications. The machine learning techniques are heavily limited to coarse position estimates since they rely on fingerprinting of the environment and training. These methods also have no recourse for dynamic NLOS conditions that may alter the fingerprinted environment. NLOS identification is also an effective technique but generally requires tight coupling to hardware in order to be useful. Most of the techniques presented in literature that are effective require low level channel information such as CIR or CIS such as the solution presented by Bregar and Mohorčič (2018). Data fusion is the most effective technique since it can be used to combine any number of the aforementioned techniques to gain more accurate results. The most effective data fusion solutions used additional hardware such as an IMU to achieve low localization error.

The algorithm proposed in this research aims to be a more general software-only solution that is not tied to any specific hardware. The proposed algorithm is range-based and does not require fingerprinting of the environment, meaning it can be deployed in a cost and time effective manner. It does not seek to actively identify LOS/NLOS conditions but uses data fusion *with a single input* to effectively deal with NLOS conditions. Unlike the majority of existing solutions proposed in literature, this algorithm manages to use data fusion to improve localization accuracy using a single input by leveraging the unique characteristics of KF and PSO localization techniques. The closest solution proposed in literature is that by Y. Wang et al. (2018). One of the key differences is the solution only filters and fuses individual measurements and doesn't act on the entire system state. As results will show, fusing position estimates provides greater adaptability and accuracy in tracking applications.

CHAPTER 4

PROPOSED ALGORITHM

In this section the PSO - Kalman Filter Fusion (PKFF) algorithm is presented. PKFF is a range-based localization algorithm that uses state vector fusion to dynamically combine position estimates from a windowed PSO algorithm and Kalman-filtered least squares trilateration (LSQ) algorithm in order to accurately localize in mixed LOS/NLOS environments. PKFF consists of three primary steps:

1. **Weighted Variance Calculation:** The first stage calculates a weighted average of all variance measurements. No explicit NLOS detection is performed.
2. **Position Estimation:** Two position estimates are calculated using windowed PSO and adjusted KF.
3. **Fusion:** The two position estimates are dynamically fused using state-vector fusion based on the weighted variance calculated in Step 1.

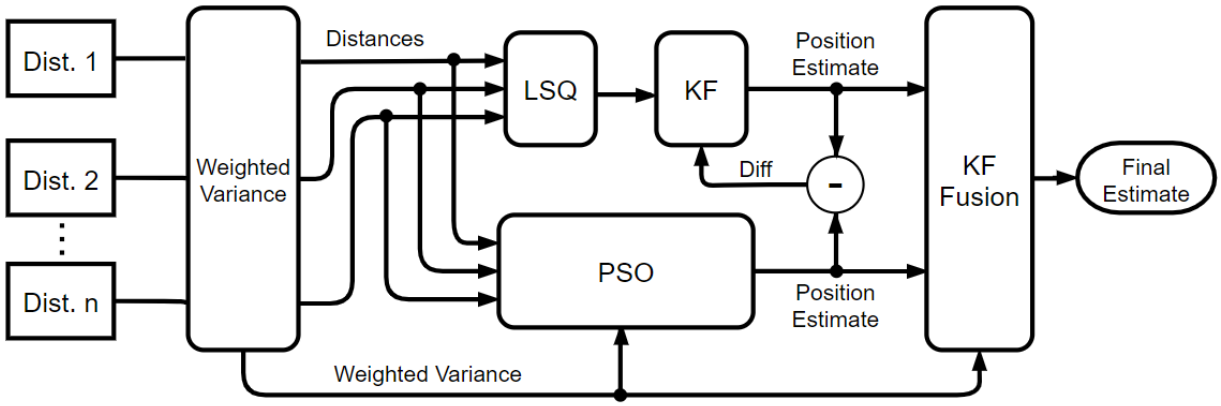


Figure 4.1: PKFF Algorithm

Algorithm 1 PKFF

```
A, B, C ← N distance measurements to each anchor
D ← [MEAN(A), MEAN(B), MEAN(C)]
S ← [VAR(A), VAR(B), VAR(C)]
function PKFF(D, S)
    W_v ← VARIANCEWEIGHT(D, S)
    P_l ← LSQ(D)
    P_k ← KALMANFILTER(P_l)
    P_p ← PARTICLESWARM(D, S, W_v)
    dv ← DISTANCE(pos_p, pos_kf)
    if dv > .5 then UPDATEKF(P_p)
    P_f ← KALMANFILTERFUSE(P_k, P_p, W_v)
    return CONSTRAIN(P_f)
```

4.1 Weighted Variance Calculation

The running variance estimator can be a useful too for estimating noise in distance measurements. At any time t, N distance measurements from a node to an anchor i can be represented by

$$D_i = [\hat{d}_1(t), \hat{d}_2(t), ..., \hat{d}_N(t)] \tag{4.1}$$

Each distance measurement at time t is modeled by

$$\hat{d}_t = d_t + n_i \tag{4.2}$$

where n_i is both the LOS and NLOS measurement noise to anchor i represented by

$$n_i = \begin{cases} n_i \sim N(0, \sigma_i^2) & \text{LOS} \\ n_i \sim N(\mu_{NLOS}, \sigma_i^2 + \sigma_{NLOS}^2) & \text{NLOS} \end{cases} \tag{4.3}$$

The running variance on N consecutive range estimates D can in theory be used to detect LOS and NLOS conditions and is modeled by:

$$\theta_{rv} = \frac{\sum_{n=1}^{N} \left(\hat{d}_n - \mu_D\right)^2}{N-1} \begin{cases} < \theta_{rv} & \rightarrow \text{LOS} \\ > \theta_{rv} & \rightarrow \text{NLOS} \end{cases} \tag{4.4}$$

where the threshold $\theta_{rv} = \sigma^2_{LOS}$ and $\mu_d = \frac{1}{N}\sum_{n=1}^{N} \hat{d}_n$ (Schroeder et al. 2007).

Even though the variance of the LOS measurements is typically known in a given indoor environment, this technique is only useful for static localization because using it for localizing a mobile node exposes an inherent flaw where if the node moves during the estimation period (which is assumed to be true for a mobile node) then the variance will be overestimated, leading to false detection of NLOS conditions even in LOS conditions (Schroeder et al. 2007). The obvious attempt to mitigate this problem is to increase the detection threshold by factoring in the maximum velocity of the object with the update interval. This solution, though, is only feasible if both the update interval and maximum velocity are not too large, which can be a subjective measure. If both of these parameters are too large then the NLOS detection threshold will be increased to a point where no NLOS conditions are detected (Schroeder et al. 2007).

4.1.1 Weighted Variance

PKFF avoids making a hard LOS / NLOS distinction and instead calculates a weighted variance that takes into account the variance of each measurement as well as the distance to the anchor. This is based on the a deterministic spatial interpolation model called inverse distance weighting (IDW). The idea behind IDW is that there is a relationship between two points (in this case the known anchor and unknown node positions) but the similarity between them is inversely related to the distance between them. A power (exponential function) can also be used to modify the distance weight (Lu and Wong 2008). PKFF uses this technique to calculate a weighted average of the measurement variances, which is then used to dynamically adjust the fusion between the PSO and KF. The weighted variance σ^2_w is given by:

$$\sigma_w^2 = \frac{\sum_{i=1}^{n} \left(\frac{\sigma_i^2}{d_i^p} \right)}{\sum_{i=1}^{n} \left(\frac{1}{d_i} \right)} \tag{4.5}$$

with $0 < \sigma_w^2 < 10$ where d_i is the distance measurement to the i^{ith} anchor, σ_i^2 is the variance of the ith distance measurement and $p = 3$ is the power.

4.2 Position Estimation

4.2.1 Adjusted Kalman Filter

The KF used in PKFF tracks a position estimate from the least squares trilateration (LSQ) algorithm described in Chaper 2. The intuition behind how the linear KF is used in PKFF stems from its behavior as a very good predictor of the system state even with uncertainties in the model. The KF can be configured to have a very ”slow” response which makes it immune to large estimate jumps at the cost of a loss in accuracy. This configuration makes the KF very precise but inaccurate in its tracking, meaning it will lose track and diverge from the true position over time. The problem then becomes a matter of finding a way to increase the accuracy of the tracking without sacrificing precision.

PKFF overcomes this issue by detecting the divergence between the PSO estimate and the KF estimate and updating the internal state of the Kalman filter with the PSO position when the difference between the two estimates is greater than 50 cm. This creates a KF that is stable enough to maintain track of a randomly moving object while still resisting the perturbations created by NLOS conditions. This is as opposed to using the distance measurements directly in an extended Kalman filter (EKF), which results in extremely poor performance and track loss due to the large differences between LOS and NLOS distance measurements.

Kalman Filter Implementation

The KF utilized uses the same random walk model as described in Chapter 2 with $\sigma^2 = .05$ and a measurement noise matrix with diagonals $[5, 5]^T$. Since this is a tracking application with no control input, the state prediction equation of the KF is simply given by:

$$\mathbf{x_k} = \mathbf{F_k}\mathbf{x_{k-1}} \tag{4.6}$$

Interestingly, the individual measurement and process noise covariance values aren't important, what matters is the ratio between the two. As long as the ratio is maintained the values can be scaled to any value and as long as ratio is maintained the system behavior will be the same. Singh and Mehra (2015) does a great job of experimentally demonstrating the effects of modifying these value in a tracking application.

4.2.2 Windowed Particle Swarm Optimization

PKFF makes a number of changes to the basic PSO algorithm in order to make it better suited for use in a mobile localization scheme. First, the weighted variance is used to calculate the PSO radius, r_{pso}, which is an inverse linear interpolation of the weighted variance σ_w^2 adjusted from the interval $[0, 10]$ proportionally to the interval $[2, 0.1]$. Particles are initialized in a circle centered around the last estimated position with a radius $r = \sigma_w^2$ and during the first run the center is selected to be the KF estimate. This creates a moving window of particles which reduces the search space and time.

Secondly, a weighted variance of each individual measurement, σ_p^2, is calculated using Equation (4.5) with $n = 1$. A distance d_w is then calculated as a linear interpolation of σ_p^2 from $[0, 10]$ to $[0.1, 0.6]$. The values 0.1 and 0.6 are the mean of the LOS and NLOS errors, respectively, as measured on a calibrated UWB system so they serve as conservative minimums. The objective function minimized by the PSO algorithm is then modeled as:

$$f(x,y) = \sum_{i=1}^{M} \frac{1}{\hat{d}_i}\Big(\sqrt{(x-x_i)^2+(y-y_i)^2} - (\hat{d}_i - d_w)\Big)^2 \tag{4.7}$$

with (x_i, y_i) the coordinates of the ith anchor, and (x, y) coordinates of the node. $\hat{d}_i$ is the measured distance to the ith anchor.

4.3 Fusion

The core PKFF fusion logic is based on a method used by Bhattacharya and Raj (2004) in which range measurements from two identical S band radar were combined using KF fusion in order to track a flight vehicle at a test range. This is important to this case because the two localization algorithms will also be ”tracking” the same target using the same input data source. The two state estimates are fused in the update step of the fusing KF according to:

$$\hat{x}_F = \hat{x}_{kf} + \hat{P}_{kf}(\hat{P}_{kf} + \hat{P}_{pso})^{-1}[\hat{x}_{pso} - \hat{x}_{kf}] \tag{4.8}$$

where x_{kf} and x_{kf} are the position state vectors while P_{kf} and P_{pso} are the measurement covariance matrices with

$$\sigma^2 = \begin{cases} w & \text{KF state} \\ 1-w & \text{PSO State} \end{cases}$$

w is a linear interpolation σ_w^2 from $[0, 10]$ to $[0, .5]$. This results in PKFF favoring the PSO state in low noise conditions and then incorporates the KF proportionally as the estimated variance increases. The output is then constrained to the bounds of the localization area and within a circle with center at the last position estimate and radius based on the estimated maximum velocity of the nodes being localized. This effectively finds the closest point on the circumference of a circle to another point and is represented by the equations:

$$\hat{x}_p = x_c + r_{pso}\frac{x_p - x_c}{\sqrt{(x_p-x_c)^2+(y_p-y_c)^2}}$$

$$\hat{y}_p = y_c + r_{pso}\frac{y_p - y_c}{\sqrt{(x_p - x_c)^2 + (y_p - y_c)^2}}$$

where (x_c, y_c) is the center point of the circle, (x_p, y_p) is the point outside the circle, and $(\hat{x}_p, \hat{y}_p)$ is the new point on the circle.

4.4 Computational Complexity

The computational complexity of PKFF is influenced by the complexity of the LSQ, KF, and PSO algorithms. Both LSQ and KF use matrix algebra so their computational complexity is based on the complexity of the standard matrix operations used. Matrix multiplication and inversion both have a complexity of $\mathcal{O}(n^3)$ and since the KF equations have many of these operations the dominating complexity of the KF is $\mathcal{O}(n^3)$ with $n = 4$ the number of state parameters in the PV model. LSQ is dominated by the $A^T A$ operation which results in a complexity of $\mathcal{O}(n^2 p)$ with n equal to the number of anchors and $p = 2$ for 2-dimensional localization. PSO has two inner loops for population size n and an outer loop for p iterations which results in a complexity of $\mathcal{O}(n^2 p)$.

CHAPTER 5

SIMULATION

In this chapter the simulation environment and results are presented. A total of three environments were simulated: a house, an office, and a store. The simulation used the model presented in Table 5.1 to represent the LOS and NLOS measurement error. PKFF was compared against the baseline multilateration algorithm described in Chapter 2 (LSQ as well as the PSO localization (PSO) and Kalman-filtered LSQ (KF) algorithms.

Table 5.1: Simulation Parameters

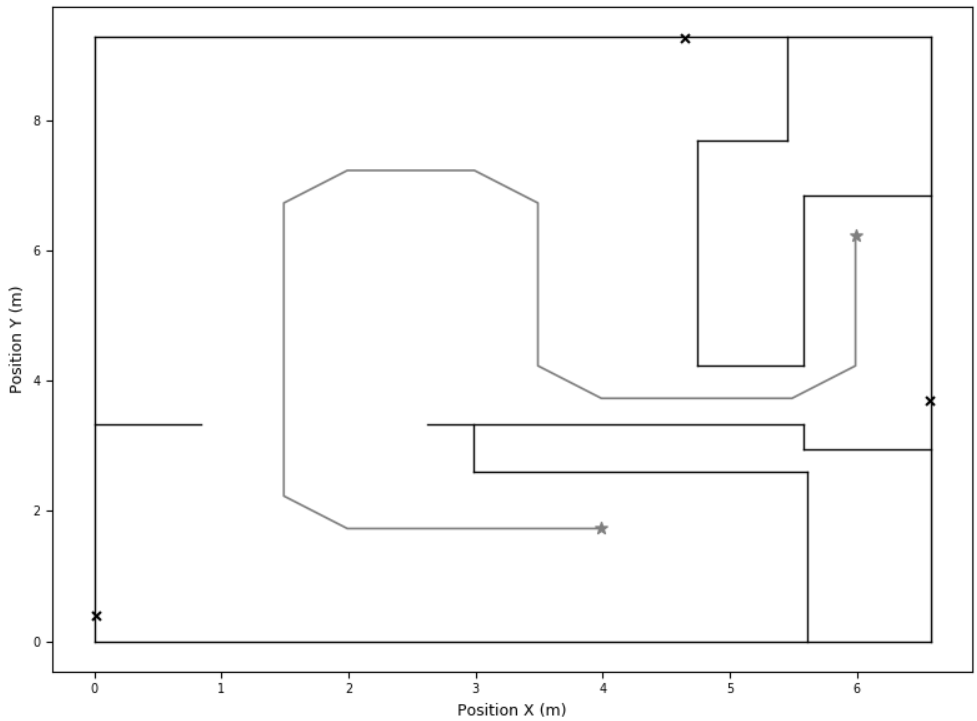
Parameter	***Model***
LOS Measurement Error	$N(\mu_{LOS}, \sigma_i^2)$
NLOS Measurement Error	$N(\mu_{NLOS}, \sigma_i^2 + \sigma_{NLOS}^2)$

5.1 Simulation Environments

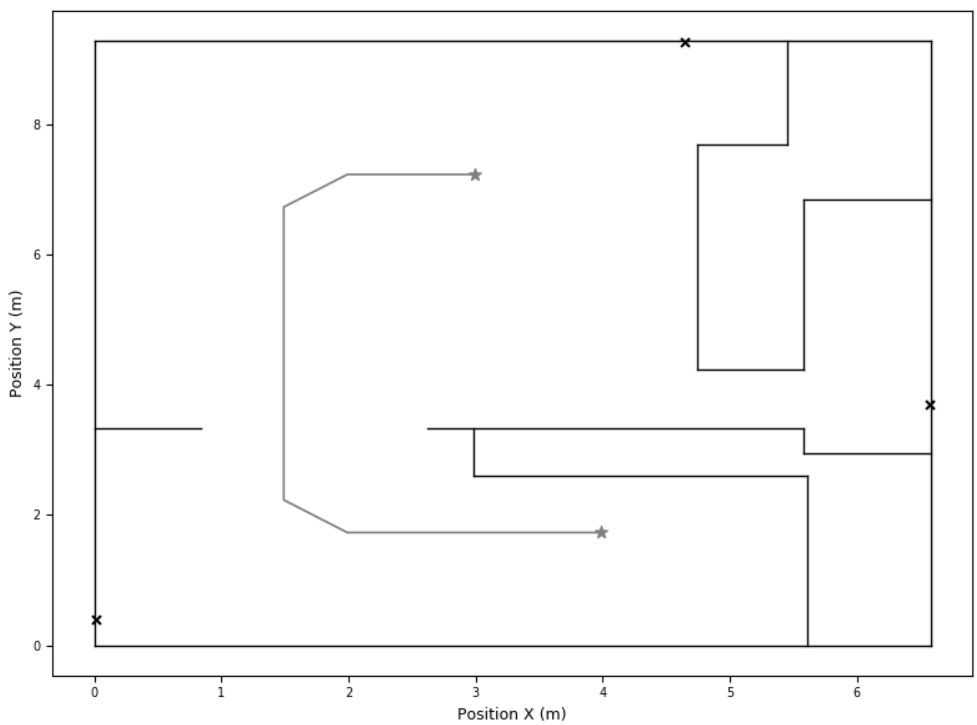
The house environment is a 61 m^2 section of a house. Two different paths were simulated, referred to as Path 1 and Path 2, and are shown in Figure 5.1 with 3 anchors. Path 1 had a total distance of 17 m with 33 measurement points and Path 2 a distance of 8.91 m with 18 measurement points. The remaining two environments are a 163 m^2 office and 223 m^2 grocery store environment shown in Figures 5.2 and 5.3. They were both simulated with 4 anchors.

5.2 Performance Metric

The localization error (LE) is the euclidean distance between the estimated position and the true position as given by Equation 5.1



(a) House Path 1



(b) House Path 2

Figure 5.1: House Environment Simulation Paths

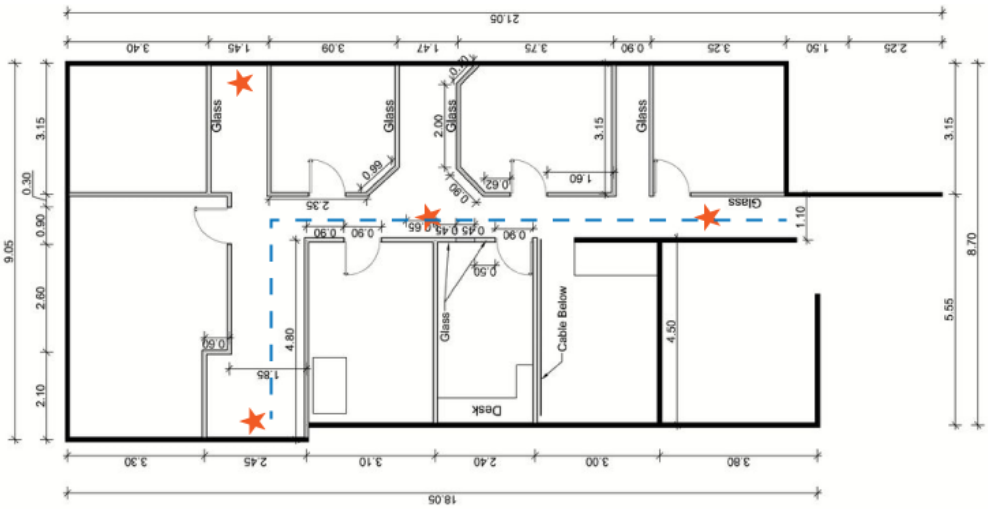


Figure 5.2: Office Environment Simulation Path

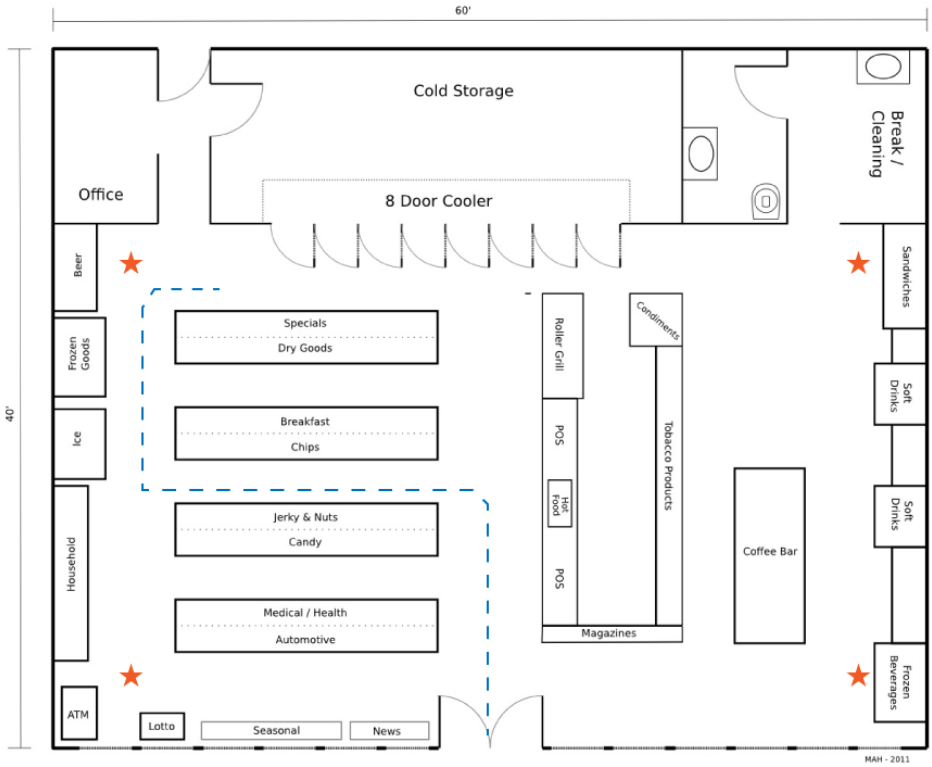


Figure 5.3: Store Environment Simulation Path

$$LE = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \tag{5.1}$$

with (x, y) being the true position and $(\hat{x}, \hat{y})$ being the estimated position of the mobile node. The primary metric used to measure performance is the Root Mean Square Error (RMSE) of the entire path. The RMSE is a useful measure for localization error because it combines both accuracy, accounting for constant error (CE), and precision, which accounts for random error (RE) (Letowski and Letowski 2011). The RMSE is defined as:

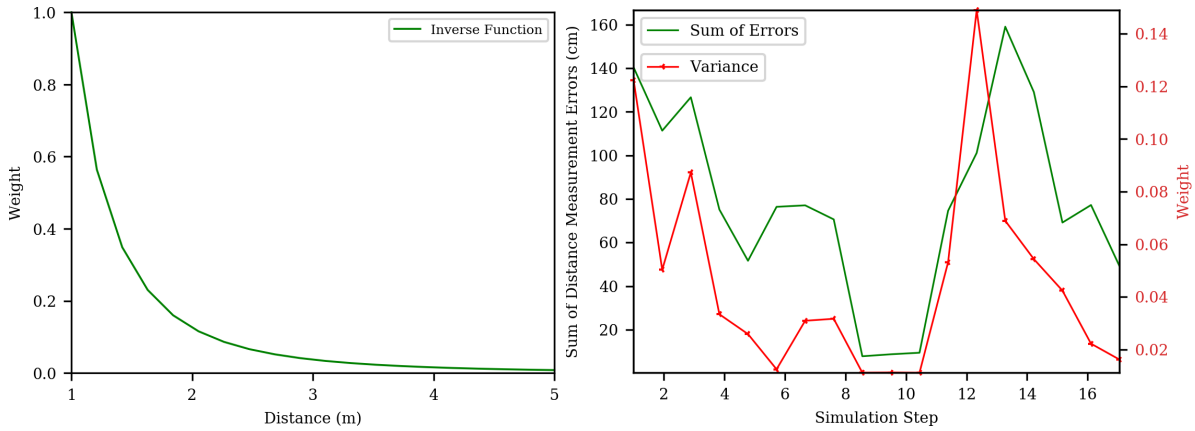
$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(LE_i)^2} \tag{5.2}$$

Where n is the number of simulation steps and LE is the localization error given by Equation 5.1.

5.3 Weighted Variance

As described in Chapter 4, the weighted variance is a value that uses all the distance measurements and associated variances to each anchor to serve as a rough approximation of the combined measurement noise for a set of measurements – this function can be seen in Figure 5.4 (a). In order to demonstrate how this weight correlates to the true distance measurement error, the calculated weight was plotted against the sum of all the distance errors to each anchor at every simulation step with a simulation measurement noise of $\sigma^2 = 5$ cm.

Figure 5.4 (b) also shows that the weighted variance generally follows the distance measurement errors and as such is able to serve as a good approximation of measurement noise. This allows the PKFF algorithm to better handle large variations in measurement error.



(a) Inverse Distance Weighting Function (b) Weighted Variance vs Measurement Error Sum

Figure 5.4: Weighted Variance

5.4 Performance with Increasing Variance

In order to test the performance of PKFF in an environment with changing measurement noise variance, a simulation was performed using Path 1 of the house environment where the variance of measurement noise was linearly increasing from 1 cm to 200 cm at each simulation step.

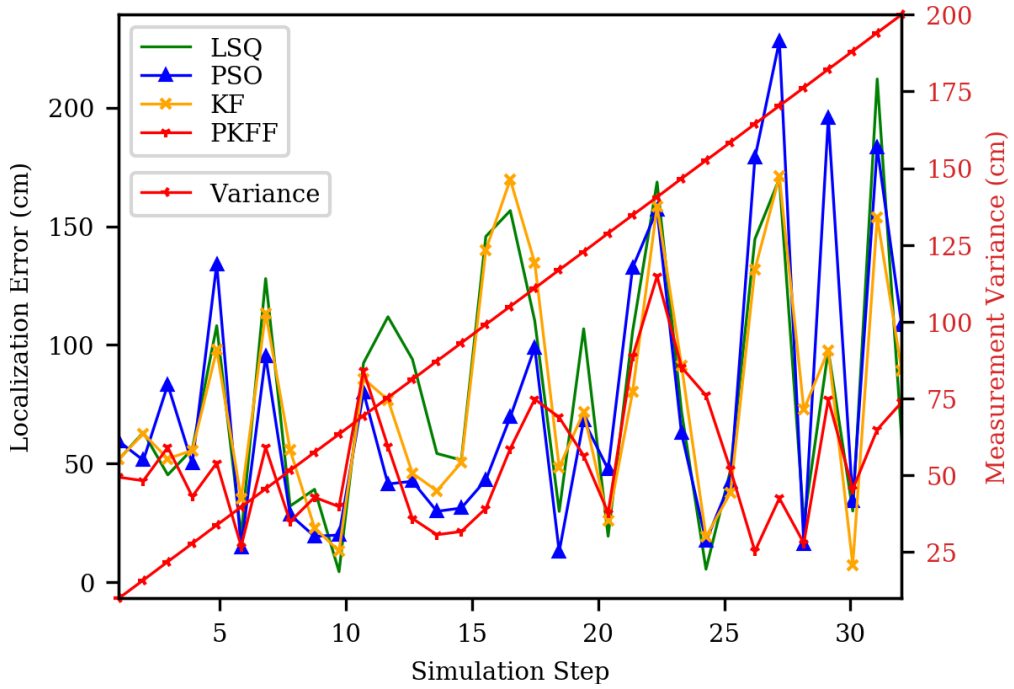


Figure 5.5: Localization Error with Increasing Variance

Table 5.2: Total RMSE

Algorithm	RMSE (cm)	Standard Deviation (cm)
LSQ	96.2	52.7
PSO	94.9	57.9
KF	90.2	46.2
PKFF	57.6	26.4

As shown in Figure 5.5 and Table 5.2, PKFF was able to achieve a 40% lower RMSE at 57.6 cm compared to LSQ at 96.2 cm. PKFF also had a 50% lower standard deviation of localization error compared to LSQ.

5.5 Performance with Increasing Anchors

The relationship between the number of anchors and RMSE is shown in Figure 5.6 at $\sigma^2 = 100$ cm. It can be seen that while increasing the anchor count generally starts providing diminishing returns in localization accuracy, in each case PKFF had the lowest RMSE which was 39 % lower than LSQ with three anchors and 27.3 % lower with six anchors. At 57.4 cm, PKFF still had a lower RMSE using only three anchors than LSQ, PSO, and KF using 6 anchors at 61.9 cm, 62.9 cm, and 59.2 cm, respectively. This shows that PKFF can serve as an alternative to adding more anchors for improving localization accuracy.

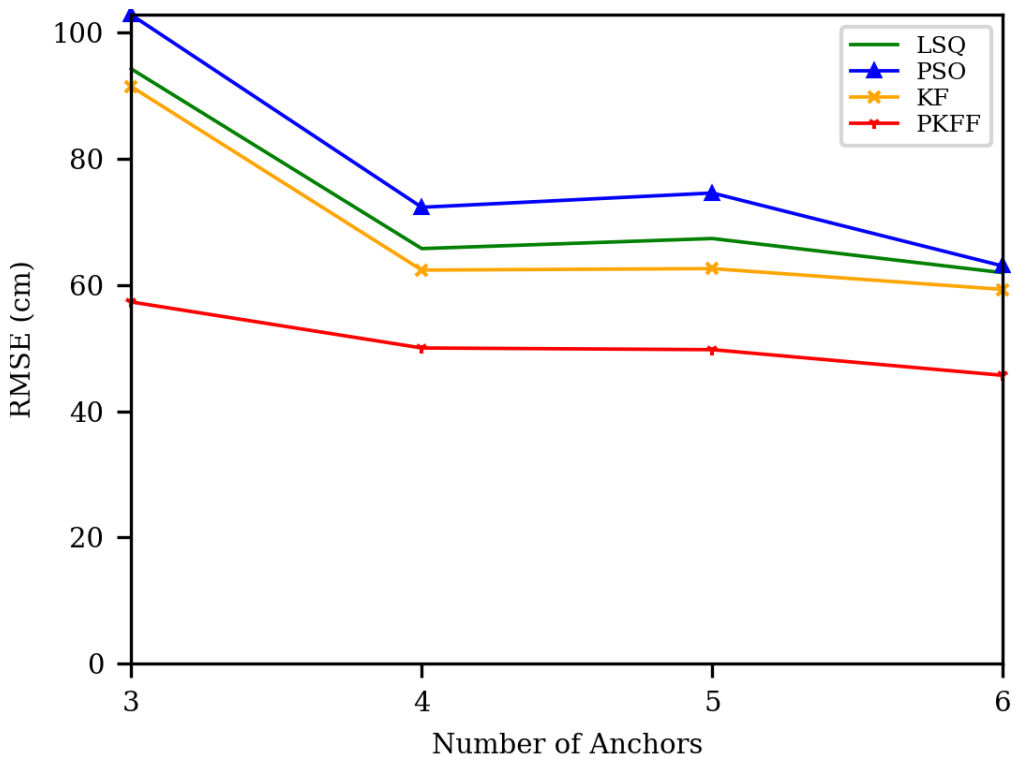


Figure 5.6: RMSE vs Number of Anchors

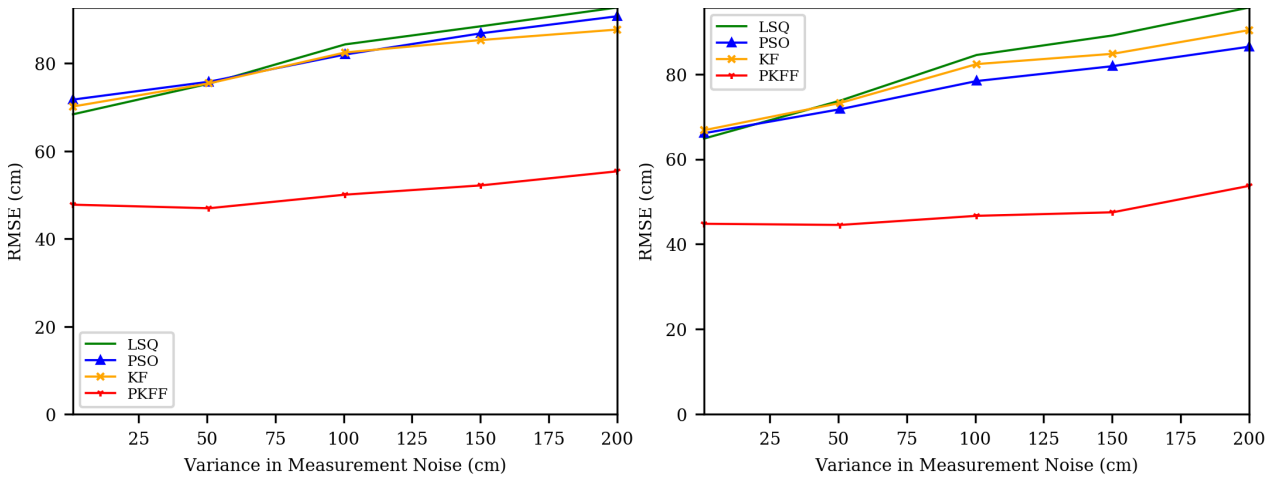
5.6 Overall Performance Simulation Results

To show the overall performance of PKFF, 500 simulation runs were performed in the house environment on both Path 1 and Path 2. Each simulation was run at 5 difference levels of variance in measurement noise $\sigma^2 = 1, 50, 100, 150, 200$.

Table 5.3: Path 1 Standard Deviation of RMSE with Increasing Variances

	$\sigma^2 = 1cm$	$\sigma^2 = 50cm$	$\sigma^2 = 100cm$	$\sigma^2 = 150cm$	$\sigma^2 = 200cm$
LSQ	23.67	20.08	18.21	17.08	14.48
PSO	28.17	25.41	24.5	23.74	22.17
KF	25.38	23.11	21.19	19.53	15.88
PKFF	15.36	10.82	9.93	9.58	11.14

Figure 5.7 show the RMSEs of all the simulation runs at each level of variance in measurement noise. PKFF had the lowest RMSE at each noise level for both paths. In Path 1, PKFF had



(a) Path 1 (b) Path 2

Figure 5.7: RMSE at Different Variances in Measurement Noise

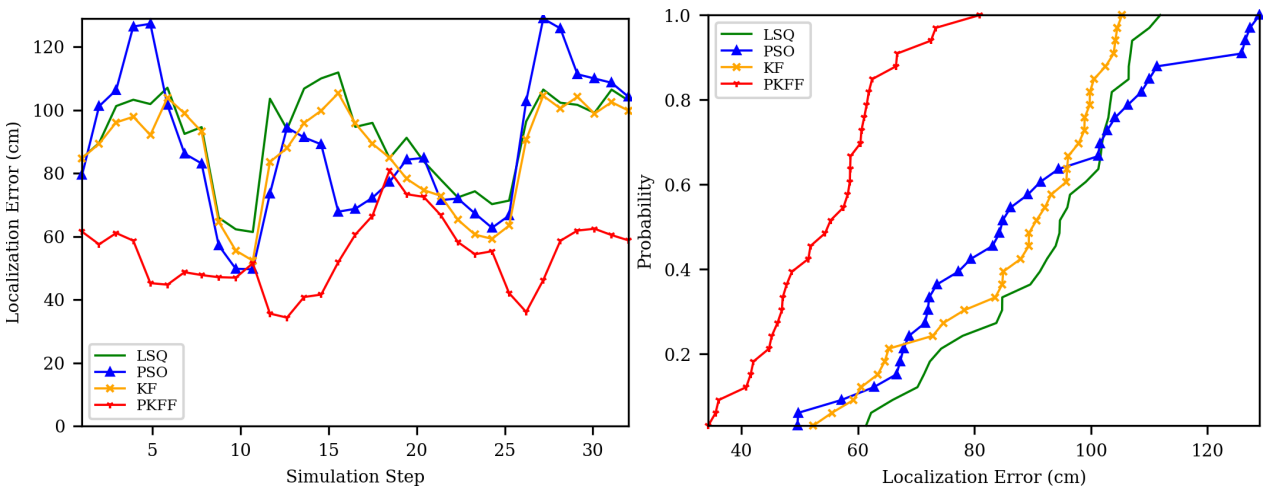
Table 5.4: Path 2 Standard Deviation of RMSE with Increasing Variances

	$\sigma^2 = 1cm$	$\sigma^2 = 50cm$	$\sigma^2 = 100cm$	$\sigma^2 = 150cm$	$\sigma^2 = 200cm$
LSQ	26.51	19.66	16.57	16.94	15.72
PSO	27.84	24.72	23.73	21.91	21.56
KF	26.17	20.67	18.93	16.99	17.51
PKFF	17.53	12.7	11.52	8.47	10.49

the smallest RMSE increase from the lowest to highest noise levels with a 15.9 % increase while LSQ, PSO, and KF had increases of 35.5 %, 26.5 %, and 24.9 % respectively. Similarly for Path 2, PKFF had a 19.8 % increase while LSQ, PSO, and KF had increases of 47.7 %, 30.8 %, and 35.3 % respectively. As shown in Tables 5.3 and 5.4, PKFF also had the lowest standard deviations at every measurement noise level for each path, showing that the algorithm is not tuned to a certain noise profile but is able to automatically adjust.

5.6.1 High Measurement Noise

This section shows the individual localization errors at a high measurement noise of $\sigma^2 = 200$ cm for Path 1. As seen in Figure 5.8, PKFF has the lowest localization error of the four algorithms with a 90-percentile RMSE of 66.7 cm while LSQ, PSO, and KF had 106.7 cm, 121.5 cm, and 103.5 cm, respectively.

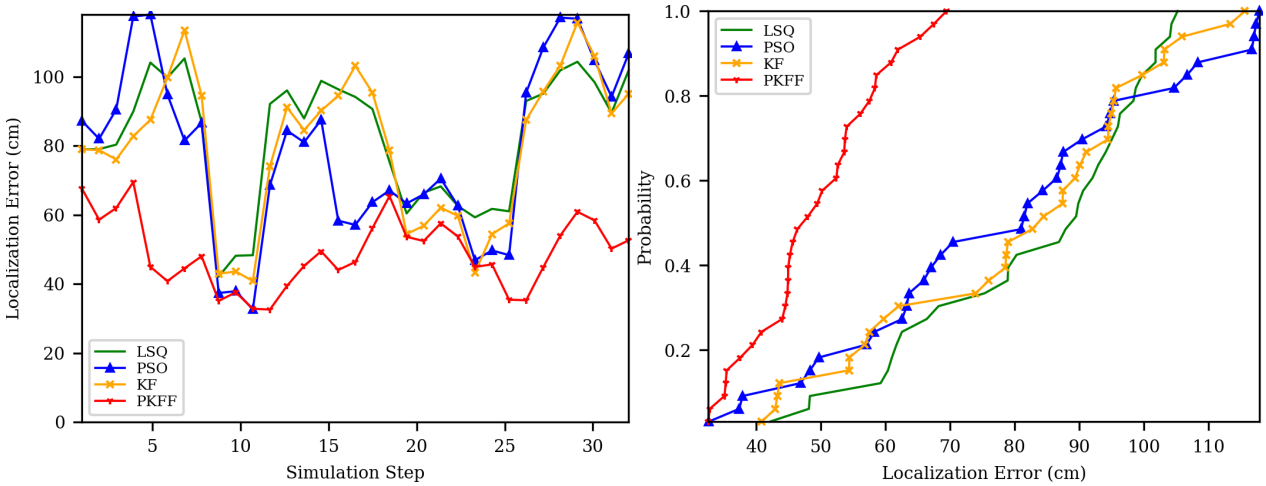


(a) Localization Error (b) CDF

Figure 5.8: Path 1 at $\sigma^2 = 200$ cm

5.6.2 Medium Measurement Noise

This section shows the individual localization errors of each algorithm at a medium measurement noise variance of $\sigma^2 = 100cm$ for Path 1. Similarly to the first case and as shown in Figure 5.9, PKFF has the lowest localization error of the four algorithms with a 90-percentile RMSE of 61.6 cm while LSQ, PSO, and KF were 101.8 cm, 114.2 cm, and 103.2 cm, respectively.

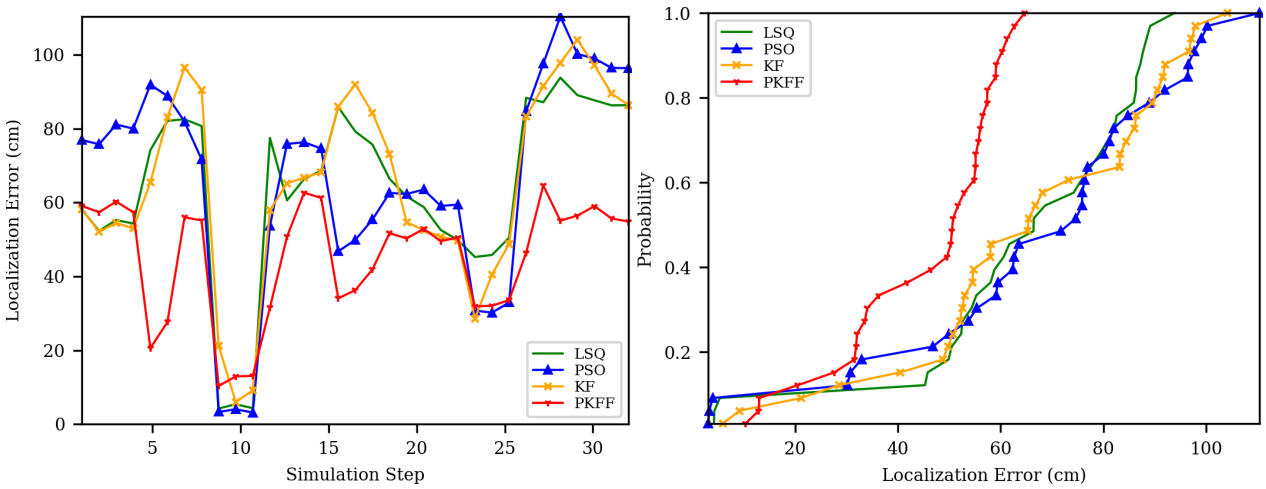


(a) Localization Error (b) CDF

Figure 5.9: Path 1 at $\sigma^2 = 100\,\text{cm}$

5.6.3 Low Measurement Noise

Finally PKFF was simulated at low measurement noise levels $\sigma^2 = 1cm$. Even at this low measurement noise level PKFF outperformed the other three algorithms. As shown in Figure 5.10, PKFF had a 90-percentile RMSE of 59.9 cm while LSQ, PSO, and KF had 87.5 cm, 97.3 cm, and 95.2 cm, respectively.



(a) Localization Error (b) CDF

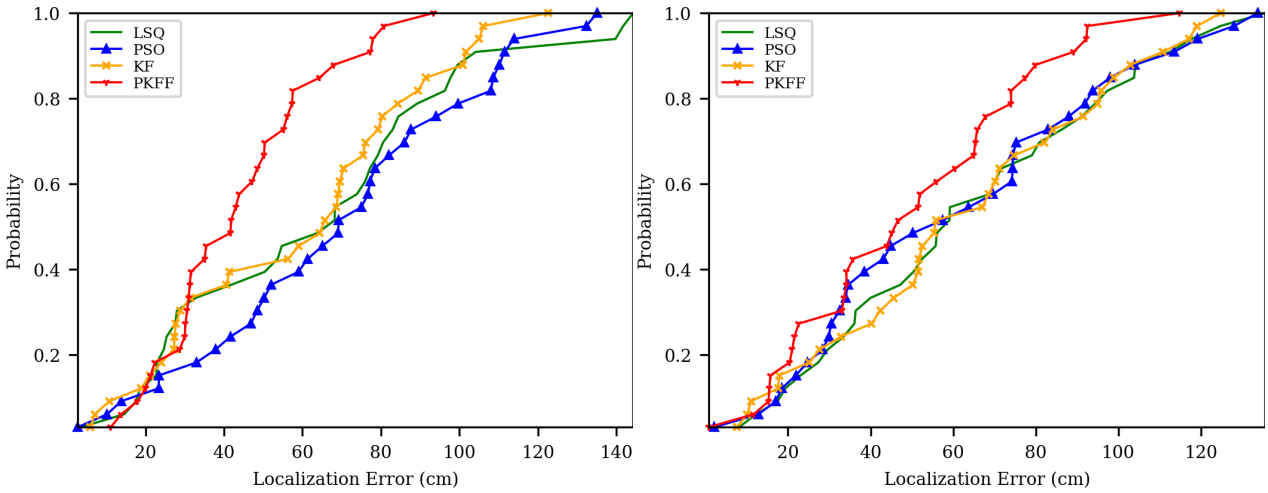
Figure 5.10: Path 1 at $\sigma^2 = 1\,\text{cm}$

5.7 Office and Store Environments

The office and store environment simulations each used 4 anchors at $\sigma^2 = 100\,\text{cm}$. PKFF had the lowest RMSE compared to LSQ, PSO, and KF for both environments as shown in Table 5.5. The RMSE for the store was 17 % higher than the office due to NLOS conditions because a higher percentage of the path was in NLOS conditions due to shelving. Subsequently, the office 90-percentile RMSE for PKFF was 75.2 cm compared to 87.7 cm for the store as shown in Figure 5.11.

Table 5.5: Simulation RMSE at $\sigma^2 = 100\,\text{cm}$

	LSQ	PSO	KF	PKFF
Office	74.1	77.3	67.1	48.2
Store	72.4	79.2	71.1	56.8



(a) Office (b) Store

Figure 5.11: Office and Store CDF at $\sigma^2 = 100\,\text{cm}$

5.8 Simulation Limitations

While simulations are quite important in the development of an algorithm, they have a number of limitations that must be considered. In the context of this work, the biggest limitation of the simulation environment was the fact that it is not practical, or possible, to accurately model all the complex factors that affect the accuracy of distance measurements in an indoor environment. For instance, the simulation map was modeled after a section of a house but the map did not include the couches, tables, and other items on the floor or hanging on the walls that would in real life have some effect on the propagation of radio waves in the environment. This also includes the material of the walls and obstacles. Another limitation of simulations is that they alone can only give an idea of relative precision (how the simulation results compare to other simulation results) but not accuracy (how the simulation results compare to the real world) and so validation experiments are necessary.

CHAPTER 6

HARDWARE VALIDATION

PKFF was evaluated on a testbed utilizing the DWM1001 Ultra-wideband (UWB) modules from Decawave. These modules are part of a commercial Real-Time Location System (RTLS) developed by Decawave and can be useful for evaluating range-based algorithms. In the testbed, the modules were not used as part of a default Decawave RTLS but were instead used as generic UWB radios for ranging purposes, utilizing the proprietary two-way ranging algorithm developed by Decawave that provides centimeter-level accuracy for indoor positioning. The modules can be configured as anchors or tags, with anchors considered part of the fixed infrastructure with known locations and tags the moving node to be localized.

Figure 6.1: DecaWave UWB Radios

6.1 Experimental Setup

The testbed was the same house environment used in simulation and utilized three UWB modules configured as anchors and one configured as a tag. The anchors where placed along the perimeter at a height of 2 m in exactly the same areas as used for the simulation. The path used for the testing is shown in Figure 6.3.

(a) Anchor 1 (b) Anchor 2

Figure 6.2: UWB Anchors Hanging on Wall

The path consisted of 33 total positions and 10 measurements were taken at each position. This path was very similar to that of the Path 1 simulations. Ground truth was determined by measuring the position of each step and marking it with tape prior to the measurement campaign as shown in Figure 6.4.

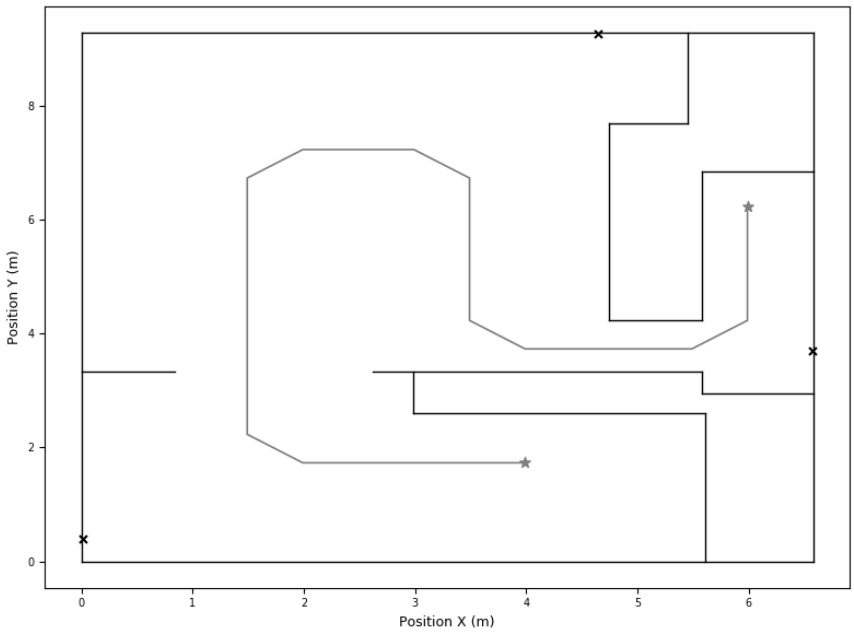


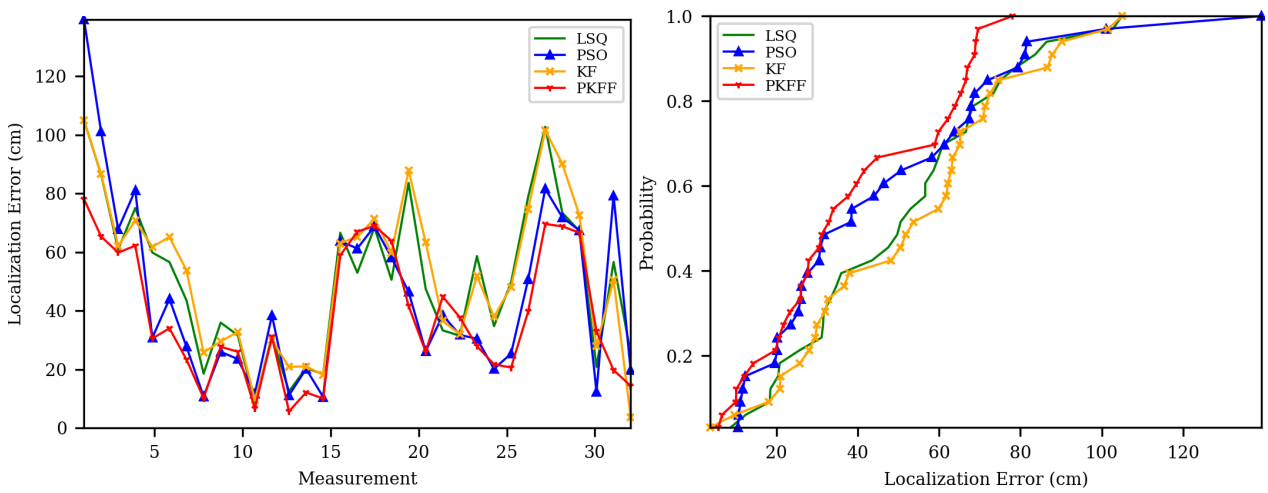
Figure 6.3: Experimental Testbed Path

(a) Experimental Path (b) Experimental Path

Figure 6.4: Testbed Path Markers

6.2 Experimental Results

With a RMSE of 44.1 cm the testbed performance was on par with the low measurement noise simulation RMSE of (47.8 cm), which is consistent with the use of low-noise UWB hardware. LSQ, PSO, and KF had RMSEs of 56.4 cm, 60.8 cm, and 58.6 cm, respectively. As shown in Figure 6.5 (a), the influence of the KF prevented any large spikes in localization error. The influence of the weighted variance can also be seen in how PKFF starts to mirror the PSO algorithm in low-noise conditions since the fusion becomes biased towards PSO.



(a) Localization Error (b) CDF

Figure 6.5: Experimental Results

Looking at the CDF in Figure 6.5 (b), PKFF has a 90-percentile RMSE of 67.6 cm while KF, PSO, and EKF had values of 82.2 cm, 80.6 cm, and 87.5 cm, respectively.

6.3 NLOS Characterization Experiment

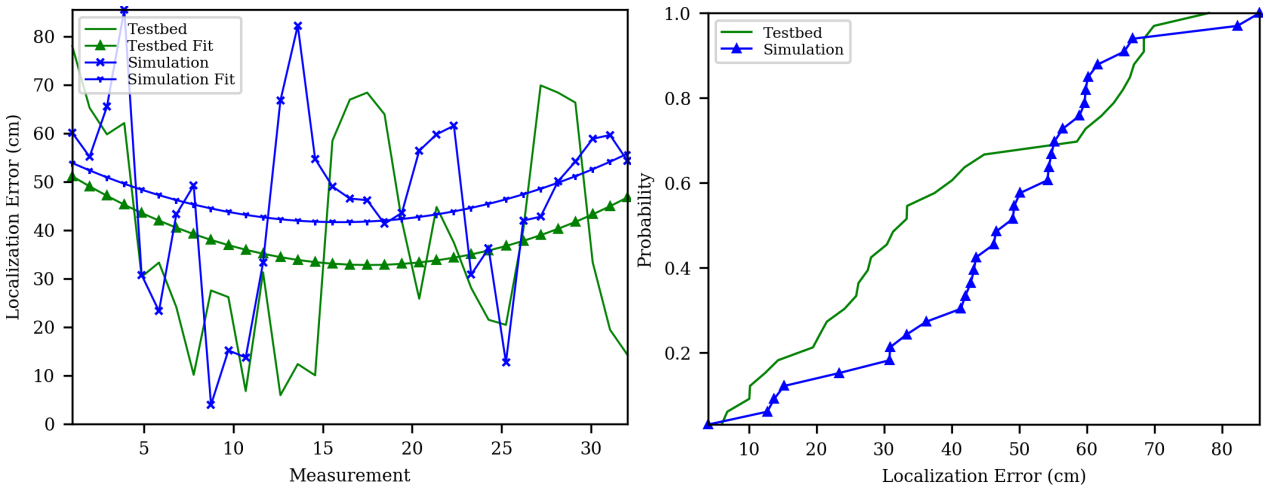
An experiment was created in order to characterize the effects of NLOS using the UWB hardware. Two UWB radios were first placed in direct LOS at a distance of 4.3 m and then placed at the same distance with a wall in between for NLOS. 1000 measurements were taken for each condition with the results given in Table 6.1. This data was then used in the simulation model.

Table 6.1: Hardware NLOS Characterization Results

	Mean of Error (cm)	Variance (cm)
LOS	9.3	8.2
NLOS	61.2	12

6.4 Simulation Validation

Finally, PKFF was validated by performing a direct comparison of the experimental testbed results with a simulation. The simulation was configured such that the measurement variances to each anchor of the simulation were replaced with those from the testbed run at each step for the same path. Results are shown in Figure 6.6.



(a) Validation results with fitting curves (b) Validation CDF

Figure 6.6: Validation Results

The testbed and simulation had very similar fits with RMSEs of 44 cm and 49.8 m for the testbed and simulation, respectively. The testbed also had a 90-percentile RMSE of 67.2 cm while the simulation's RMSE was 64.6 cm. This shows that the simulations present an accurate representation of the performance of PKFF.

6.5 Experimental Limitations

While the experimental testbed was critical for the validation of the simulations and algorithm in general, it still had some key limitation. The biggest limitation of the testbed was the fact that only one type of measurement hardware was used. The UWB hardware used had an extremely low measurement noise floor so the only direct comparison that could be made was with the low noise simulations. Another limitation was the accuracy of the ground truth. Even though great care was taken to ensure all the measurement points in the experimental path were accurately measured, there was still some inherent uncertainty in the ground truth measurements. This again ties into why validating the simulations is important since simulation can have perfect ground truth.

CHAPTER 7

DISCUSSION

The simulation and experimental results showed that PKFF outperformed the other algorithms by having the lowest RMSE in every case. When looking at the overall performance in Figures 5.7, the first observation is how flat the curve for PKFF is compared to the other algorithms through the increasing levels of measurement noise. This can be attributed to the dynamic nature of PKFF and how it uses the weighted variance as a joint measure of measurement uncertainty. Another observation is how the very low measurement noise simulations had higher standard deviations of error for PKFF. This is due to PKFF relying mainly on the PSO algorithm when it detects low noise levels, which is not as stable as a KF. Even with that, PKFF is still more accurate in low measurement noise scenarios. The low and high measurement noise simulations were also very useful in demonstrating that PKFF was actually dynamic and not just tuned to suppress high measurement noise errors. This was evident in the cases where there were large jumps in localization errors in the other algorithms but not PKFF due to the weighted variance estimate.

7.1 Limitation of PKFF

While the performance of PKFF has proven to be significantly better than the comparisons, it still has some limitations which must be considered. The most important limitation is that PKFF performs relative to the accuracy of measurement method for the ranging data, which is true of all range-based algorithms. In the case of this work, UWB noise characteristics were used so the accuracy was centimeter-level. Using a laser ranging method would result in millimeter-level accuracy while using something like RSSI would result in meter-level accuracy. Even in all these cases, PKFF still provides more accurate ranging relative to the particular ranging method utilized. Another limitation is that since PKFF relies on measurement variances, it can sometimes be fooled into a false negative scenario where the joint weighted variance doesn't actually correspond to the

measurement error for a particular measurement. This is most evident in Figure 5.4 that compared the weighted variance to the total distance error where there is a single point that overestimates the error. The fusing KF is therefore important in helping stabilize the estimates.

7.2 Lessons Learned

The simulations and testbed experiment were all very important in the development of PKFF. Even with the intuition and theory behind the operation of PKFF in mind, it was extremely difficult to make progress without having a proper simulation configured. The greatest factor in the effectiveness of the simulations was making the simulation environment as close to the experimental testbed as possible. Since the basic model used for LOS / NLOS errors was still valid in simulation, it provided important insight into the behavior and overall effectiveness of the algorithm. Because the simulation environment was modeled after the experimental testbed, the testbed was able to successfully validate the simulations and as a result validate PKFF. This underscores the importance of making the simulation as close to the experimental setup as possible.

CHAPTER 8

CONCLUSION

Simulations and validation with an experimental testbed show that PKFF meets its stated goal of improving the accuracy of range-based localization in mixed LOS/NLOS environments without the use of any additional hardware or fingerprinting. PKFF consistently had the lowest RMSE across all tested levels of measurement noise and environments when compared to the LSQ, PSO and KF algorithms. With a 90-percentile RMSE of 61 cm, PKFF had a 40 % lower RMSE compared to LSQ which had a 90-percentile RMSE of 102 cm while PSO and KF had 114 cm and 103 cm, respectively. Results also showed that PKFF performed better using only three anchors than the other algorithms tested with 6 anchors. Even though it is possible to gain even more accurate results by incorporating sensors like IMUs, PKFF demonstrates that accuracy improvements can still be gained with no additional hardware or tight-coupling to specific hardware, making PKFF extremely useful as a general solution for accuracy improvements in range-based systems.

8.1 Future Work

There are a number of opportunities for further research, starting with testing various methods of ranging such as RSSI. This would allow PKFF to be evaluated with even more hardware and measurement noise conditions. Secondly, the effects of various motion profiles on localization accuracy is another area of research that would prove useful to extending the effectiveness of PKFF. Finally, just having a more accurate ground truth measurement scheme such as using camera tracking with fiduciary markers or laser-based ranging would allow for the validation of more dynamic scenarios.

REFERENCES

Ahmadi, Hanen, Federico Viani, Alessandro Polo, and Ridha Bouallegue. 2017. Learning ensemble strategy for static and dynamic localization in wireless sensor networks [in en]. Fusion, *International Journal of Network Management* 27 (4): e1979.

Alberto, Julián, and Patino Murillo. 2011. Kalman-based schemes for mobile nodes localization in ad-hoc networks Esquemas de localizacion de nodos moviles en redes ad-hoc basados en filtros de Kalman.

Bailey, T., and H. Durrant-Whyte. 2006. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics Automation Magazine* 13, no. 3 (September): 108–117.

Belmonte-Hernández, A., G. Hernández-Peñaloza, F. Álvarez, and G. Conti. 2017. Adaptive Fingerprinting in Multi-Sensor Fusion for Accurate Indoor Tracking. Fusion, *IEEE Sensors Journal* 17, no. 15 (August): 4983–4998.

Ben Ayed, Siwar, Hanene Trichili, and Adel M. Alimi. 2015. Data fusion architectures: A survey and comparison. In *2015 15th International Conference on Intelligent Systems Design and Applications (ISDA),* 277–282. Marrakech, Morocco: IEEE, December.

Bhattacharya, Shrabani, and R. Appavu Raj. 2004. Performance evaluation of multi-sensor data fusion technique for test range application [in en]. *Sadhana* 29, no. 2 (April): 237–247.

Boudjemaa, Redouane, and Alistair Forbes. 2004. *Parameter estimation methods in data fusion.* February.

Bregar, Klemen, and Mihael Mohorčič. 2018. Improving Indoor Localization Using Convolutional Neural Networks on Computationally Restricted Devices. NLOS, *IEEE Access* 6:17429–17441.

Briese, D., H. Kunze, and G. Rose. 2017. UWB localization using adaptive covariance Kalman Filter based on sensor fusion. In *2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB),* 1–7. Fusion. September.

Brookner, Eli. 1998. Tracking and Kalman Filtering Made Easy.

Decarli, Nicolò, Davide Dardari, Sinan Gezici, and Antonio Alberto D'Amico. 2010. LOS/NLOS detection for UWB signals: A comparative study using experimental data. In *IEEE 5th International Symposium on Wireless Pervasive Computing 2010,* 169–173. NLOS. May.

Durrant-Whyte, Hugh F. 1988. Sensor Models and Multisensor Integration [in en]. *The International Journal of Robotics Research* 7, no. 6 (December): 97–113.

Fan, Jiancun, Susu Chen, Xinmin Luo, Ying Zhang, and Geoffrey Ye Li. 2019. A Machine Learning Approach for Hierarchical Localization Based on Multipath MIMO Fingerprints. ML, *IEEE Communications Letters* 23, no. 10 (October): 1765–1768.

Gopakumar, A., and L. Jacob. 2008. Localization in wireless sensor networks using particle swarm optimization. In *2008 IET International Conference on Wireless, Mobile and Multimedia Networks,* 227–230. January.

Gu, T., Y. Tang, R. Wang, L. Lu, Z. Wang, and L. Chang. 2019. Indoor Localization Fusion Algorithm Based on Signal Filtering optimization Of Multi-sensor. In *2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI),* 250–255. Fusion. June.

Habtie, Ayalew Belay, Ajith Abraham, and Dida Midekso. 2015. Comparing Measurement and State Vector Data Fusion Algorithms for Mobile Phone Tracking Using A-GPS and U-TDOA Measurements. In *Hybrid Artificial Intelligent Systems,* edited by Enrique Onieva, Igor Santos, Eneko Osaba, Héctor Quintián, and Emilio Corchado, 9121:592–604. Cham: Springer International Publishing.

Kaya, S. B., and A. Z. Alkar. 2018. Indoor localization and tracking by multi sensor fusion in Kalman filter. In *2018 26th Signal Processing and Communications Applications Conference (SIU),* 1–4. Fusion. May.

Kennedy, J., and R. Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks,* vol. 4, 1942–1948 vol.4. November.

Kumar, Sudhir, and Rajesh M. Hegde. 2017. A Review of Localization and Tracking Algorithms in Wireless Sensor Networks. Survey, *arXiv:1701.02080 [cs]* (January). arXiv: `1701.02080 [cs]`.

Labbe, Roger. 2018. *Kalman and Bayesian Filters in Python.*

Landolsi, M. A., and R. Shubair. 2018. TOAI/AOA/RSS Maximum Likelihood Data Fusion for Efficient Localization in Wireless Networks. In *2018 15th International Multi-Conference on Systems, Signals Devices (SSD),* 458–462. Fusion. March.

Letowski, Tomasz, and Szymon Letowski. 2011. Localization error: Accuracy and precision of auditory localization. April.

Lu, George, and David Wong. 2008. An adaptive inverse-distance weighting spatial interpolation technique. *Computers & Geosciences* 34 (September): 1044–1055.

Mao, Guoqiang, and Baris Fidan. 2009. *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking* [in English]. IGI Global.

Mitchell, H B. 2012. *Data Fusion: Concepts and Ideas* [in en]. Berlin, Heidelberg: Springer Berlin Heidelberg.

Miyagusuku, R., A. Yamashita, and H. Asama. 2019. Data Information Fusion From Multiple Access Points for WiFi-Based Self-localization. Fusion, *IEEE Robotics and Automation Letters* 4, no. 2 (April): 269–276.

Mosallaei, M., K. Salahshoor, and M. R. Bayat. 2007. Process Fault Detection and Diagnosis by Synchronous and Asynchronous Decentralized Kalman Filtering using State-Vector Fusion Technique. In *Sensor Networks and Information 2007 3rd International Conference on Intelligent Sensors,* 209–214. December.

Nastac, Dumitru-Iulian, Florentin Alexandru Iftimie, Octavian Arsene, Virgil Ilian, and Bogdan Cramariuc. 2017. Indoor positioning WLAN based fingerprinting as supervised machine learning problem. In *2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME),* 194–199. ML. October.

Noel, Mathew M., Parag P. Joshi, and Thomas C. Jannett. 2006. Improved Maximum Likelihood Estimation of Target Position in Wireless Sensor Networks using Particle Swarm Optimization. *Third International Conference on Information Technology: New Generations (ITNG'06):* 274–279.

Paul, Anup Kumar, and Takuro Sato. 2017. Localization in Wireless Sensor Networks: A Survey on Algorithms, Measurement Techniques, Applications and Challenges [in en]. Survey, *Journal of Sensor and Actuator Networks* 6, no. 4 (December): 24.

Ramesh, Maneesha V., P. L. Divya, Raghavendra V. Kulkarni, and Rekha Manoj. 2012. A Swarm Intelligence Based Distributed Localization Technique for Wireless Sensor Network. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics,* 367–373. ICACCI '12. Chennai, India: ACM.

Röbesaat, Jenny, Peilin Zhang, Mohamed Abdelaal, and Oliver E. Theel. 2017. An Improved BLE Indoor Localization with Kalman-Based Fusion: An Experimental Study. In *Sensors.* Fusion.

Sadowski, S., and P. Spachos. 2018. RSSI-Based Indoor Localization With the Internet of Things. *IEEE Access* 6:30149–30161.

Safavi, S., U. A. Khan, S. Kar, and J. M. F. Moura. 2018. Distributed Localization: A Linear Theory. *Proceedings of the IEEE* 106, no. 7 (July): 1204–1223.

Salamah, Ahmed H., Mohamed Tamazin, Maha A. Sharkas, and Mohamed Khedr. 2016. An enhanced WiFi indoor localization system based on machine learning. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN),* 1–8. ML. October.

Schroeder, Jens, Stefan Galler, Kyandoghere Kyamakya, and Klaus Jobmann. 2007. NLOS detection algorithms for Ultra-Wideband localization. In *Navigation and Communication 2007 4th Workshop on Positioning,* 159–166. NLOS. March.

Shareef, Ali, and Yifeng Zhu. 2009. Localization Using Extended Kalman Filters in Wireless Sensor Networks. April.

Singh, Yashpal, and Rajesh Mehra. 2015. Relative Study of Measurement Noise Covariance R and Process Noise Covariance Q of the Kalman Filter in Estimation.

Wang, Fafa, Tingli Su, Xuebo Jin, Yangyang Zheng, Jianlei Kong, and Yuting Bai. 2019. Indoor Tracking by RFID Fusion with IMU Data [in en]. Fusion, *Asian Journal of Control* 21 (4): 1768–1777.

Wang, S., D. Hu, X. Sun, S. Yan, J. Huang, W. Zhen, and Y. Li. 2018. A Data Fusion Method of Indoor Location Based on Adaptive UKF. in *2018 7th International Conference on Digital Home (ICDH),* 257–263. Fusion. November.

Wang, Yan, Jinquan Hang, Long Cheng, Chen Li, and Xin Song. 2018. A Hierarchical Voting Based Mixed Filter Localization Method for Wireless Sensor Network in Mixed LOS/N-LOS Environments [in en]. NLOS, *Sensors* 18, no. 7 (July): 2348.

Wu, Chi, Hongwei Hou, Wenjin Wang, Qing Huang, and Xiqi Gao. 2018. TDOA Based Indoor Positioning with NLOS Identification by Machine Learning. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP),* 1–6. NLOS. October.

Xu, W., Z. Wang, and S.A. Zekavat. 2011. Non-line-of-sight identification via phase difference statistics across two-antenna elements. NLOS, *IET Communications* 5, no. 13 (September): 1814–1822.

Yassin, A., Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios. Secondquarter 2017. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. Survey, *IEEE Communications Surveys Tutorials* 19 (2): 1327–1346.

Yi, L., S. G. Razul, Z. Lin, and C. M. See. 2014. Target Tracking in Mixed LOS/NLOS Environments Based on Individual Measurement Estimation and LOS Detection. NLOS, *IEEE Transactions on Wireless Communications* 13, no. 1 (January): 99–111.

Yu, Xiaosheng, Peng Ji, Ying Wang, and Hao Chu. 2017. *Mean Shift-Based Mobile Localization Method in Mixed LOS/NLOS Environments for Wireless Sensor Network* [in en]. https://www.hindawi.com/journals/js/2017/5325174/. Research Article. NLOS.

Zhang, Yunzhou, Wenyan Fu, Dongfei Wei, Jianjun Jiang, and Bing Yang. 2013. Moving target localization in indoor wireless sensor networks mixed with LOS/NLOS situations. NLOS, *EURASIP Journal on Wireless Communications and Networking* 2013, no. 1 (December): 291.

Zhao, Jianguo, and Jiegui Wang. 2017. WiFi indoor positioning algorithm based on machine learning. In *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC),* 279–283. ML. July.

Zhu, Yaping, Weiwei Xia, Feng Yan, and Lianfeng Shen. 2019. NLOS Identification via AdaBoost for Wireless Network Localization. NLOS, *IEEE Communications Letters* 23, no. 12 (December): 2234–2237.

Zou, Han, Baoqi Huang, Xiaoxuan Lu, Hao Jiang, and Lihua Xie. 2016. A Robust Indoor Positioning System Based on the Procrustes Analysis and Weighted Extreme Learning Machine. ML, *IEEE Transactions on Wireless Communications* 15, no. 2 (February): 1252–1266.