Wi-Fi Localization with a Case Study of Mobile Robot

Asst. Prof. Dr. Hamid M. Hasan\textsuperscript{a*}, Atheer J. Baqir\textsuperscript{b}

\textsuperscript{a,b}University of Technology, Control and systems Eng. Dept. Iraq
\textsuperscript{a}Email: Hamid2012net@gmail.com, \textsuperscript{b}Email: Atheerjaafar@yahoo.com

Abstract

Estimating the location of the mobile robot in an indoor environment is a fundamental challenge nowadays. The accuracy of positioning mechanisms such as GPS is often limited for indoor environments. In this study, we develop the feasibility of building an indoor localization system that is cost effective for large scale deployments, can operate over existing Wi-Fi networks, and can provide flexibility to accommodate new sensor observations as they become available. This paper focuses on the use of Wi-Fi for communicating with and localizing the robot by Wi-Fi received signal strength indication (RSSI) to calculate the distance between the robot and access point (AP). Also we need to figure out the accurate orientation of the platform in 2D environment.

Keywords: localization; mobile robot; Wi-Fi.

1. Introduction

Successful localization is a very important issue for indoor mobile robots [1]. In the past, a variety of approaches to localization have been explored, include using sonar [2], laser scanners [3], kinematic of mobile robot [4], infrared beacons [5] and computer vision [6]. With the increasing prevalence of wireless LAN, “Wi-Fi,” considerable work has been done on using signal strength measurements from Wi-Fi access points for localization [7]. Wi-Fi location determination systems use the popular 802.11b network infrastructure to determine the user location. This makes these systems attractive in indoor environments where traditional techniques, such as Global Positioning System (GPS) fail [8, 9]. Wireless networks have become a critical component of the networking infrastructure and are available in most corporate environments (universities, airports, train stations, tribunals, hospitals, etc.) and in many commercial buildings (cafes, restaurants, cinemas, shopping centers, etc.) [10, 11]. Then, new homes are slowly starting to add Wi-Fi services in order to enable mobility to perform many routine tasks. Thorough studies of Wi-Fi signal propagation properties.

* Corresponding author.
These studies determine the relationship between received signal strength (RSSI), bandwidth, link quality, and distance, which is required for understanding Wi-Fi AP deployment and communication [12].

There are multiple techniques implemented to determine location, one of them is called Wi-Fi fingerprinting where every point in the region of interest is covered by at least k Aps[13].

Another Wi-Fi Localization technique that was considered was Wi-Fi Triangulation [14], this technology uses Wi-Fi signals to estimate the distance between the user and the transmitter. The distance was used to generate a circle around each transmitter. Then, by getting the intersection of the three circles, the location of the user was pinpointed. If given the correct information, it will produce a unique answer.

As you can see above, all techniques are depends on more than one AP, in this paper we try to develop localization technique using single AP only.

2. Background

To determine the robot position we need the calculation of the distance between the AP and the receiver which mounted on the robot. This can be achieved using the path loss equation

2.1 Log distance path loss equation

In this section, localization system designed based on RSSI signal will be presented based on the log-distance path loss model [9] shown in equation (1).

\[ PL (dB) = PL (d0) + 10 n \log_10 \frac{d}{d0} \]  

Where

PL indicates the path loss level, the value of n depends on the surroundings and building type (assumed 3.25 in calculations), d0 is the close-in reference distance which is determined from measurements close to the transmitter, d is the distance between the transmitter and the receiver.

2.2 Kalman filter

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) solution of the least-squares method. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown [17]. Surly we notice that there a huge difference between true and calculated readings in some points, in this paper we will use this filter to find an intermediate path for the robot between spacing values [18].

Kalman filter can be described by the following two equations:

State equation:
\[ x_{k+1} = Ax_k + Bu_k + w_k \quad \ldots \ldots (2) \]

Output equation:

\[ y_k = Cx_k + z_k \quad \ldots \ldots \ldots \ldots \ldots (3) \]

In the above equations, \( A, B, \) and \( C \) are matrices; \( k \) is the time index; \( x \) is called the state of the system; \( u \) is a known input to the system; \( y \) is the measured output; and \( w \) and \( z \) are the noise. The variable \( w \) is called the process noise, and \( z \) is called the measurement noise. Each of these quantities is (in general) vectors and therefore contains more than one element.

The vector \( x \) contains all of the information about the present state of the system,

So if we will measure the position every \( T \) seconds, the \( A, B \) and \( C \) matrices will be.

Where

\[
A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}, \quad C = [1 \ 0]
\]

\[ w_k = 15.3 z_k = 0.4 \]

3. Method and tools

The range between the Wi-Fi source and the receiver after filtering is used with combination of inertial measurement unit (IMU) to accurately determine the mobile robot position. Real experimental environment is used for evaluation, which consisting a prototype of two wheel drive mobile robot (WMR) with (AP). The prototype WMR plus Wi-Fi receiver and IMU was built from scratch as shown in figure (1).

![Figure 1: WMR](image-url)
Where we use Wi-Fi shield as a Wi-Fi receiver, MBU 6050 as IMU to compute orientation of the robot where it gives us three values (Yaw, Pitch and Roll) here we just benefit from Yaw since it deal with horizontal movement, L298n motor driver with two motors to control robot motion [16], Arduino Uno and one AP.

Since the equation (1) will give us the distance without directions in term of(x,y), we can benefit from yaw value obtained from IMU which can give us the robot orientation.

To make sure of our calculations, we can use al-Kashi equation as shown in equation (4) for corrections [17].

\[
c^2 = a^2 + b^2 - 2ab \cos y \………………………… (4)
\]

Where

\(x\) is the Yaw value, \(y\) is the complementary of Yaw value \((180 - x)\), \(a\) is the distance between start point and AP, \(b\) is the distance between start point and robot, \(c\) is the distance between the robot and AP, as shown in fig (2).

![Figure 2: virtual triangle](image)

Putting all values in the equation (4) produce a second order equation with one unknown value \((b)\), then the equation will become.

\[
b^2 - (2acos y)b + (a^2 - c^2) = 0 \………….. (5)
\]

To solve this quadratic equation, we cannot use the quadratic formula since it gives an imaginary results, so the best way to solve this problem is by use a simple loop program that varying \(b\) value until the equation equal to zero, as follow in figure (3)
Now we need to generalize this formula to include the second and third position ... etc of the robot, see figure (4) where it is an extension of figure (2).

In the second position the Yaw value will be the difference between the last value and the current value of Yaw, while the angle between the rib c and the extended line from rib b equal to the summation of the Yaw value and robot angle, also the initial orientation of the robot should be known, in case of figure (2) equals 90˚, the relationship will become as follows.

\[ m = \text{robot initial orientation} + \text{Yaw} + \text{robot last angle} \]  

\[ m = \text{robot initial orientation} + \text{Yaw1} + \text{robot last angle} + (\text{Yaw2} - \text{Yaw1}) \]  

where

**robot initial orientation**: is the orientation at the start position

**robot last angle**: the last angle that the robot deviates from the AP (equal z in figure 2)

Since Yaw2 value in the position 2 equal the current Yaw value we can generalize the relationship as

\[ m = \text{robot initial orientation} + \text{Yaw} + \text{robot last angle} \]  

where

**Yaw value**: is the Yaw current reading that taken from the MPU
Now we can estimate the (x,y) values as shown in table (3) by using Pythagorean theorem depending on distance and angle (Θ) values where.

\[ X = \text{distance} \times \cos\Theta \]  
\[ Y = \text{distance} \times \sin\Theta \]  

3.1 Programming section

Now we need to program the arduino microcontroller (in this prototype we use two arduino pieces) the first one (arduino Uno) to control robot movement and to connect with Wi-Fi shield for reading RSSI value, the second one (arduino nano) to connect with MPU 6050 for reading Yaw value, figure (4) shows Wi-Fi localization block diagram.

Hence we can control robot movement by varying the BWM of the two DC motors, change the pin mode to HIGH or LOW to control wheel action forward or backward, and you can put delay time for reading RSSI value as you like.
4. Experiments and results

Now we will present some of experiments where we will move the robot in different paths (circle path, rectangle path,
irregular path, corridor path) then we will compare between all of these paths, see figure (4).

**Figure 6:** circle path

**Figure 7:** rectangle path
Figure 8: irregular path

Figure 9: corridor path
As shown above the green line is the true path while the blue is the calculated results after filtering. Table (1) will show the mean square error and max error of all paths.

<table>
<thead>
<tr>
<th></th>
<th>Error</th>
<th>irregular path</th>
<th>corridor path</th>
<th>circle path</th>
<th>rectangle path</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE for calculated</td>
<td>22.21 CM</td>
<td>141.5 CM</td>
<td>54.8 CM</td>
<td>44.38 CM</td>
<td></td>
</tr>
<tr>
<td>MSE for filtered</td>
<td>15.55 CM</td>
<td>34.6 CM</td>
<td>31.88 CM</td>
<td>29.43 CM</td>
<td></td>
</tr>
<tr>
<td>MAX Error in calculated path</td>
<td>44 CM</td>
<td>394 CM</td>
<td>92 CM</td>
<td>102 CM</td>
<td></td>
</tr>
<tr>
<td>MAX Error in filtered path</td>
<td>33 CM</td>
<td>83 CM</td>
<td>57 CM</td>
<td>67.32 CM</td>
<td></td>
</tr>
</tbody>
</table>

5. Conclusion and future work

In this paper we present a Wi-Fi technique that depend on one AP and to estimate robot localization, in compare with another Wi-Fi techniques like fingerprinting which depend on k number of APs or triangulation technique which depend on 3 APs, we try to introduce a technique with less infrastructure and more efficiency, the challenge was we were use a cheap component to design the robot, so we suffer from slipping and the effect of noise which lead to reduce resolution.

For future, we think that using more efficient component will serve the result, like using more precise robot platform and AP with far range (may be use AC1900 Smart Dual-Band Gigabit Wi-Fi router instead of 300 Mbps wireless router that we used it currently).

References


[17] Caius Suliman1, Cristina Cruceru1, Florin Moldoveanu. Mobile Robot Position Estimation Using the Kalman Filter. Scientific Bulletin of the Petru Maior University of Tirgu Mures Vol. 6 (XXIII), 2009 ISSN 1841-9267
