Indoor Navigation using WiFi signals

David Ivan Vilaseca
School of Electronic Engineering,
University of Buenos Aires
Buenos Aires, Argentina
vilaseca@gmail.com

Juan Ignacio Giribet
School of Electronic Engineering,
University of Buenos Aires and
Argentine Institute of Mathematics - CONICET
Buenos Aires, Argentina
jgiribet@fi.uba.ar

***Abstract*—Every day we use location-dependant services, but there is still a lack of information in terms of a popular indoor location system. In this paper we show methods and some advantages of using already installed infraestructure such as WiFi Access Points to get navigation information. For these purpose an indoor navigation system was implemented using both inertial sensors and WiFi signal strength information. The whole system architectute, both hardware and software, is designed with a modular aproach in order to make easy the test and benchmark of different algotithms and data fusion techniques. Basic algorithms have been implemented to validate the whole system using simulated data through a software-in-the-loop setup.**

***Index Terms*—Indoor Navigation, Integrated Navigation, WiFi RSSI.**

I. Introduction

Since ancient times, man tries to know his location in order to solve all kind of logistic and strategic problems. One of the most disrupting improvements in the last decades was the Global Navigation Satellite System, making devices location-aware. The next breakthrough is to take this kind of service (which is now a comodity) to indoor locations where satellite signals are not available.

For that purpose we introduce some data fusion techniques using both inertial measurements and WiFi signal information used to get information about position, velocity and attitude of the navigating body.

Our approach includes supervised classification methods, system dynamics propagation and non-linear extensions of Kalman filtering, more precisely EKF.

Merging inertial measurements with WiFi signals has already been studied in [3], [4], [6]–[8], [13] among others. It is not the intention of this work to focus on the algorithms we developed to solve the navigation problem, but give a brief description of the hardware setup we used to evaluate these algorithms.

II. Integrated vs Inertial Navigation

There are two popular aproaches to consider when trying to get navigation solutions. The simplest one is to use inertial navigation (Fig 1). The core idea is to integrate inertial measurements such as specific force and angular velocity to get position, velocity and attitude information. This technique was originally used in battleships to follow the routes as planned, using big and expensive mechanical inertial measurements units (IMU) [10].

In the past decades, huge improvements were made in MEMS sensors, making possible to manufacture smaller and cheaper IMUs. This technology made possible to use inertial navigation techniques in consumer electronic products (see for instance [2], [12]). The system, in our case, is defined by

$$\dot{p} = v \tag{1}$$

$$\dot{v} = C_b^i f^b + g \tag{2}$$

$$\dot{C_b^i} = C_b^i S(\omega^b) \tag{3}$$

With initial conditions $p(0) = p_0, v(0) = v_0, C_b^i(0) = C_0$. ($\dot{x}$ stands for the time derivative of x)

Where, for a given time t, $p(t) = \left[\begin{array}{ccc} x & y & z \end{array} \right]^T \in \mathbb{R}^3$ is the body position respect to an inertial frame (i), $v(t) = \left[\begin{array}{ccc} v_x & v_y & v_z \end{array} \right]^T \in \mathbb{R}^3$ it is velocity, and $C_b^i(t) \in \mathbb{R}^{3\times 3}$ is the rotation matrix from the body frame (b) to the inertial frame (i). Magnitude $f^b(t) \in \mathbb{R}^3$ is the specific force measured by the accelerometer and $\omega^b(t) \in \mathbb{R}^3$ is the angular velocity measured by the gyroscope. Since accelerometers and gyroscope are attached to the body, its measurements are given in the body frame. The specific force is a combination of gravitational an inertial acceleration, and for that reason the navigator has to take account of the gravity force $g \in \mathbb{R}^3$ (assumed constant at indoor navigation environments).

The main drawback with inertial navigation is the cumulative error generated because of the integrated measurements (see Eqs 1, 2, 3). Both, accelerometer and gyroscope outputs are contaminated with zero mean white noise. To get angular position integrating angular velocity a linear random walk is added to the result, and to get position integrating twice acceleration, a quadratic error is added to the result. So, given the amount of noise at the sensors, the quality of the navigation solution get worse every second. This effect is even worst if other error sources are taken into account e.g., bias, scale factor, non-linearity, non-orthogonality, among others.

At this point it makes sense to introduce the integrated navigation model (Fig 2), where a new information source is added. In most applications this additional information comes from external sources (e.g., GPS, radar, sonar, magnetometer). Typically the new source is not good as the inertial one on the short term, but definitely better than the navigation solution of the INS for long periods of time.

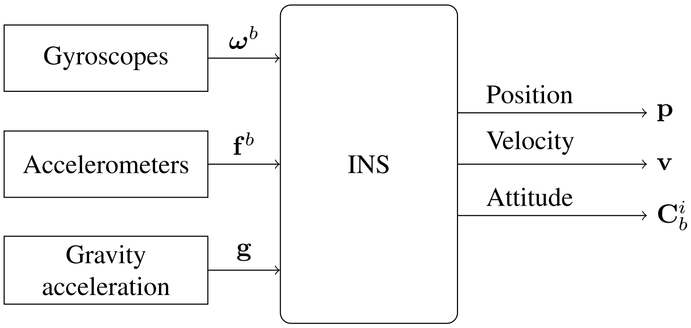

Fig. 1. Inertial Navigation System

Both information sources are complemented someway, in our case using an Extended Kalman Filter (EKF), which is the most popular fusion algorithm used in navigation. However, the proposed concept applies equally well to other recently developed methods ([1], [5], [9], [11]) which may compete advantageously with the EKF in certain applications.

In this work, the external data (the observation) is provided by a WiFi sensor giving position information (see III-D). This position is compared to the result of the INS and fed to the EKF in order to compute the p, v and C_b^i errors and enhance the navigation solution.

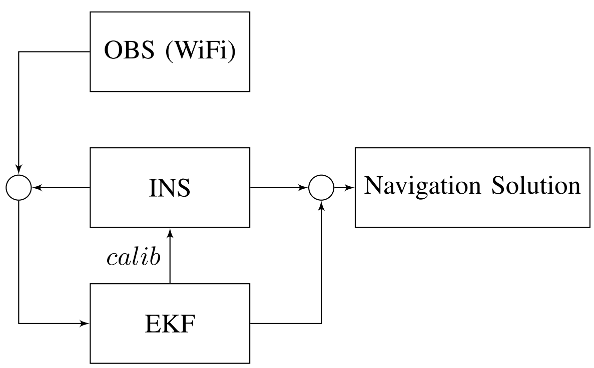

Fig. 2. Integrated Navigation System

Also, there are several sensor parameters such as bias and scale factors which can be estimated by the EKF and perform an online calibration of the INS block (see Fig. 2).

III. WiFi Signals

A. Why to use WiFi

Using a system such as WiFi (IEEE802.11) for any purpose different than communications is not a straightforward path. And there are several issues to solve in order to get things done in such scenarios. This is a fact and it is important to think about it from the start point.

Our approach of designing a WiFi-based navigation system is based on the popularity of WiFi devices and access points (AP). This is very important in terms of availability because, like GPS, the goal is to have as much coverture as possible.

Any reference-based navigation system requires some infrastructure to send or retransmit signals. Almost every building and indoor place in urban environments has one or more WiFi AP already installed, solving the infrastructure problem. Other advantage of using WiFi is the high availability and low cost of hardware. But the tradeoff is to use a system which is not designed for navigation purposes. In other words, we get a low-cost, high-availability system paying in terms of final accuracy, but if the result is good enough to make the system work, then we are in a sweet point.

B. Nature of Beacons

IEEE802.11 implements a periodic beacon which is transmitted by any AP who wants to be discoverable. This beacon is available regardless the network is encrypted or password protected, and there is no need to be connected to certain AP to hear the beacon.

At the receiver side the Rx power (RSSI) may be calculated. There is no standard way to do this and the units and scales used depends on the driver vendor.

The default value for beacon rate in most COTS[1] devices is 10Hz, which is a appropriate operating point for low-speed dynamic navigation.

C. Measuring Beacons

There are several Linux applications to get RSSI measurements but most of them are designed at user interface level. The use for that kind of applications is to help choosing a network from a list.

We found three problems trying to implement the navigator with this kind of applications

1) There are a lot of abstraction levels between the measurement request and the actual measurement. Since the timing is not important for network-choosing applications, not always a measurement request triggers a real measurement and the result is some cache information about the available networks.
2) Because of the previous item, it is very difficult to obtain periodic measurements.
3) And those measurements are not fast enough to get good navigation performance.

With this in mind and after several tests, we opted for modify the program `airodump-ng` from the Aircrack package. After a small refactor we get rid of the user interface and print the data in a convenient CSV format, and we also added some IPC between the navigator program and the modified `airodump-ng` version to trigger the measurements and avoid calling the program each time. This detail saves a lot of overhead time accessing the network card.

D. Beacon Processing

We are using a supervised classification method (Bayesian filtering) to get the position estimation given the RSSI measurement.

[1]Commercial Of-The-Shelf

As any supervised classification algorithm, there are two stages involved. The first one (training) consists on creating a database of RSSI histograms for known locations in space. For each point defined in the space we get statistical information about the RSSI received at this point, from all the AP available.

With all this information, the a priori database is created for later use.

In the second stage (classification), a RSSI measurement is made and the a posteriori probability of location is maximized using the database from the training stage.

So the position estimation result using this method is a probability density function of the position points defined, with higher values on the most probable location points.

This information is used by the EKF to enhance the INS navigation solution.

IV. Hardware Architecture

A. Overview

The navigator has four main hardware blocks: an Inertial Measurement Unit (IMU), a WiFi USB dongle, an On Board Computer (OBC) and a Low Level Computer (LLC).

The IMU and WiFi dongle are the sensor part, allowing the system to measure 3-axis acceleration, 3-axis angular velocity and RSSI over WiFi channels.

The OBC and LLC are processing units, each one with a different task and architecture making simpler the development and testing.

B. OBC

The OBC must be able to solve the navigation algorithm. This includes propagating dynamic models, solving an Extended Kalman filter (i.e. calculating matrix inverses) and also getting and processing the WiFi navigation data (in our case we are using a Bayesian filter which handles several database files with the a priori information).

We choose a BeagleBoard C4 board (Fig 3) as the OBC because it is capable of solving all the problems mentioned above and also has a important user community, which simplifies the developing and debugging process when problems are found on the road.

The BeagleBoard has a OMAP3530DCBB72 running at 720MHz, with a Linux as operating system.

C. LLC

The function of this computer is to make some basic processing to the raw IMU data, and handle this information, insert a timestamp and send it to the OBC. The important part of this unit is that runs a Real Time Operating System (RTOS) being able to solve time critic tasks.

It is reasonable for future implementations of this system to include other low-level sensors such as magnetometers, ultrasonic sensors, barometers, among others. In this scenario is very useful to have available some interface with easy access to low level peripherals.

The LLC is based on a 32 bits LPC1769 with Cortex-M3 core running FreeRTOS (Fig 4).

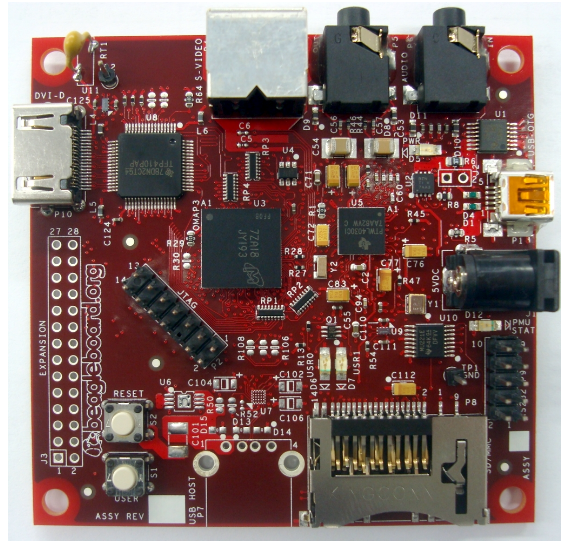

Fig. 3. OBC

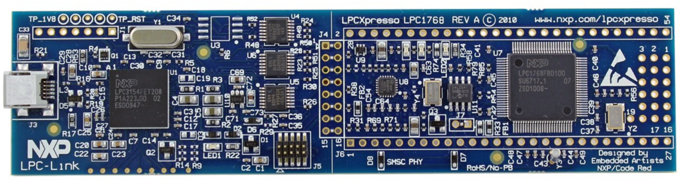

Fig. 4. LLC

D. IMU

The IMU (Fig 5) is based on MEMS sensors: 3-axis MMA7361L accelerometer from Freescale, 2-axis LPR510AL gyroscope (pitch, roll) and 1-axiz LY510ALH gyroscope (yaw) from STMicroelectronics.

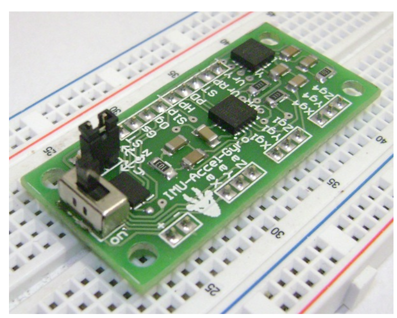

Fig. 5. IMU

All the sensors has analog interface, where the output is a voltage proportional to the quantity measured. The IMU data is sampled by the LLC ADC is at a periodic interval thanks to the RTOS driving the LLC.

E. WiFi

The WiFi interface selected is a TL-WN321G from TP-LINK (Fig 6). It has USB interface and the chipset is fully compatible with Linux operating systems. The need of driving this kind of devices is one of the main reasons for splitting the processing units.

Fig. 6. WiFi dongle

V. Data Fusion

With data from inertial and WiFi sensors available the question is what to do with all this information.

In a high level description, the IMU feeds a dynamic propagator which calculates the position, velocity and attitude given the inertial measurements and initial conditions. Because of the integration nature of the algorithm, the final result has an error added, as mentioned at Section II. This result makes impossible to navigate for long time periods trusting only in inertial measurements.

On the other hand, the RSSI measurements from the WiFi device feed a Bayesian filter which estimates a coarse position given the actual measurement and the a priori database with information about the RSSI distribution over space. The accuracy of this kind of sensor is not as good as the inertial counterpart at short periods of time, but considering the drift of the inertial part, there is a crossover point somewhere in time where the WiFi estimation is better than the inertial one. Furthermore, based on WiFi measurement it is possible to estimate the position but not the attitude.

Our approach under this circumstances is to use both measurements, taking advantage of each one when is convenient. For this purpose an Extended Kalman Filter have been implemented to take into account the measurement quality of each sensor (in statistical terms) and to give a solution as a weighted average of each data source.

The EKF has two steps for computing the navigation solution. The first step is the actualization step where the system dynamics are propagated, and the second step is the observation step, where the new information is fed into the filter. We developed the EKF to compute the INS error instead of the navigation variables themselves to keep a more-linear environment. (i.e. the error dynamics are less no linear than the navigation variables).

At the observation step there are several ways to incorporate the new information to the filter, considering the probabilistic nature of the WiFi measurement.

The first approach is to take the most-probable location as the true location, using some convenient covariance attached to it. But the output of the Bayesian filter have more information allowing more complex schemes. For example, it is possible to take the most probable location with a small covariance and the second most probable location with a bigger one making two observations instead of one. A not recommended approach to use the Bayesian filter information is to compute the baricenter of the probability density function before feeding the information into the EKF, because of the complex power distribution over space of the electromagnetic signals at 2.4GHz. Other possible way to improve the interaction between the EKF and the Bayesian filter is to use the covariance of the state vector of the first one to discard outliers generated by the second one. Our approach for this work consider the simplest scheme mentioned here but it is capable to be modified to introduce any of the complex methods.

VI. Testing and Profiling

Being able to test and verify the embedded systems is not a trivial task, and a good testing environment may be the difference between finding or not specific problems.

When the device do not have any user interface and it is moving around somewhere else where the workstation is located, the testing environment turns to be more necessary. For that reason our design includes a data logger / streamer via TCP/IP and a graphical control panel to generate reports and simplify debugging and profiling the system.

A. TCP data-logger

The TCP data-logger is part of the OBC software, and it is capable of logging data on a local filesystem for later analysis, or stream the data via TCP socket in real time. Several logs/streams may be used at the same time for different variables, and each channel is executed in a different thread to avoid creating locks in the main program flow.

This kind of tool is extremely versatile, and at different stages of the development was connected to MATLAB, LabView, `nc` (Bash).

B. Control panel

The control panel is based on LabView to simplify the front end and the GUI. All the state variables such as position, velocity, attitude, and sensor calibration data are plotted in realtime. This panel runs as a client connecting directly to the TCP logger running on the navigation body (server side).

VII. Results

To test the whole system under laboratory conditions we implemented a software-in-the-loop environment where all the input data (inertial measurements and WiFi data) is synthesized. In this scenario we have the ground truth data to compare the results and benchmark the different algorithms.

We ran tests on a 10m x 10m simulated WiFi grid (i.e. this is the probability space defined by the bayesian filter). Our IMU measurements rate was 100Hz, and the WiFi measurement rate 1Hz. The result of this simulation is consistent with real expermental data used to validate the model.

In the following figures is possible to compare the results of the navigation algorithm using only data from the IMU (Inertial Navigation - Fig 8) vs the navigation algorithm using both IMU and WiFi data (Integrated Navigation - Fig 9).

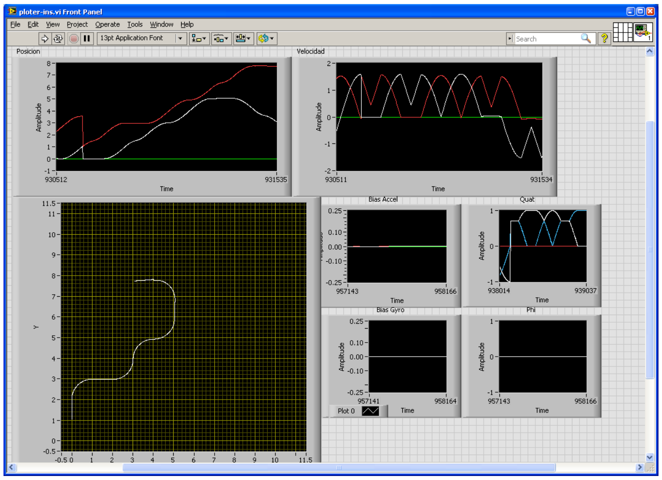

Fig. 7. Control Panel

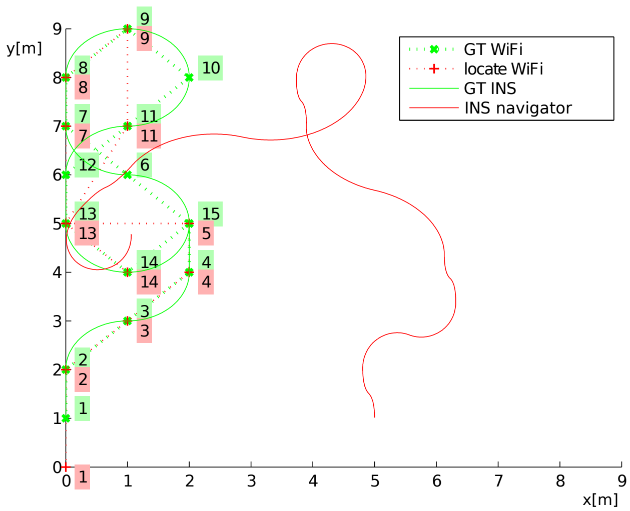

Fig. 8. Inertial Navigation Only

In both experiments, the initial conditions are misgiven with a 5m error on the horizontal position. The red path is the result of the algorithm, and the green one is the ground truth.

Numbered labels indicates the external measurements (in this case, WiFi bayesian filter location). The green labels show the truth position and the read labels indicates the result of the bayesian filter.

There are two important differences to note. First: after some settling time, the integrated navigation error is bounded meanwhile the inertial navigation is not. Second: the error due to wrong initial conditions is compensated in the integrated navigation case thanks to external observations, but in the inertial navigation there is no way to know this kind of errors.

VIII. Conclusions and Future Work

WiFi signals are not designed for navigation. But valuable information can be extracted from them and use it to generate position information. This is convenient because GPS availability is poor at indoor environments and WiFi AP coverage

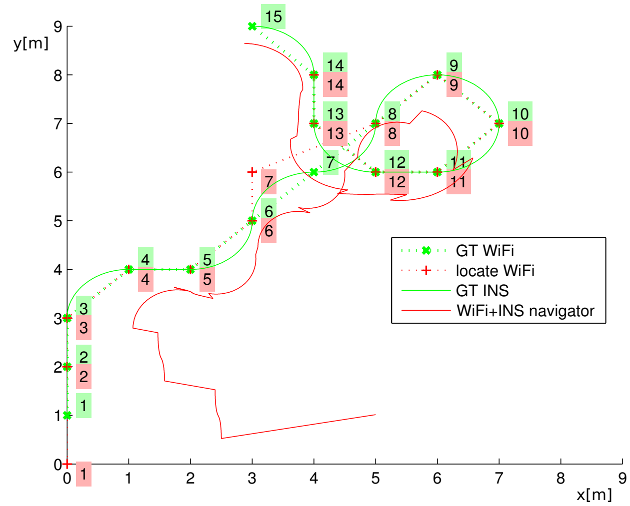

Fig. 9. Integrated Navigation.

is good. All the WiFi infrastructure already installed is worth to be used for this purpose.

We obtained errors from 2m to 4m on position, depending on the conditions. This are typical values for state-of-the-art WiFi location systems, and are good enough for personal navigations, surveillance and tracking of valuable objects inside buildings, etc.

Good results in terms of position definition are obtained using no less than 3 routers. But using more than 5 do not generate major improvements.

One of the drawbacks of the whole setup is the two step learning method for the Bayesian filter. Making this process automatic and during the navigation would be a huge advance. For that purpose, SLAM techniques may be used, keeping the a priori database updated.

Non isotropy of the WiFi antenna receptor should be used to extract features about orientation in the Bayesian filter scheme. This is important, because in a no-movement situation, the orientation can't be observed from external references.

Using a SDR front end would allow to get more information than the beacon and the RSSI measurement. With this kind of devices, the spectrum of features that can be used is far wider than using a 802.11 driver.

References

[1] N. Gordon Arulampalam M. S., S. Maskell and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Sign. Proc. Vol. 50, No. 2*, pages 646 – 659, 2002.

[2] N. El Sheimy, EW Shin, and X. Niu. Kalman filter face-off: Extended vs. unscented kalman filters for integrated gps and mems inertial. In *Proceedings of the International Symposium on Global Navigation Satellite Systems (GNSS'06)*, pages 48–54, 2006.

[3] F. Evennou and F. Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *EURASIP journal on applied signal processing*, 2006:164–164, 2006.

[4] K. Frank, B. Krach, N. Catterall, and P. Robertson. Development and evaluation of a combined wlan & inertial indoor pedestrian positioning system. In *ION GNSS*, 2009.

[5] J Llinas Hall David L. *Handbook of multisensor data fusion*. Number v. 1 in AIAA education series. USA, CRC Press, 2001.

[6] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, 2007.

[7] Anindya S. Paul and Eric A. Wan. Wi-fi based indoor localization and tracking using sigma-point kalman filtering methods. *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 646 – 659, 2008.

[8] L. Pei, R. Chen, J. Liu, H. Kuusniemi, T. Tenhunen, and Y. Chen. Using inquiry-based bluetooth rssi probability distributions for indoor positioning. *Journal of Global Positioning Systems*, 9(2):122–130, 2010.

[9] E.H. Shin and N. El-Sheimy. An unscented kalman filter for in-motion alignment of low-cost imus. In *Position Location and Navigation Symposium, 2004. PLANS 2004*, pages 273–279. IEEE, 2004.

[10] D.H. Titterton and J.L. Weston. *Strapdown inertial navigation technology*, volume 17. Peter Peregrinus Ltd, 2004.

[11] R. Van Der Merwe and E.A. Wan. Sigma-point kalman filters for integrated navigation. In *Proceedings of the 60th Annual Meeting of the Institute of Navigation (ION)*, pages 641–654, 2004.

[12] J.H. Wang. *Intelligent MEMS INS/GPS integration for land vehicle navigation*, volume 67. University Microfilms International, P. O. Box 1764, Ann Arbor, MI, 48106, USA, 2006.

[13] GV Zaruba, M. Huber, FA Kamangar, and I. Chlamtac. Indoor location tracking using rssi readings from a single wi-fi access point. *Wireless networks*, 13(2):221–235, 2007.