Mobile Sensor Network Localization in Harsh Environments

Harsha Chenji and Radu Stoleru

Texas A&M University, College Station TX, USA
{cjh,stoleru}@cse.tamu.edu

Abstract. The node localization problem in mobile sensor networks has recently received significant attention. Particle filters, adapted from robotics, have produced good localization accuracies in conventional settings, but suffer significantly when used in challenging indoor and mobile environments characterized by a high degree of radio irregularity. We propose FuzLoc, a fuzzy logic-based approach for mobile node localization in challenging environments and formulate the localization problem as a fuzzy multilateration problem, with a fuzzy grid-prediction scheme for sparse networks. We demonstrate the performance and feasibility of our localization scheme through extensive simulations and a proof-of-concept implementation on hardware, respectively. Simulation results augmented by data gathered from our 42 node indoor testbed demonstrate improvements in the localization accuracy from 20%-40% when the radio irregularity is high.

1 Introduction

Wireless sensor networks are increasingly a part of the modern landscape. Disciplines as diverse as volcanic eruption prediction [1] and disaster response [2] benefit from the addition of sensing and networking. One common requirement of many wireless sensor network (WSN) systems is localization, where deployed nodes in a network endeavor to discover their positions. The precise location of a pre-eruption tremor or of a patient in distress are two compelling examples of the need for accurate localization.

In some cases, localization is simple. For smaller networks covering small areas, fixed gateway devices and one-hop communications provide enough resolution. Larger networks may be provisioned with location information at the time of deployment [3]. GPS is a viable option for small outdoor deployments where cost and the power budget permit. However, in many common environments, localization is very difficult. GPS-based localization is not viable when the GPS receiver can't see the satellites. Signal strength-based solutions fail when there is a high degree of RF multi-path or interference, like most indoor and urban environments. Mobility in these harsh environments further complicates the problem.

Some solutions for localization in harsh, mobile environments assume availability of high-precision clocks and specialized hardware. [4] and [5] rely on accurate measurement of TDOA and distance traveled. Radio interferometry localizes

R. Rajaraman et al. (Eds.): DCOSS 2010, LNCS 6131, pp. 244–257, 2010.

nodes to within centimeters in [6] in non-multipath environments. All of these solutions rely on stable environments with low multi-path where a measured or sensed range reliably predicts the actual distance between two nodes.

Harsh, mobile environments, however, fail to meet these assumptions. The RSS-distance relationship [7] is inconstant and connectivity may vary dramatically. Assuming a predictable relationship between distance and RSS is problematic due to errors induced by multi-path and fading. Connectivity information gathered by the nodes could actually be misinformation. In many cases, compounding of errors may occur if the localization method relies on previous location estimations.

Fuzzy logic offers an inexpensive and robust way to deal with highly variable RSS measurements made in noisy, uncertain environments. Empirical measurements are used to produce rules that the fuzzy inference system uses to interpret input. The output of this process recovers the actual value compensated for variability in the local environment. We employ this basic technique in two constituent subsystems of FuzLoc - the Fuzzy Non Linear System (FNLS) and the Fuzzy Grid Prediction System (FGPS).

Our contributions include: i) a fuzzy logic-based method of building rule sets that characterizes the local signal environment, ii) a fuzzy non-linear system construct that uses these rules to convert extremely noisy RSS measurements to locations using a mechanism that compensates for inherent error, iii) a fuzzy grid prediction system that optimizes localization under conditions of low connectivity and low anchor density, iv) extensive simulations and comparisons to state of the art algorithms and v) a proof-of-concept implementation on motes.

The paper is organized as follows. Section 2 motivates our work and Section 3 introduces the fuzzy logic-based framework. Section 4 evaluates the proposed localization technique, followed by Conclusions in Section 5.

2 Motivation and Background

Several authors [8,9,10,11] have proposed Monte Carlo-based techniques, frequently used in robotics, for localization in mobile sensor networks. These localization techniques assume that a subset of nodes, called anchors, know their location. Nodes and anchors move randomly in the deployment area. Maximum velocity of a node is bounded but the actual velocity is unknown to nodes or anchors. Anchors periodically broadcast their locations.

This paper is motivated by our interest in a localization technique for a mobile sensor network, deployed in a harsh environment and the set of interesting/ surprising results obtained from simulations of three state of the art localization techniques for mobile sensor networks, namely MCL [8], MSL [9] and OTMCL [11]. Using the simulators developed by the authors of [8,9,11], a scenario assuming highly irregular radio ranges was developed, typical of harsh indoor or extremely obstructed outdoor environments. The irregularity in the radio range is modeled in these simulators as a degree of irregularity (DoI) parameter [8]. The DoI represents the maximum radio range variation per unit degree change in direction.

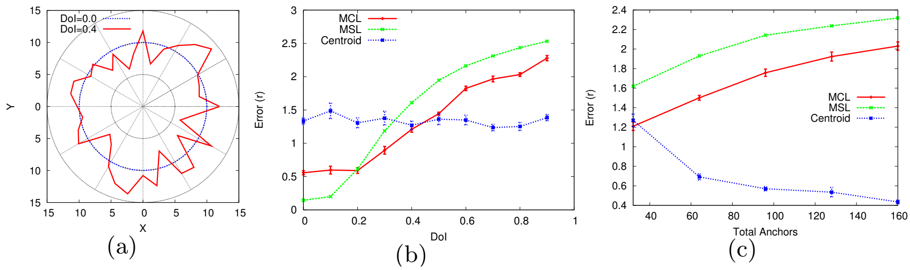


Fig. 1. (a) Radio patterns for two different degrees of radio irregularity (DoI); (b) the effect of DoI on localization error in MSL, MCL and Centroid; and (c) the effect of anchor density on localization error, at DoI=0.4, for MSL, MCL and Centroid

An example is depicted in Figure 1(a). When DoI=0.4 the actual communication range is randomly chosen from [0.6r, 1.4r].

Simulation results, for a network of 320 nodes, 32 anchors deployed in a 500×500 moving at 0.2r (r, the radio range) are shown in Figures 1(b) and 1(c). Figure 1(b) demonstrates that the DoI parameter has a significant negative effect on the localization accuracy. At DoI=0, both MCL, MSL and OTMCL achieve localization errors of 0.2r and 0.5r. With an increase in the DoI to 0.4, their localization error increases 400%. More surprisingly, as depicted in Figure 1(c), at a high DoI value, an increase in the number of anchors has a detrimental effect on localization accuracy. This result is counter-intuitive since access to more anchors implies that nodes have more opportunities to receive accurate location information, as exemplified by the performance of Centroid [12], in the same figure. A similar observation is made in [10] although no further study was performed. Our results suggest that samples get successively polluted with time, since the nodes used for filtering the samples may not be actual neighbors. The number of polluted samples increases with increasing anchor density.

The challenges identified above were partially addressed in recent work in sensor network node localization [13] that makes use of variables typical of range-based localization techniques (e.g., RSSI) to improve the accuracy of range-free techniques. In a similar vein, we propose to formulate the localization problem as a fuzzy inference problem using RSSI in a fuzzy logic-based localization system where the concept of distances used are very loose, such as “High”, “Medium” or “Low”.

2.1 Related Work

Existing work can be classified as range-based or range-free although a few techniques do not fall cleanly into these categories.

Range-based Localization Methods: These methods require an estimate of the distance or angle between two nodes to localize and may operative with both absolute and relative coordinate systems. A frequent requirement is the presence of at least three anchors so that necessary uniqueness and geometric constraints are satisfied. GPS is a familiar range-based method that uses the time of arrival of signals from satellites to obtain a precise location in latitude-longitude format. Some methods use surveying to predetermine RSSI values at any point in the area of deployment. Many solutions use time difference of arrival (TDoA) [4] [5]. For all of these methods, typical drawbacks include additional hardware, higher computational loads, increased node size, higher energy consumption and increased cost. A lighter weight solution leverages existing cellular telephony networks. Fuzzy logic is used to locate cellular phones in a hexagonal grid in a cellular network [14]. It assumes a fixed number of anchors but handles mobility very well. The computation and refining are not suitable for resource-constrained embedded sensors.

Range-free Localization Methods: Hop counting is a technique frequently used in these scenarios. A major drawback for hop-counting is that it fails in networks with irregular topologies such as those with a concave shape. Mobility incurs large overhead since all the hop counts will have to be refreshed frequently. APIT is a similar method which divides the area of deployment into triangles formed by anchors and then estimates the location. It assumes a large anchor density and higher radio ranges for the anchor nodes. An advantage is, however, that extremely low computational resources are needed. Both methods introduce large errors for ad-hoc networks. A hybrid method [13] that uses RSS and connectivity is worthy of note.

2.2 Background

Fuzzy logic has been applied in robot localization [15,16], because it reworks classical set theory and enables it to have non-rigid, or fuzzy, set boundaries. A *fuzzy set*, sometimes called *fuzzy bin*, is defined by an associated function $\mu(x)$, which describes the degree of membership of a crisp number in the set. A *crisp number* can belong to more than one fuzzy set at a given time, with varying degrees of membership. A *fuzzy number* is a special fuzzy bin where the membership is 1 at one and only one point. A fuzzy number represents a multi-valued, imprecise quantity unlike a single valued traditional number. One popular $\mu(x)$ function, depicted in Figure 2(a), is the *triangular membership function*:

$$\mu(x) = \begin{cases} 0 & \text{if } x < a \\ (x-a)/(b-a) & \text{if } a \leq x \leq b \\ (c-x)/(c-b) & \text{if } b \leq x \leq c \\ 0 & \text{if } x > c \end{cases} \tag{1}$$

where (a, b, c) defines a triangular bin. As shown, the $WEAK$ fuzzy set can be represented as (-90,-70,-50) and $MEDIUM$ as (-70,-50,-30). A crisp number, RSSI = -55dBm has a membership of 0.25 in $WEAK$ and 0.75 in $MEDIUM$.

A *fuzzy system* is defined by a set of fuzzy rules which relate linguistic variables in the form of an IF-THEN clause. Typically the IF clause contains the input linguistic variable (e.g., RSSI) and the THEN clause contains the output linguistic variable (e.g., DISTANCE).

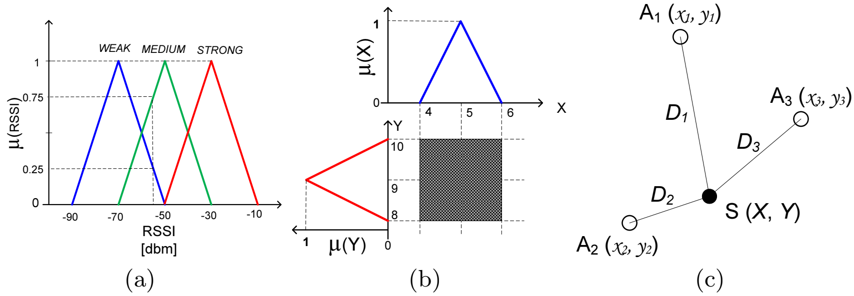


Fig. 2. (a) A triangular membership function $\mu_{(RSSI)}$ consisting of three bins. A fuzzified RSSI value of -55 pertains to two bins, $WEAK$ and $MEDIUM$, with degrees of membership of 0.25 and 0.75, respectively; (b) Representation of a fuzzy location, using two triangular membership functions; and (c) a sensor node S with fuzzy coordinates X and Y, to be located using three anchors positioned at (x_1, y_1), (x_2, y_2) and (x_3, y_3).

3 A Fuzzy Logic-Based Node Localization Framework

The two dimensional location of a node can be represented as a pair (X, Y), where both X and Y are fuzzy numbers, as depicted in Figure 2(b). This section develops the theoretical foundation behind the computation of this fuzzy location, using imprecise and noisy RSSI measurements labeled as "HIGH", "MEDIUM" or "LOW".

3.1 Fuzzy Non Linear System (FNLS)

As depicted in Figure 2(c), consider a node s about to be localized, in the vicinity of three anchor nodes A_j $(j = \overline{1, 3})$. Each anchor node is equipped with a set of fuzzy rules that map fuzzy RSSI values to fuzzy distance values:

Rule i: **IF** RSSI is $RSSI_i$ **THEN** DIST is $Dist_i$

where $RSSI_i$ and $Dist_i$ are fuzzy linguistic variables (e.g. LOW, $MEDIUM$, $HIGH$) and "is" means "is a member of".

For a more general case, when the node S is within radio range of n anchors, the node localization problem can be formulated as a fuzzy multilateration problem. The following:

$$\begin{aligned} F_1 &= (X - x_1)^2 + (Y - y_1)^2 - D_1^2 = 0 \\ F_2 &= (X - x_2)^2 + (Y - y_2)^2 - D_2^2 = 0 \\ &\dots \\ F_n &= (X - x_n)^2 + (Y - y_n)^2 - D_n^2 = 0 \end{aligned} \tag{2}$$

defines a non-linear system of equations describing the relation between the locations of the nodes and anchors and the distances among them. The variables

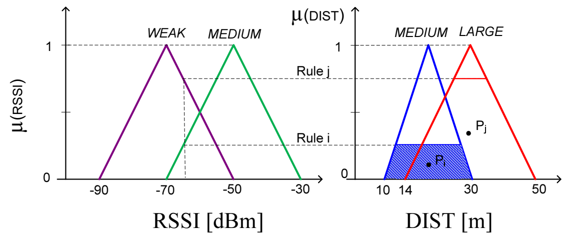


Fig. 3. The fuzzification process for an input RSSI value of -62dB. In this example, the fuzzy rule base maps this value through two rules: "Rule i: IF RSS is WEAK, THEN distance is LARGE" and "Rule j: IF RSS is MEDIUM, THEN distance is MEDIUM".

X, Y and D_k $(k = \overline{1,n})$ are fuzzy numbers, while (x_k, y_k) $(k = \overline{1,n})$ are crisp numbers. The objective is to minimize the mean square error over all equations.

FIS Subsystem. A definition of the process of obtaining the fuzzy distances D_k is needed before solving the system of equations. This process, called fuzzy inference, transforms a crisp RSSI value obtained from a packet sent by a node and received by an anchor into a fuzzy number, i.e., distance D_k between node and anchor. Figure 3 depicts an example for the fuzzifying process. As shown, an RSSI value of -62dBm has different membership values $\mu(RSSI)$ for the fuzzy bins $WEAK$ and $MEDIUM$. The two fuzzy bins, in this example, are mapped by a fuzzy rule base formed by two fuzzy rules: "Rule i" and "Rule j". These two fuzzy rules define the mapping from the RSSI fuzzy sets to the DIST fuzzy sets. As shown in Figure 3, the two fuzzy rules indicate the membership $\mu(DIST)$ in the distance domain. P_i and P_j indicate the center of gravity of the trapezoid formed by the mapping of the RSSI into fuzzy bins $MEDIUM$ and $LARGE$, respectively.

Typically, a single RSSI value matches multiple fuzzy rules. Let's assume that the fuzzy rule base maps an RSSI value to a set of m fuzzy DIST bins. The set of centers of gravity P_l $(l = \overline{1,m})$ is denoted by $P = \{P_1, P_2 \ldots P_m\}$ in Figure 3. The intuition for computing the output as a fuzzy number D_k derives from the center-average defuzzification method as follows: First, calculate the centroid of all points in P - call it P_c. Next, take the centroid of all the points in P whose abscissa is less than that of P_c i.e., $L = \{P_n | x(P_n) \leq x(P_c)\}$. Similarly, $G = \{P_n | x(P_n) \geq x(P_c)\}$ is the set of points whose abscissa is greater than that of P. The abscissae of three points P, L and G represent the resulting fuzzy distance D_k, formally described as:

$$D_k = (a, b, c) = \left(\left(\frac{\sum L_n}{|L|} \right)_x, (P_c)_x, \left(\frac{\sum G_n}{|G|} \right)_x \right) \quad (3)$$

In order to solve the non-linear system of Equations 2, in two fuzzy variables, the fuzzy variant of the iterative classical Newton method based on the Jacobian matrix [17,18] is used. To accomplish this, the fuzzy numbers are expressed in their parametric form $X = (\underline{X}, \overline{X})$ where $\underline{X}$ and $\overline{X}$ are continuous bounded non-decreasing and non-increasing, respectively, functions. These functions effectively represent the "left half" and "right half" of the membership function.

For a triangular membership function, such as defined in Equation 1 and depicted in Figure 2(a), a parametric representation in $r \in [0,1]$ is $X = (a + (b-a)r, c-(c-b)r)$. The system of Equations 2 are, therefore, represented in the parametric form. Without any loss of generality, assume that X and Y are positive. Then, each F_n in Equation 2 can be split into:

$$\begin{aligned} \underline{F_n} &= (\underline{X} - x_n)^2 + (\underline{Y} - y_n)^2 - \underline{D_n}^2 = 0 \\ \overline{F_n} &= (\overline{X} - x_n)^2 + (\overline{Y} - y_n)^2 - \overline{D_n}^2 = 0 \end{aligned} \tag{4}$$

The Jacobian J is constructed as:

$$J = \begin{bmatrix} \underline{F_1}_{\underline{X}} & \underline{F_1}_{\overline{X}} & \underline{F_1}_{\underline{Y}} & \underline{F_1}_{\overline{Y}} \\ \overline{F_1}_{\underline{X}} & \overline{F_1}_{\overline{X}} & \overline{F_1}_{\underline{Y}} & \overline{F_1}_{\overline{Y}} \\ \underline{F_2}_{\underline{X}} & \underline{F_2}_{\overline{X}} & \underline{F_2}_{\underline{Y}} & \underline{F_2}_{\overline{Y}} \\ \overline{F_2}_{\underline{X}} & \overline{F_2}_{\overline{X}} & \overline{F_2}_{\underline{Y}} & \overline{F_2}_{\overline{Y}} \\ \dots & \dots & \dots & \dots \\ \underline{F_n}_{\underline{X}} & \underline{F_n}_{\overline{X}} & \underline{F_n}_{\underline{Y}} & \underline{F_n}_{\overline{Y}} \\ \overline{F_n}_{\underline{X}} & \overline{F_n}_{\overline{X}} & \overline{F_n}_{\underline{Y}} & \overline{F_n}_{\overline{Y}} \end{bmatrix} = \begin{bmatrix} 2(\underline{X} - x_1) & 0 & 2(\underline{Y} - y_1) & 0 \\ 0 & 2(\overline{X} - x_1) & 0 & 2(\overline{Y} - y_1) \\ 2(\underline{X} - x_2) & 0 & 2(\underline{Y} - y_2) & 0 \\ 0 & 2(\overline{X} - x_2) & 0 & 2(\overline{Y} - y_2) \\ \dots & \dots & \dots & \dots \\ 2(\underline{X} - x_n) & 0 & 2(\underline{Y} - y_n) & 0 \\ 0 & 2(\overline{X} - x_n) & 0 & 2(\overline{Y} - y_n) \end{bmatrix} \tag{5}$$

The initial guess (X_0, Y_0) is computed from the average of the coordinates of the anchors. Define a matrix $F = [\underline{F_1}\ \overline{F_1}\ \underline{F_2}\ \overline{F_2}\ \dots\ \underline{F_n}\ \overline{F_n}]^T$. (X_0, Y_0) is updated in increments at every iteration by calculating an update matrix Δ using the matrices J and F evaluated at the current values of X and Y:

$$\Delta = \begin{bmatrix} \underline{h}(r) \\ \overline{h}(r) \\ \underline{k}(r) \\ \overline{k}(r) \end{bmatrix} = \begin{bmatrix} \underline{X}(r)^t - \underline{X}(r)^{t-1} \\ \overline{X}(r)^t - \overline{X}(r)^{t-1} \\ \underline{Y}(r)^t - \underline{Y}(r)^{t-1} \\ \overline{Y}(r)^t - \overline{Y}(r)^{t-1} \end{bmatrix} = -J^{-1}F \tag{6}$$

The process is repeated until Δ converges to 0 within ϵ.

3.2 Fuzzy Grid Prediction System (FGPS)

In mobile sensor networks with low anchor densities, it might frequently be the case that a node does not have enough anchors for multilateration. To address this problem we extend our fuzzy logic-based localization framework to predict an area, e.g., a cell in a grid, where the node might be. The idea is inspired from cellular systems [19]. We propose to virtualize the anchors, so that a node is within a set of Virtual Anchors at any point in time.

Consider the area in which the network is deployed to be subdivided into a grid of G cells, as depicted in Figure 3.2. We denote the probability that a node S is in a cell j ($j = 1 \dots G$) by p_j. To infer these probabilities, we construct a fuzzy system, whose input is the distance d_j between S and the center of cell j, and the output is a scalar $0 < p_j < 1$ for each j. The key idea is that the nearer a node is to the center of the cell, the more likely it is that the node can be found in that cell. A rule in our fuzzy system is as follows:

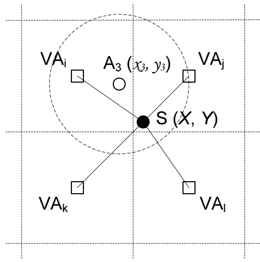


Fig. 4. A sensor S and the grid cells in its vicinity, is within radio range of anchor A_3

Rule i: **IF** (DIST_{grd1} is D_{i1}) and ... and (DIST_{grdG} is D_{iG}) **THEN** ($PROB_{grd1}$ is P_{i1}) and ... and ($PROB_{grdG}$ is P_{iG})

where D_{ij} is the fuzzy bin representing the distance between the node and the center of cell j, and P_{ij} is the fuzzy bin representing the probability that node S is in cell j.

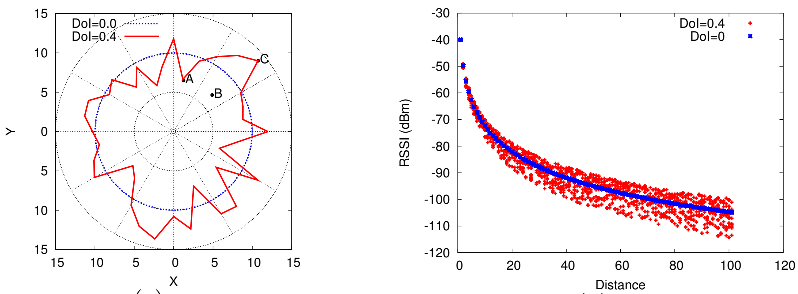
For each rule i, we calculate p_j by first fuzzifying d_j, applying it to the rule, and then defuzzifying the aggregate, as we described in Section 3.1. Once the most probable cell is found, the location of the node can be computed as the the intersection between this cell and the circle defined by the anchor.

It is paramount to remark that we can obtain p_j only if the node S has at least one anchor in its vicinity, i.e., we can estimate D_{ij}. The technique we propose for estimating D_{ij} is described in the Virtual Anchors section below. Before proceeding, we describe how to update p_j when no anchor is in the vicinity of node S. Since there is a high correlation between the current and previous cell a node is in, we construct a Recursive Least Squares [20] filter which predicts the cell in which the node S might be. For each cell j, we store m previous samples of p_j which serves as input to the filter.

Virtual Anchors. The fuzzy system requires that we calculate the distance from the node to the virtual anchor. We have to find the average distance instead, because we do not know the node's location. These average distances can be calculated only when at least one anchor is in the node's vicinity. The locus of the node around the anchor is a circle with radius as the radio range and center as the location of the anchor; the average distance to a virtual anchor can be easily calculated as the average of the distances between the virtual anchor and each point on this circle.

4 Performance Evaluation

Implementation. We implemented the FIS subsystem on EPIC motes running TinyOS 2.1 with an onboard CC2420 radio. The localization system was configured with 8 bins each for the distance and the RSSI, with ten preset rules. Upon receiving a message, the motes would read the RSSI from the radio and proceed to defuzzify it into a distance. This basic functionality was accomplished with



(a) (b)

Fig. 5. (a) The DoI model with three points of interest: although A and B are equally distant, their RSS values differ significantly in our EDoI model; and (b) RSSI vs. distance for the radio model used in the simulator, at DoI=0.4 and 0

285 lines of code, occupying 19,932B in ROM and 1,859B in RAM (including the code required to send and receive messages over the radio). The FIS subsystem results were similar within rounding errors to those of the simulator's, with identical rules and binning. To the best of our knowledge, no mobile wireless sensor testbeds are available for public use. For the sake of repeatability of results and because of lack of mobile testbed infrastructure, we evaluate the performance of our localization scheme through extensive simulations. We use the data gathered from our static 42 node indoor testbed, as shown in Figure 8(b), for validating the FIS subsystem performance.

Simulation. Our proposed fuzzy logic-based localization system was implemented as an extension of the simulator provided by the authors of [8]. For performance evaluation, we compare FuzLoc, our solution, with MCL, MSL, Centroid and a "Perfect FuzLoc" algorithm, which is fuzzy multilateration with a theoretical ~0% uncertainty in the ranges. These choices are justified by the fact that the we wanted to evaluate our solution against state of art Monte Carlo based solutions, as well as simpler techniques (e.g., Centroid), and demonstrate the advantages of our solution. We chose not to compare our solution against solutions that use additional hardware, such as OTMCL [11] which uses a compass.

Since our fuzzy logic-based localization technique makes use of the RSSI, we extended the DoI model [21]. In our EDoI model, the RSSI at the connectivity range predicted by the DoI model is associated with the minimum achievable received power at a receiver, i.e., receiver sensitivity. As shown in Figure 5(a), for points A and C (evaluated by the DoI model at the maximum radio ranges in two different directions), in our DoI model the RSSI is equal to the receiver sensitivity, -94dBm. For point B, we apply a log-normal fading model, such that the RSSI at point C (in the same direction as point B) is equal to receive sensitivity -94dBm. The predicted RSSI at point B is thus -60dBm. Formally, our EDoI model computes the RSSI, as follows:

$$RSSI(d) = S_i \frac{\log_{10} d}{\log_{10}[r(1 + DoI \times rand())]} \tag{7}$$

where S_i is the receiver sensitivity, r is the ideal radio range, DoI is the radio degree of irregularity and $rand$ is a random number $\mathcal{U}[0,1]$.

We simulate a set (N) of 320 sensor nodes deployed in a 500 × 500 area. Of the 320 nodes deployed, 32 nodes are designated anchors (set S). The radio range (r) of a node is 50 and the default DoI is 0.4. We chose these simulation parameters for consistency with results reported in [8], [9]. The default receiver sensitivity (S_i) is -94dBm, and a plot depicting the predicted RSSI by our EDoI model, is shown in Figure 5(b). The default maximum node velocity is to 0.2r. This velocity has been reported in [8] and confirmed in [11], to be optimal. We investigate the performance of all solutions for node velocities up to 0.5r. The results are averaged over all nodes over 10 runs, with each node taking 50 steps per run for a total of atleast 16000 trials per data point. The default setup uses 10 fuzzy triangular bins and the defuzzification method is center-average. The fuzzy location is defuzzified into a crisp location by considering only the center values of the abscissa and the ordinate.

Radio Irregularity. We performed simulations for different DoI values with all other parameters kept constant. Figure 6(a) depicts our results, indicating the deterioration in localization accuracy of MCL and MSL. The effect of compounded errors due to polluted samples (incorrectly computed locations which are used to compute the location in future iterations) has been investigated as the "kidnapped robot problem" [22] in robot localization. The kidnapped robot test verifies whether the localization algorithm is able to recover from localization failures, as signified by the sudden change in location due to "kidnapping". It has been shown that such uncorrected algorithms collapse when the observed sample is far from the estimated sample. MSL demonstrates an even more pronounced effect, since it also uses non-anchor neighbors for filtering, thus leading to even more pollution.

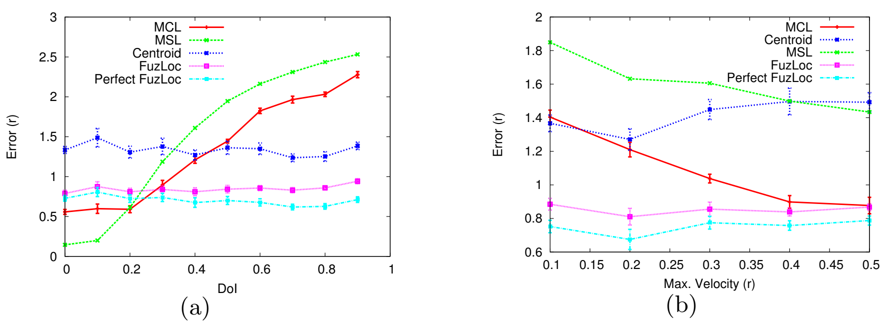


Fig. 6. Localization accuracy as affected by (a) DoI (N=320, S=32, v=0.2r); and (b) maximum node velocity (N=320, S=32, DoI=0.4)

Maximum Node Velocity. We investigate the effect of maximum node velocity on localization accuracy, for velocities up to 0.5r, a reasonably fast moving speed. The performance results are depicted in Figure 6(b). MCL and MSL assume that nodes know their maximum velocity. Hence, they use the velocity as a filtering condition, which improves their performance. Moreover, high velocity means having more anchors to filter against, leading to the freshening of samples at every instance. Figure 6(b) shows that MCL and MSL decrease their localization error from 1.4r to 0.9r, and 1.9r to 1.4r, respectively. Since Centroid and FuzLoc do not use the velocity, their performance is not expected to improve. Figure 6(b) indicates that their performance is not deteriorating.

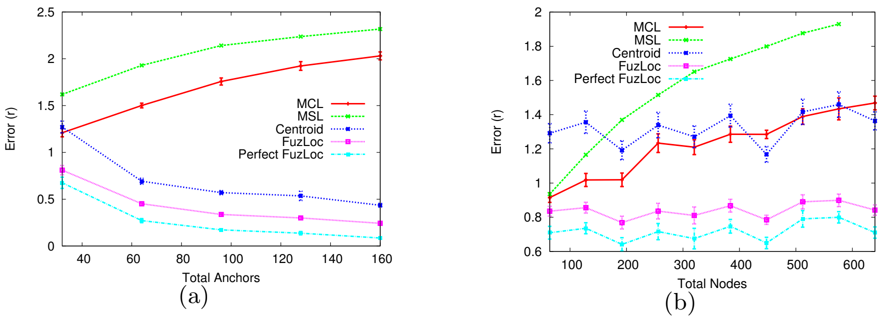


Fig. 7. Localization accuracy as affected by (a) anchor density at DoI=0.4 (N=320, v=0.2r); and (b) node density (S=32, v=0.2r, DoI=0.4)

Anchor Density. Anchor density is a critical parameter for anchor-based localization schemes. Figure 7(a) displays the impact of anchor density on the localization schemes where the number of anchors varies from 10% (32 anchors) to 50% (160 anchors), and the DoI is constant at 0.4. The accuracy of MCL and MSL deteriorates because an increase in anchor density is associated with an increase in the number of polluting sources. The mismatch of observed and actual radio ranges causes spurious anchors to appear as node's direct and indirect seeds. MSL considers non-anchor neighbors, hence it experiences higher pollution. Centroid performs better with increasing anchor density, as expected. FuzLoc also has a decrease in localization error, with a larger number of anchors. We observe that FuzLoc is not greatly affected by DoI and ranging errors.

Node Density. For this performance evaluation scenario we maintained the percentage of anchors fixed at 10%. As shown in Figure 7(b), the evaluated algorithms either suffer or are unaffected. None of the localization algorithms benefits from an increase in the node density. As shown, Centroid and FuzLoc are not substantially affected, except by the inherent randomness in simulation. MCL considers indirect seeds for sampling, hence a high node density means more anchors are misreported as indirect seeds. MSL considers non-anchor neighbors, hence at high node densities, it experiences a huge amount of sample pollution. While non-anchor neighbors help MSL to improve accuracy at low DoI, they become harmful at higher DoI values.

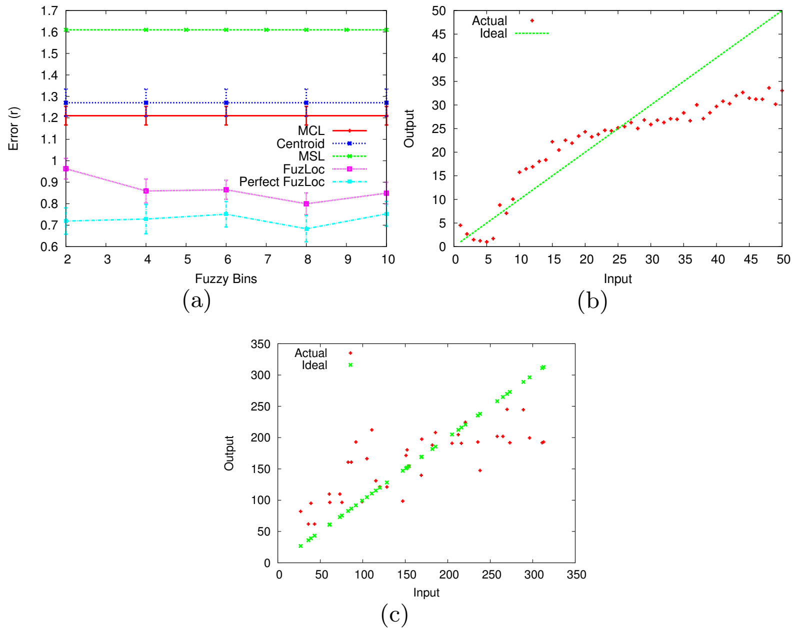


Fig. 8. (a) Localization accuracy as affected by the number of fuzzy bins (N=320, S=32, v=0.2r, DoI=0.4); (b) Performance of the FNLS FIS subsystem based on the simulated EDoI radio model; and (c) Performance based on real data gathered from our indoor testbed

Number of Bins. The number of bins in the fuzzy system is a design parameter - the greater the number of bins, the higher the accuracy of the system. Our evaluation of the influence of the number of bins is depicted in Figure 8(a). As shown, as the number of bins increases, the localization error of FuzLoc decreases. This is because more and more RSSs find a bin with high membership. The change in the number of bins, is expected to not affect MCL, MSL, Centroid, or even Perfect FuzLoc. Figure 8(a) shows that the aforementioned schemes remain invariant whereas FuzLoc experiences decreasing error with an increase in the number of bins.

FIS Performance. Real data gathered from our testbed (42 nodes deployed over a 600 sq.ft. indoor multipath environment) was used to evaluate the radio model and the FIS used in the simulator, by comparing the accuracy of the FIS subsystem as shown in Fig. 8(b) and Fig. 8(c). Figure 8(b) shows the performance of the FNLS FIS engine. Input distance on the X axis is translated into an RSS which is then defuzzified into a distance on the Y axis. Figure 8(c) is similar except that the RSS is extracted from the real dataset. After training the system with 30 random RSS-Distance pairs, RSS values deduced from distances were fed into the system so that a distance should be inferred. The straight line shows the ideal case. The FIS is somewhat efficient, except at the fringes, where the fuzzy system was not found to be trained due to the limited number of rules. With more rules, the system becomes more and more efficient. This result indicates the resiliency of FuzLoc to ranging errors.

Overhead. A typical FIS does not require significant storage capacity. If there were 8 bins, for example, a single byte could represent a bin. Hence, each FNLS rule requires just 2 bytes of storage. Typically, an anchor creates approximately 30 rules during the period of deployment which translates to 60 bytes of storage. The FGPS FIS however, requires 50 bytes for each rule (25 bins in the input, 25 in the output). Note that regular nodes do not store rules, only the anchors store rules. Moreover, due to the nature of the triangular bin shapes, simple calculations are required in order to (de)fuzzify. MCL requires at least 50 weighted samples for low localization error. Centroid does not store any history and thus has the smallest storage requirement. Amorphous stores announcements made by the anchors which are flooded throughout the network. If there are 320 nodes, 32 of which are anchors, MCL requires each node to store 50 samples - $(50 \times 4 \times 2 \times 320) = 128,000$B. Fuzzy on the other hand requires around 1,500 bytes for FGPS and around 60 for FNLS, which sums up to roughly 50% of the storage MCL requires. We note here that communication overhead is very similar among all evaluated localization techniques.

5 Conclusions

We have proposed FuzLoc, a fuzzy logic based localization method for harsh environments. The constituent systems use fuzzy multilateration and a grid predictor to compute the location as an area. Our method has been evaluated based on a variety of metrics. They prove that it is resistant to high DoI environments while providing a low localization error without any extra hardware. Only anchors need to have a slightly higher storage requirement. A deployment with more anchors at high DoI decreases the error. FuzLoc's principle limitation is that it does not work very well in static networks unless the anchors have pre-loaded fuzzy rules or the anchors have a large radio range. The distributed protocol requires some amount of data to be transmitted. This could be problematic in networks populated by resource-constrained nodes. Future work includes a module for iteratively redesigning the fuzzy bins to optimize based on network characteristics.

Acknowledgement. This work was funded in part by NSF grant CNS 0923203. The authors also wish to thank Mike George for his help.

References

1. Werner-Allen, G., Lorincz, K., Ruiz, M., Marcillo, O., Johnson, J., Lees, J., Welsh, M.: Deploying a wireless sensor network on an active volcano. IEEE Internet Computing 10(2) (March-April 2006)
2. George, S., Zhou, W., Chenji, H., Won, M., Lee, Y., Pazarloglou, A., Stoleru, R., Barooah, P.: A wireless ad hoc and sensor network architecture for situation management in disaster response. IEEE Communications Magazine (March 2010)

3. He, T., Krishnamurthy, S., Luo, L., Yan, T., Gu, L., Stoleru, R., Zhou, G., Cao, Q., Vicaire, P., Stankovic, J., Abdelzaher, T., Hui, J., Krogh, B.: VigilNet: an integrated sensor network system for energy-efficient surveillance. ACM Trans. Sens. Netw. 2(1) (2006)
4. Girod, L., Lukac, M., Trifa, V., Estrin, D.: A self-calibrating distributed acoustic sensing platform. In: SenSys (2006)
5. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: MobiCom (2000)
6. Kusy, B., Sallai, J., Balogh, G., Ledeczi, A., Protopopescu, V., Tolliver, J., DeNap, F., Parang, M.: Radio interferometric tracking of mobile wireless nodes. In: MobiSys (2007)
7. Whitehouse, K., Karlof, C., Culler, D.E.: A practical evaluation of radio signal strength for ranging-based localization. Mobile Computing and Communications Review 11(1) (2007)
8. Hu, L., Evans, D.: Localization for mobile sensor networks. In: MobiCom (2004)
9. Rudafshani, M., Datta, S.: Localization in wireless sensor networks. In: IPSN (2007)
10. Baggio, A., Langendoen, K.: Monte carlo localization for mobile wireless sensor networks. Ad Hoc Netw. 6(5) (2008)
11. Martins, M., Chen, H., Sezaki, K.: OTMCL: Orientation tracking-based monte carlo localization for mobile sensor networks. In: INSS (June 2009)
12. Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low cost outdoor localization for very small devices. IEEE Personal Communications Magazine (2000)
13. Zhong, Z., He, T.: Achieving range-free localization beyond connectivity. In: SenSys (2009)
14. Shen, X., Mark, J.W., Ye, J.: Mobile location estimation in CDMA cellular networks by using fuzzy logic. Wirel. Pers. Commun. 22(1) (2002)
15. LeBlanc, K., Saffiotti, A.: Multirobot object localization: A fuzzy fusion approach. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 39(5), 1259–1276 (2009)
16. Saffiotti, A.: The uses of fuzzy logic in autonomous robot navigation. Soft Computing-A Fusion of Foundations, Methodologies and Applications 1(4), 180–197 (1997)
17. Kelley, C.T.: Fundamentals of Algorithms - Solving Nonlinear Equations with Newton's Method. SIAM, Philadelphia (2003)
18. Shokri, J.: On systems of fuzzy nonlinear equations. Appl. Math. Sci, Ruse (2008)
19. Shen, X., Mark, J.W., Ye, J.: User mobility profile prediction: an adaptive fuzzy inference approach. Wirel. Netw. 6(5) (2000)
20. Hayes, M.: Statistical digital signal processing and modeling. Wiley India Pvt. Ltd, Chichester (2008)
21. He, T., Huang, C., Blum, B.M., Stankovic, J.A., Abdelzaher, T.: Range-free localization schemes for large scale sensor networks. In: MobiCom (2003)
22. Engelson, S., McDermott, D.: Error correction in mobile robotmap learning. In: ICRA (1992)