Side Localization to Increase Localization Accuracy

Walid M. Ibrahim
School of Computing
Queen's University
Canada
Email: walid@cs.queensu.ca

Najah A. Abu Ali
Faculty of Information Technology
UAE University
United Arab Emirates
najah@uaeu.ac.ae

Abd-Elhamid M. Taha
College of Engineering
Alfaisal University
Saudi Arabia
ataha@alfaisal.edu

Hossam S. Hassanein
School of Computing
Queen's University
Canada
hossam@cs.queensu.ca

***Abstract*—Estimating the location of sensor nodes in wireless sensor networks is a fundamental requirement in a variety of sensing applications. In large scale dense deployments where the area covered by sensor nodes is very large, it is impossible to localize all sensor nodes using single-hop localization techniques. A solution to this problem is to use a multi-hop localization technique to estimate sensor node positions. In some deployments it is required to maintain the anchor nodes at the edge of the simulated area. In previous work, we introduced a new localization scheme that uses distance measurements to localize sensor nodes using a collinear and non-collinear mobile anchor nodes placed at the edge of the sensed area. A Kalman Filter was then used to improve the location accuracy for each node. In this scheme each SN estimated its location from two independent directions then use such information to improve localization accuracy. In this paper, we extend the work to use side localization using hop measurements and fixed anchor node. We also compare the performance of using side localization for both hop and distance measurement. Through simulation we show that side localization using distance and hop measurements outperform DV-Hop and DV-Distance, which are mainstream localization protocols. The weighted mean hop measurement gives higher localization accuracy than using using distance measurement. However, if Kalman Filter is used distance measurement gives better localization accuracy.**

***Index Terms*—Localization, Positioning, Multihop, Collinearity, Flip Ambiguity, Kalman Filter.**

I. Introduction

In Wireless Sensor Network (WSN) monitoring applications, it is important to locate the position of an event to take the correct action. To localize Sensor Nodes (SNs) in WSN, anchor nodes broadcast their location with operating instruction to the SNs, and SNs use such information to estimate their positions. Most localization schemes require a high-density deployment of anchor nodes to ensure SNs have enough references to estimate their positions.

Depending on the application and size of the terrain, localization techniques can either be single-hop or multi-hop. In single-hop techniques, un-localized SNs require a minimum of three anchor nodes in 2-D and four anchor nodes in 3-D within their transmission range in order to estimate their locations using one of the distance measurements: Received Signal Strength Indicator (RSSI), Time of Arrival (ToA) or Angle of Arrival (AoA) [1].

However in a large scale deployment, the terrain where the SNs are deployed is so vast that not all the SNs can be located in the transmission range of three anchor nodes at the same time. Therefore it is impossible to localize all SNs using single-hop localization techniques. Thus a multi-hop localization technique is used to estimate the positions of SNs in large scale environments. Multi-hop localization techniques are classified into two major categories: *distance-based* and *connectivity-based*. Distance-based or range-based localization techniques rely on the individual inter-sensor distance data. Connectivity-based or range-free localization techniques depend mainly on the connectivity information (hop count) to estimate the locations of the un-localized SNs.

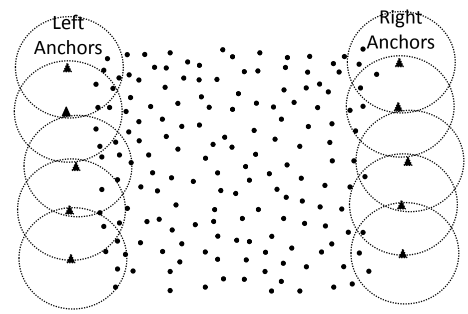

Fig. 1: WSN with Collinear Anchor Nodes.

In many scenarios the sensed area is located in an isolated terrain (i.e., dense rain forest or rocky terrain). SNs are usually randomly deployed into isolated terrain to collect information about the environment and send such information to the SNs at the edge. In such isolated environment especially in rain forest the GPS signal is distorted from the tops of the trees. Thus to collect, process and maintain the information several anchor nodes are placed at each side of the isolated terrain. Such anchor nodes are used to localize the position of SNs. The collected information is processed with the position of a SN. Fig. 1 illustrates the studied environment.

In this paper, we use both connection and distance based methods to estimate the distance between SNs. After that we use the location information coming from both directions of the network to increase the localization accuracy for the SNs. The SNs estimate their positions two times from the anchor nodes at the edge of the sensed area to decrease the effect of the error propagation. After this process, we use Kalman Filter to decrease the localization error coming from the longer hop direction based on the information coming from the shorter hop direction. Simulation results show that using information from two different directions significantly

978-1-4799-7100-8/14/$31.00 ©2014 IEEE

increases the localization accuracy.

The remainder of this paper is organized as follows. Related work is covered in Section II. Section III presents the proposed scheme. The simulation environment and results are discussed in Section IV. Conclusions are given in section V.

II. Background and Previous Research

In motivating our proposed research work to address the challenges we discussed, we first will review the previous localization systems techniques in relation to two relevant aspects: multi-hop localization techniques, and the problem of flip ambiguity .

A. Multi-hop Localization

Multi-hop localization schemes are based on either distance-based or connectivity-based strategies. In connectivity-based strategies the SNs obtain the absolute measurements of node distances using Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), or Time Difference of Arrival (TDoA) [2]–[4], while in distance-based strategies the SNs use the connectivity information to estimate the location of SNs based on the position of the anchor nodes [2, 5]–[7].

Niculescu and Nath propose two localization schemes, one based on distance measurement, the other is based on connectivity information [2]. The authors' distance-based scheme is called Distance Vector (DV)-distance, and has the anchor node sending beacon messages to all its immediate neighbors. Immediate (first-hop) neighbors to the anchor node estimate the distance to the anchor by using signal strength measurement. These neighboring nodes then forward the beacon message to the second-hop neighbors to infer the distance to the anchor, and so on until the network is completely covered in a controlled flooding manner. Once an unknown node has three or more distances estimated to different anchor nodes, it computes its position using multilateration.

The second scheme proposed in [2] is the DV-hop, which operates in three stages. First, the the algorithm computes the number of hops for all the SNs to the anchor nodes. Next, the anchor node gets the number of hops required to reach the other anchor nodes, calculating the average length of one hop by dividing the total distance by the number of hops. SNs then estimate the distance by multiplying the number of hops by the average length for one hop.

Stoleru et al. propose a scheme called MDS-MAP that uses multi-dimensional scaling (MDS) to determine SN locations by using only connectivity information [3]. The operation of MDS-MAP consists of three steps: 1) Finding the shortest paths for all pairs; 2) applying classical MDS to the distance matrix; 3) using three or more anchor nodes to transform the relative map to positions based on the positions of anchor nodes.

Wu et al. propose a self-configurable positioning technique for multi-hop wireless networks [4]. A number of nodes at each corner of the network serve together as an anchor for estimating the distances by a Euclidean distance estimation model. The authors used ToA to estimate the distance for each hop. Once ToA information is received by an SN, the sum of these distances is computed by minimizing an error objective function.

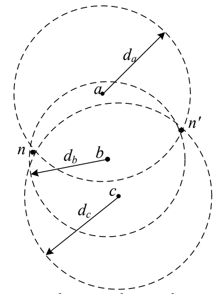

Fig. 2: Collinear anchor nodes a, b and c causing a flip ambiguity for SN n.

The above solutions work well in isotropic networks, i.e., in networks where the hop count between two nodes is proportional to their geometric distance. However, they exhibit a dramatic drop in performance when used in anisotropic networks, i.e., in networks with non-uniform node distribution where there is a concave region at its center. For connectivity based multi-hop localization Savarese et al. [5] propose AHLoS (Ad-Hoc Localization System) algorithm, where a small fraction of nodes have the knowledge of their position to estimate the location of other SNs using collaborative and iterative multi-lateration algorithm. In AHLoS at least three SNs know their position in order to estimate the position of other nodes. Nagpal et al. [6] calculate a global coordinate system for the whole network by estimating the Euclidian distance of each hop between SNs. The SNs use the number of communication hops to estimate how far they are from anchor nodes. When an SN receives at least three different positions from different anchor nodes, the SN combines the distance from the anchor nodes and estimates its position based on the hop count to each anchor. Akbas et al. [7] localize the position of SNs floating in the Amazon river based on stationary anchor nodes placed at a bank of the river. Their localization algorithm uses multi-hop between SNs and anchor nodes. Each SN keeps a single weight value for each anchor it is associated with. The saved weight represents how far the SNs are to each anchor node. The anchor node uses these weights to estimate the SNs position.

B. Flip Ambiguity

When anchor nodes are located on a straight line, the flip ambiguity problem results [8]. Traditionally, the term "flip ambiguity" labels the confusion resulting from collinear anchor nodes. As illustrated in Figure 2, anchor nodes a, b, and c are collinear. Node n estimates its position through measurements d_a, d_b, and d_c. Each measurement defines a ranging circle centered at the anchor node. Due to measurement errors, the three measured circles do not intersect at a common point, which causes ambiguities in determining whether the position of the SN is n or n' [9].

The problem of flip ambiguity is approached from different perspectives in the literature. The work done by Eren et al. and Goldenberg et al. test the unique localization conditions and construct localizable networks using the rigidity theory [10, 11]. The authors show that maintaining a global rigidity in the localized networks decreases the collinearity of anchor nodes. However, it is hard to maintain the global rigidity of the network unless it is compensated by a priori information from the network [12].

Localization algorithms in [9, 13] identify possible flip ambiguities caused by collinearity of anchor nodes and decrease the effect of flip ambiguity during the localization processes. Moore et al. propose a robust quadrilaterals localization scheme to identify possible flip ambiguities in fully connected sensor quadruples [9]. The scheme has two steps. In the first step, the distance measurement between two anchor nodes S_A and S_B is used to estimate the two possible locations of the un-localized SN S_D. Then in the second step, a third anchor node S_C is used to decide which of the two possible locations for the un-localized SN satisfy the distance constraint. If both locations satisfy the condition, the scheme will ignore this SN. In [13], it was was noted by Sittile that if sensors S_A and S_C are used in the first step in [9] instead of sensors S_A and S_B, and sensor S_B is used in step 2 instead of sensor S_C, this may result in a different value for the robustness criterion, which would affect the overall localization performance. Such dependency is eliminated by including all three permutations when localizing S_D, i.e., (S_A, S_B, S_C), (S_A, S_C, S_B) and (S_B, S_C, S_A). This inclusion, however, increases the computational complexity of the algorithm.

To reduce the error caused by trilateration, Yang et al. [14] propose a sequential localization scheme that estimates SNs location and controls the errors introduced in each step. In their sequential scheme, a set of anchor nodes is chosen and the expected error is tracked in each step to minimize the error. However, flip ambiguity cannot be avoided by error control as it can be triggered even by the smallest errors if the anchor nodes used to localize the SN are collinear. Basu et al. solved the problem of collinearity by using both distance and angle measurements [15], where the localization problem is transferred to a convex form and solved using linear programming. However, the scheme by Basu et al. cannot work if either the distance or angle measurement does not have a clear boundary. Also, the scheme depends on the knowledge of both distance and angle measurements, which requires additional hardware. To identify and reduce the error caused by flip ambiguities, Kannan et al. introduce a scheme that recognizes SNs with possible flips using simulated annealing, and offer a refined scheme through the use of a ranging model and a bounder check, despite the refinement, however, the scheme may not identify all flips [16].

III. The Side Localization Scheme

The side localization scheme is described in details in this section. The two main goals for this approach are: 1) to enhance the position estimation of localized SN without deploying anchor nodes in the sensing area as the cost of anchor node is much higher than normal SNs and 2) to propose a solution that overcomes the collinearity problem that appear from using a mobile vehicle that moves in straight lines

To solve the collinearity that results from the studied environment that is shown in Fig. 1, we propose a new localization scheme that estimates the distance between two nodes using RSSI measurements. SNs then estimate their position using the estimated distance, message flow direction and laws of trigonometry [17]. In the following, we first formulate the localization problem. The proposed scheme is then described. Finally, Kalman Filter is used to reduce localization errors.

A. Problem Formulation

We consider a two-dimensional WSN localization problem, where there are number of anchor nodes located at both ends of the sensing area as shown in Fig. 1. Assume that there are M SNs that are deployed randomly in the sensing area, where the SNs need to localize their positions. The position of i^{th} SN is denoted by $\mathbf{x}_i = [x_i \quad y_i]^T$. The distance measured between the i^{th} and j^{th} SN is

$$d_{i,j} = \sqrt{(x_j - x_i + x_{err})^2 + (y_j - y_i + y_{err})^2} \quad where\ x_{err}\ and\ y_{err} \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

where $r_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the noise free distance between SN i and j, and $\varepsilon_{i,j} \sim \mathcal{N}(0, \sigma_{i,j}^2)$ represents the uncorrelated noise. $\sigma_{i,j}^2$ is assumed to be accurately estimated and is known a priori [18]. Without loss of generality we assume that there are N anchor nodes located at the edge of the sensing area. Let α_i^l and $\alpha_i^r, \forall i = 1, 2, \ldots, n$, respectively be the positions where the left and right anchor nodes broadcast their positions while they are moving on the edges of the sensing area. Each SN localizes its position two times from the left and right sides and saves the number of hops to the left and right edge. The estimated positions of i^{th} SN from the left and right side that are p and q hops away from the left and right anchor nodes are represented by $\tilde{\mathbf{x}}_i^{l,p}$ and $\tilde{\mathbf{x}}_i^{r,q}$, respectively. For example, $\mathbf{x}_k^{l,3}$ means node k received a packet that is 3 hops away from the left edge.

B. Side Localization Scheme

We extended the scheme we proposed in [19] to work with fixed anchor instead of mobile anchor. The extended scheme use either hop count or RSSI to estimate the distances between SNs. In this paper we explain the algorithm used to estimate the location of SN using hop information, while [19] explains in details the algorithm using RSSI.

The algorithm works as follow: Each SN requires a minimum of two SNs with a known position from each direction in order to estimate its position from one direction.

This localization scheme has three different lists: *unknownPosList*, *leftPosList* and *rightPosList*. The *unknownPosList* saves anchor SN positions when the direction of the message is unknown at the beginning. The *leftPosList* and *rightPosList* are used when the SN has enough information that enables the SN to identify whether the message is coming from the right or left direction. This algorithm has three phases to estimate the location of SNs.

In the first phase, the anchor nodes on the side of the sensing area send their location with the hop count initialized with zero. The SNs saves the number of hops to each anchor node through the shortest path along with the anchor node's location in the *unknownPosList*. Thus at the end of the first phase, each SN maintains a list of $\{x_i, y_i, h_i\}$, where x_i and y_i are the coordinates of anchor i and h_i is the shortest number of hops to reach anchor i. SNs exchange the shortest hop location packets only with their neighbors. When an anchor node receives a location packet from other anchor nodes, it estimates the average distance for a single hop for the entire network. The average distance of a single hop of anchor i is calculated as follows:

$$avgHop_i = \sum_{j=1}^{M} \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{h_j}, \;\; where \; i \neq j. \tag{2}$$

In the second stage, the anchor nodes broadcast their estimates of the average distance for a single hop. When a SN receives the message of the average distance for a single hops, it starts to classify the anchor nodes saved in its unknownPosList. This process is done as follow. First, the SN calculates the different between the ys of the first 2 saved elements in the unknownPosList. If the difference between them is smaller than a given threshold, this means the change in y coordinate is very small, thus these two anchors are in the same direction and vice versa. But if the difference between them is greater than the given threshold, then this means that each anchor is located in different direction. If y_1 is less than y_2, then y_1 is saved in leftPosList and y_2 is saved in rightPosList and viceversa. This is repeated for all the elements in the unknownPosList. After the SN has classified the elements in the unknownPosList, it starts to estimate its location $\tilde{\mathbf{x}}_k^{l,p}$ using leftPosList and $\tilde{\mathbf{x}}_k^{r,q}$ using rightPosList s stated in [19], where , $\tilde{\mathbf{x}}_k^{l,p}$ is the estimated position from the left direction that is p hops away from the left edge and $\tilde{\mathbf{x}}_k^{r,q}$ is the estimated position from the right direction that is q hops away from the right edge. The value of the distance of average hop received as the distance between the distance between SNs. After that the SNs forward its location as long as with the distance of average hop received.

In the final stage, the anchor nodes estimates its location using the information of anchor nodes coming from the other direction. i.e. if the anchor node is one the left side, it uses the location information coming from the right side to estimate its location. After that the anchor node compares the estimated error with the actual location and identify the localization error and divide it by the number of hops to estimate the average localization error per hop. The anchor node forwards the localization error the SNs. The SNs uses the average localization error to enhance its location using the Kalman Filter. The Kalman Filter is discussed in details in the following subsection.

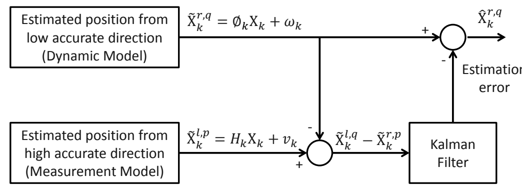

Fig. 3: The Kalman Filter left/right integration for $p < q$ [20].

After SN i has estimated its location from both directions, the SN can use the mean to estimate its position. However, the estimated position from the direction with the larger number of hops contains more errors than the direction with smaller hops number (i.e., if $q < p$, then $\tilde{\mathbf{x}}_k^{r,q}$ is more accurate than $\tilde{\mathbf{x}}_k^{l,p}$). By using mean the SN does not take into consideration the error propagated for each hop. Thus, the weighted mean can be used to consider the propagation error for each hop. The weighted mean estimation is calculated as follow:

$$\tilde{\mathbf{x}}_k = \frac{(\tilde{\mathbf{x}}_k^{l,p} \times q) + (\tilde{\mathbf{x}}_k^{r,q} \times p)}{p + q} \tag{3}$$

However, the weighted mean does not take into consideration the error gained from each hop. Thus the Kalman Filter is used in place of the weighted mean. The Kalman Filter reduces the estimation errors by taking into consideration the error gained from each hop along with the number of hops.

C. Location Enhancement using Kalman Filter

We propose to use Kalman Filter in place of the weighted mean. Kalman Filter is an optimal estimation tool that enhances one measurement giving a more accurate measurement from another source using a sequential recursive algorithm [20]. We use a Kalman Filter that corrects the estimated location of the side that has the larger number of hops using the information provided from the side that has the smaller number of hops. This helps to estimate the error resulting from the larger number of hops. Figure 3 shows the Kalman Filter block diagram used in this study.

In order to complete the development of the state-space of the discrete time Kalman Filter equations, the system's dynamic and measurement models for the SN have to be defined. The system static and measurement model equations if $p < q$ ($\tilde{\mathbf{x}}_k^{l,p}$ and $\tilde{\mathbf{x}}_k^{r,q}$ are switched if $q < p$) are represented as follows, respectively:

$$\mathbf{x}_k^{r,p} = \phi_k \mathbf{x}_k + \omega_k^p \tag{4}$$

$$z_k = \mathbf{x}_k^{l,q} = H_k \mathbf{x}_k + \nu_k^q \tag{5}$$

where $\mathbf{x}_k$ is the actual location of the SN, ϕ_k is a static transmission matrix that relates $\mathbf{x}_k$ with its previous state. Since there is no change in the SN state, i.e., location, the $\phi_{k,k-1}$ matrix is represented as an identity matrix, $Q_k^p = E[\omega_k^p \; (\omega_k^p)^T]$ and $R_k^q = E[\nu_k^q \; (\nu_k^q)^T]$ are the covariance matrices for the p and q hop count coming from the left and right directions. Q_k and R_k are assumed to be uncorrelated as they are received from two different directions with different numbers of hops. The Q_k and R_k for single hop is calculated at the anchor node

TABLE I: A Summary of Kalman Filter equations for $p < q$.

Kalman Filter Algorithm	
Covariance matrix initialization: $P_0 = E\langle(\mathbf{x}-\tilde{\mathbf{x}}_0)(\mathbf{x}-\tilde{\mathbf{x}}_0)^T\rangle$	(6)
State estimate extrapolation: $\hat{\mathbf{x}}_k^r(-) = \phi_k \hat{\mathbf{x}}_{k-1}^r(+)$	(7)
A priori covariance matrix: $P_k(-) = \phi_k P_k(+)\phi_k^T + Q_{k-1}$	(8)
Kalman gain matrix: $K_k = P_k(-)H_k^T(H_k P_k(-)H_k^T + R_k)^{-1}$	(9)
Update the estimated location: $\hat{\mathbf{x}}_k^r(+) = \tilde{\mathbf{x}}_k^r(-) + K_k(\tilde{\mathbf{x}}_k^l - H_k\tilde{\mathbf{x}}_k^r(-))$	(10)
A posteriori covariance matrix: $P_k(+) = (I - K_k H_k)P_k(-)$	(11)

after estimating its location from the anchors node located at the other side and dividing the location error by number of hops. In order to calculate Q_k and R_k for multi-hops, we multiple the average error per hop by the number of hops.

The Kalman Filter equations used in this study are summarized in Table I. The steps using Kalman Filter are as follows if $p < q$ ($\tilde{\mathbf{x}}_k^{l,p}$ and $\tilde{\mathbf{x}}_k^{r,q}$ are switched if $q < p$). First, the covariance matrix is initialized at the left border SN using Equation 6. After that, the SN calculates the priori covariance and Kalman gain matrices using Equations 8 and 9. Then, the right location $\tilde{\mathbf{x}}_k^r$ is updated to $\hat{\mathbf{x}}_k^r$ using Equation 10. Later, the SN calculates the posteriori covariance matrix using Equation 11 using the computed values of K, pervious state of $\hat{\mathbf{x}}_k^r(-)$ and the accurate data $\hat{\mathbf{x}}_k^l$ in Equations 9, 7 and 8 respectively. Finally, the SN forwards the Posteriori $P_k(+)$ matrix to the next hop SNs to be used as the priori covariance matrix $P_k(-)$. The SNs that are away from the edge of the network do the same steps except they use the received posteriori covariance matrix instead of creating a new one. Finally the SNs estimate their new location using the following equation:

$$\tilde{\mathbf{x}}_k = \frac{\tilde{\mathbf{x}}_k^{l,p} + \hat{\mathbf{x}}_k^{r,q}}{2} \tag{12}$$

IV. Performance Evaluation

In this section, we evaluate the performance of our scheme in two localization scenarios. The first scenario investigates the accuracy of the localization estimation as the number of hops increases. The second scenario compares the effect of increasing the number of hops by increasing the width of the simulation area. We compare the localization mean square error between ten different estimation techniques. The ten estimations are: 1) DV-Distance; 2) DV-Hop; 3) shortest Hop only using distance measurement; 4) shortest Hop only using hop measurement; 5) using the mean of both sides; 6) the weighted mean of both sides using distance measurement; 7) the weighted mean of both sides using distance measurement; 8) the weighted mean of both sides using hop measurement; 9) Kalman Filter using hop measurement and 10) Kalman Filter using hop measurement. In our simulation, using NS3, the communication range of anchor and SN is set to 30m. All results are averages of ten different independent runs with distinct random seeds.

A. Localization error per number of hops

In this scenario, we compare the localization error for each hop level as the number of hops of the shortest side increases in the same simulation area. We randomly deploy a 200 SNs in a simulation area with dimension of $500m \times 100m$, since we are interested in studying the effect of the number of hops on our localization accuracy, which is affected by the width of the simulated area. Thus, we increase the width of the simulated area to be 4 times the length. The maximum number of hops from one end to another using the above dimension is 20.

Fig. 4 illustrates that as the number of hops increases the localization error increases for all the estimation techniques except for the mean estimation. The figure shows that using Kalman Filter using distance measurement gives the least estimation error, while the mean estimation gives the highest estimation error. The mean estimation for both hop and distance measurement, DV-Hop and DV-Distance give the worst estimation when the difference between the number of hops is larger as the error from the direction that has a larger number of hops is huge, which affects the overall estimation accuracy when we take the mean. However by taking the weighted mean, we give a lower weight for the estimation from the direction that has a larger number of hops.

The weighted mean using hop measurement gives higher localization accuracy than using weighted mean using distance measurement. However Kalman Filter with distance measurement gives higher localization accuracy than using Kalman Filter with hop measurement.

The improvement of Kalman Filter over the weighted mean is between 24% and 44% with an overall mean of 37% for distance measurement, while between 8% and 24% with an overall mean of 16% for hop measurement. The position estimation using one side only gives a very high accuracy when the difference between the two directions is the max (i.e. when the SN is near the edge of the simulation area).

B. Localization error per width change

In this scenario, we compare the overall localization error as we increase the number of hops by increasing the simulation area. We randomly deployed a 200 SNs in a simulation area with width of $100m$ and the width of the simulation area is changed from $200m$ to $400m$ with $40m$ step.

Fig. 5 shows that using the mean gives the worst localization accuracy while using the Kalman Filter gives the best accuracy. It is noted that weighted mean using hop measurements gives better localization accuracy than weighted mean using distance measurements. One the other hand Kalman filter using distance measurements gives better results than using hop measurements. Distance Kalman Filter gives a better localization accuracy than hop Kalman Filter by 12%, distance weighted mean by 30%, hop weighted mean by 22%, DV-Distance by 51%, and DV-hop by 44% . The reason that the Kalman Filter gives a better results than the weighted mean is the Kalman Filter estimates and assigns the weights automatically. Moreover, Kalman Filter takes into consideration the propagation error

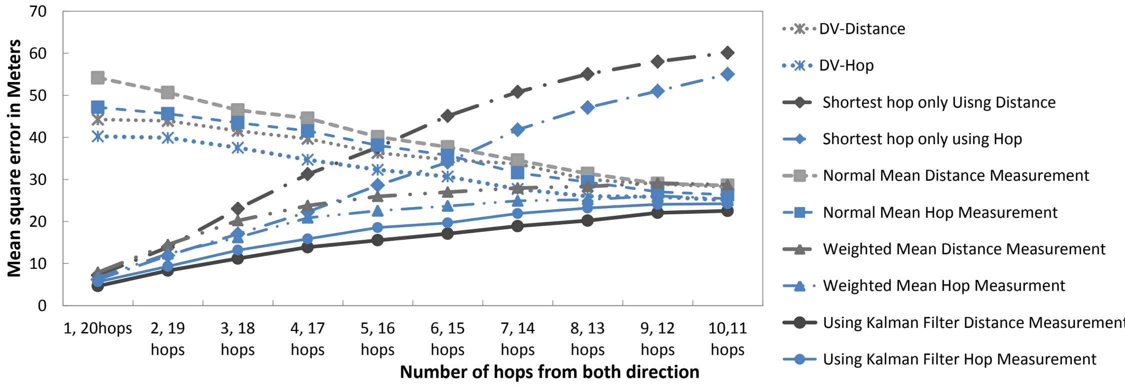

Fig. 4: Relation between localization error and hops.

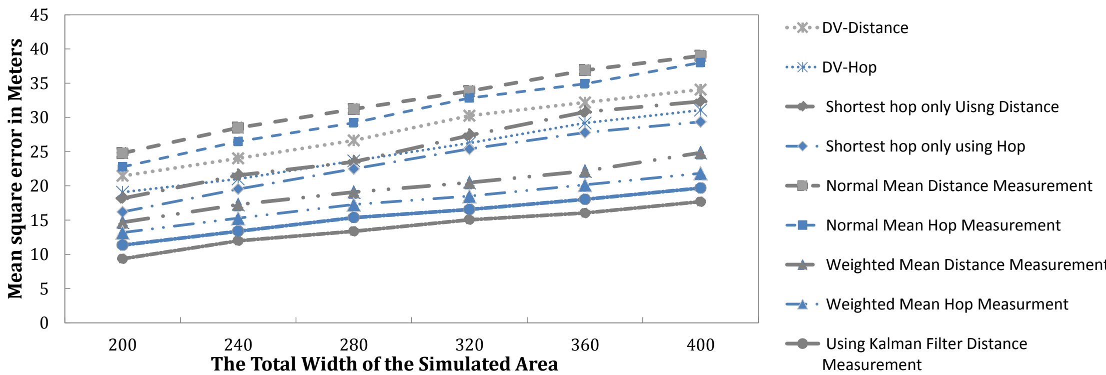

Fig. 5: Relation between localization error and width increase.

per hop while the weights in the weighted mean are fixed and the propagation errors per hop are not taken into consideration.

V. Conclusion

In this paper, we enhanced the side localization scheme to localize SNs using fixed anchor nodes placed at the side of the sensing area using multihop. The side localization scheme using hop measurement is divided into three phases. The second phase, the scheme estimates the location for each node from two different directions using the estimated distance between nodes and the flow direction of the message. The third phase, we apply Kalman Filter to improve localization accuracy. We study the different between using distance and hop measurement in side localization. From simulation study, we show that our side localization scheme outperform both DV-Hop and DV-Distance. Simulations results show that using hop measurement gives higher localization accuracy than using using distance measurement in weighted mean. The use of Kalman Filter tremendously increases the localization accuracy for both distance and hop measurement, however distance measurement gives better localization accuracy than hop measurement.

Acknowledgment

This work was made possible by NPRP grant # NPRP4-553-2-210 from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] J. Wang, R. Ghosh, and S. Das, "A survey on sensor localization," *Journal of Control Theory and Applications*, vol. 8, pp. 2 – 11, 2010.

[2] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, vol. 22, Jan. 2003.

[3] R. Stoleru, T. He, and J. Stankovic, "Range-free localization," in *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, ser. Advances in Information Security, R. Poovendran, S. Roy, and C. Wang, Eds. Springer US, 2007, vol. 30, pp. 3–31.

[4] H. Wu, C. Wang, and N.-F. Tzeng, "Novel self-configurable positioning technique for multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, 2005.

[5] C. Savarese, J. Rabaey, and J. Beutel, "Location in distributed ad-hoc wireless sensor networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 2001, pp. 2037 – 2040.

[6] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network," in *Information Proc. in Sensor Networks*, 2003, vol. 2634, pp. 553 – 553.

[7] M. Akbas, R. Matthias, and D. Turgut, "Local positioning for environmental monitoring in wireless sensor and actor networks," in *IEEE 35th Conference on Local Computer Networks*, Oct. 2010, pp. 806 –813.

[8] G. Han, H. Xu, J. Jiang, L. Shu, T. Hara, and S. Nishio, "Path planning using a mobile anchor node based on trilateration in wireless sensor networks," *Wireless Communications and Mobile Computing*, 2011.

[9] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *2nd International Conf. on Embedded Networked Sensor Systems*, 2004, pp. 50–61.

[10] T. Eren, O. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Anderson, and P. Belhumeur, "Rigidity, computation, and randomization in network localization," in *23rd AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, 2004, pp. 2673–2684.

[11] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. Yang, A. Young, A. Morse, and A. Savvides, "Network localization in partially localizable networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 1, 2005, pp. 313–326.

[12] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529 – 2553, 2007.

[13] F. Sottile and M. Spirito, “Robust localization for wireless sensor networks,” in *Proceedings IEEE 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2008, pp. 46–54.
[14] Z. Yang and Y. Liu, “Quality of trilateration: Confidence-based iterative localization,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 631–640, 2010.
[15] A. Basu, J. Gao, J. Mitchell, and G. Sabhnani, “Distributed localization using noisy distance and angle information,” in *7th International Symposium on Mobile ad hoc Networking and Computing*, 2006, pp. 262–273.
[16] A. Kannan, G. Mao, and B. Vucetic, “Simulated annealing based wireless sensor network localization with flip ambiguity mitigation,” in *63rd Vehicular Technology Conference*, vol. 2, 2006, pp. 1022–1026.
[17] M. D. Hestenes and R. Hill, *Algebra and Trigonometry*, 2nd ed. Prentice Hall Professional Technical Reference, 1986.
[18] W. Ibrahim, N. Ali, A.-E. Taha, and H. Hassanein, “Improving the accuracy of simulation models for localization schemes,” in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, April 2014, pp. 1–6.
[19] W. M. Ibrahim, A.-E. M. Taha, and H. S. Hassanein, “Using smart vehicles for localizing isolated things,” *Computer Communications*, 2014.
[20] M. Grewal and A. Andrews, “Kalman Filtering: Theory and Practice Using MATLAB,” 2001.