Robust Wireless Multihop Localization Using Mobile Anchors

Walid M. Ibrahim
School of Computing,
Queen's University
Kingston, Ontario K7L 3N6
Canada
Email: walid@cs.queensu.ca

Abd-Elhamid M. Taha
College of Engineering,
Alfaisal University
PO Box 50927, Riyadh 11533
Saudi Arabia
Email: ataha@alfaisal.edu

Hossam S. Hassanein
School of Computing,
Queen's University
Kingston, Ontario K7L 3N6
Canada
Email: hossam@cs.queensu.ca

***Abstract*—Knowing the position of sensor nodes in an environmental monitoring is useful to identify the location of events. However deploying GPS receivers or other anchor sensors is expensive, since the role of anchor nodes ends after localizing sensor nodes' positions and they are transferred into ordinary sensor nodes. In this paper, we introduce a new localization scheme for a wireless sensor network that can localize sensor nodes using a collinear and non-collinear mobile anchor node. This scheme benefits from the estimated distance between neighbor nodes and additional information provided by the anchor node about the flow direction of the message. Each node localizes it's position from two independent directions. A Kalman Filter is then used to improve the location accuracy for each node. Through simulation studies, we show that the scheme using a Kalman Filter decreases the estimation errors than using single direction by 31% and 16% better than using weighted averages. As well, our new scheme overcomes the collinearity problem that appears from using mobile anchor nodes.**

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

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. However, anchor nodes are more expensive than SNs. Moreover, anchor nodes have a limited use after the localization process is completed and the anchor nodes are transferred into SNs. Therefore, the current research direction in WSN localization moves toward designing a new localization schemes that use mobile anchors to decrease the cost of the entire network and minimize the excessive waste of resources from anchor nodes [1, 2].

The environment considered in this work is an isolated terrain (i.e., dense rain forest or rocky terrain) with no central access roads. SNs are usually randomly deployed into isolated terrain to collect information about the environment and send such information to the SNs at the edge. To collect and process the information, two mobile vehicles mounted with GPS units uses two different roads located at each side of the isolated terrain. These two vehicles are used to localize the position of SNs. The collected information is processed with the position of a SN. Fig. 1 illustrate the studied environment.

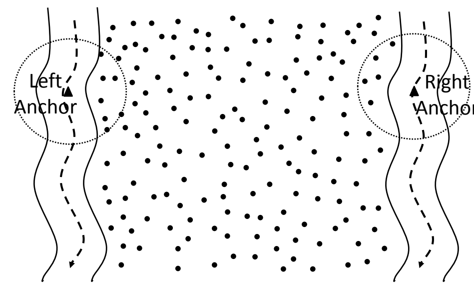


Fig. 1: WSN with Collinear Anchor Nodes.

In this paper, we use the location information coming from both directions of the network to increase the localization accuracy for the SNs. Due to the limited transmission range of a WSN (approx. 30m), the SNs located away from the road edge will not receive signals from the mobile anchor node. Thus, such SNs cannot directly estimate their positions. A multihop localization scheme can be used to estimate the position of the SN located in the middle of the isolated terrain, but due to the error propagation the estimation accuracy decreases as the number of hops increases [3]. The SNs estimate their positions two times from both mobile anchors 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 increases localization accuracy.

The access road can be curved or straight. However, when a mobile anchor node moves in a straight line to localize SNs, the collinearity problem appears. The collinearity problem appears when the anchor nodes are on the same line [4]. In this case, it is hard to identify whether the SN is on the left or right of the anchor nodes, which cause the position of the SN to be flipped [5]. In our proposed scheme, we design an algorithm that overcomes the collinearity that appears when mobile anchor node moves in a straight line by benefit from the direction of the message send by the mobile anchor node.

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. Related Work

WSN localization has been an active research topic in recent years [6]. In this paper, we focus on localization schemes that use multihop or mobile anchor node.

A. Multihop Localization

Multihop localization schemes are based on either connectivity or distance based strategies. In connectivity-based strategies, the SNs use the connectivity information to estimate the location of SNs based on the position of anchor nodes. Niculescu and Nath [7] proposed ad-hoc positioning system (APS) algorithm that uses hop by hop propagation with the position of anchors to estimate the distance between each SN and the anchor nodes. APS then uses the distance between SNs and all anchor nodes using the triangulation procedure similar to the one that is used in the GPS algorithm. Nagpal et al. [8] calculate a global coordination 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 anchors. When a SN receives at least 3 different positions from different anchors, the SN combines the distance from the anchors and estimates its position based on the hop count to each anchor. Akbas et al. [3] localize the position of SNs flooding in the Amazon river based on a stationary anchors placed at the bank of the river. Their localization algorithm uses multihop between SNs and anchor node. Each SN keeps a single weight value for each anchor it is associated with. The saved weight represents how far is the SNs to each anchor node. The anchors use these weights to estimate the SNs position. None of aforementioned schemes considers the collinearity of anchors in their design.

Moore et al. proposed a robust quadrilaterals localization scheme that avoids flip ambiguity caused by collinearity of anchors [5]. Basu et al. solved the problem of collinearity by using both distance and angle measurements [9]. They transferred the localization problem to convex form and solved the convex form using linear programming. However, Basu et al. scheme cannot work if either the distance or angle measurement does not have a clear bound. Moreover, Basu et al. scheme depends on the knowledge of both distance and angle measurements, which makes their scheme to be expensive. Kannan et al. identify the SNs that have possible flips using simulated annealing technique, then they refine the results of these SNs using ranging model and boundary check [10]. However, Kannan et al. scheme may not identify all flips.

To the best of our knowledge, existing approaches exploit multihop localization schemes using a stationary anchor node. In our study, we use a mobile anchor node. Moreover, existing schemes either avoid or try to refine the result of flipped SNs. Our scheme, on the other hand, is designed to localize SNs using either collinear or non-collinear anchor nodes.

B. Mobile Anchor Node Localization

Mobile anchor nodes are proposed to lower the cost of utilizing a large number of stationary anchor nodes. Pathirana et al. use mobile anchor nodes that move in random paths to localize SNs in delay-tolerant sensor network [11]. Hauanxiang et al. use a mobile anchor node that adopts Gauss-Markov motion model, and used the weighted centroid algorithm to localize the position of SNs [1]. They used a genetic algorithm to decrease the estimated error. Guangjie et al. show that localization using mobile anchor node that moves randomly results in poor performance in terms of localization time and accuracy [4].

To overcome the poor performance of random movement for mobile anchor nodes, Koutsonikolas et al. proposed a pre-determined path for a single mobile anchor node [12]. They also address the collinearity problem when a single anchor node is used. The collinearity problem is defined as the localization error of an un-localized SNs is large when it receives a 3 beacon localization messages from anchor node that moves in a straight line. Different trajectory types such as Circle, S-Curves, Rectangle, Spiral and Triangle are proposed to overcome the collinearity problem of a single mobile anchor node [2, 4, 13]. Predetermined paths are efficient if the deployment area has a regular shape (i.e. square or rectangle) and the density of sensors is uniform, but can lead to wasteful anchor movement in irregular areas and non-uniform sensors' density. Wang et al. proposed a scheme that handle non-uniform and irregular placement scenarios [14]. In Wang et al. scheme, the mobile anchor node sends a start message all over the network and when a SN receives the start message it adds the neighbor SNs surrounding it and then the SN forwards the message. When the anchor node receives the start message back, it calculates the shortest path to localize all SNs. The anchor node moves in half-circle movements to avoid the collinearity property for the mobile anchor node.

Existing schemes using a mobile anchor node are a single hop schemes. Moreover, they use a predefined path that avoids the collinearity of anchor nodes by using non-linear path that increase the distant traveled by the anchor node. In our work, we combine both mobile anchors and multihop localization. Moreover, we propose a scheme that overcomes the collinearity of the anchor node when it moves in a straight line, by using the direction of the message to localize SNs.

III. The Robust Multihop Localization

The robust multihop localization scheme using two mobile anchor nodes 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 [15]. 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 two roads 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

$$d_{i,j} = d_{j,i} = r_{i,j} + \varepsilon_{i,j} \qquad \forall i, j = 1, 2, \ldots, M \tag{1}$$

where $r_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the noise free distance between node 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 [16]. Without loss of generality, let α_i^l and $\alpha_i^r, \forall i = 1, 2, \ldots, n$, respectively be the positions where the left and right mobile anchor nodes broadcast their positions while they are moving on the edges of the sensing area. We assume that each mobile anchor node knows whether it is moving on the left or right road of the sensing area. Such information is currently available in modern intelligent vehicles. The anchor node sends the direction of the packet (i.e. whether the packet is coming from the left or right side of the sensing area) along with its position in the localization packet that is sent to localize the SN. 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.

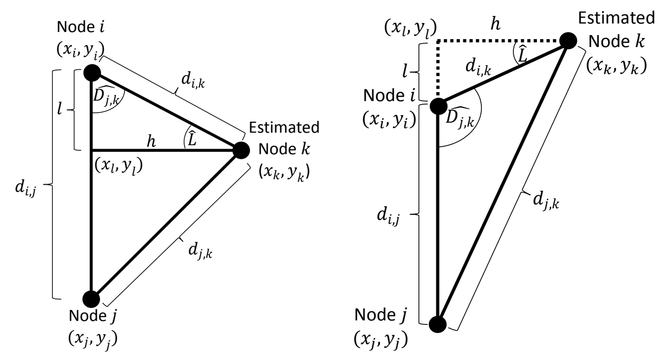
B. Robust Localization Scheme Development

In this scheme each SN estimates its position $\tilde{\mathbf{x}}_i$ using estimations from both left ($\tilde{\mathbf{x}}_i^{l,p}$) and right ($\tilde{\mathbf{x}}_i^{r,q}$) directions. Our localization scheme benefits from the direction of the message send by anchor nodes to estimate the SNs position. Each SN requires a minimum of two nodes with a known position in addition to the direction of the message sent in order to estimate its position from each direction. Moreover the SN uses the left and right estimations to increase the estimation accuracy of SNs.

Our localization scheme works as follow: when a SN receives a localization message, it ensures that the message is not replicated by checking the sequence number in the message. If the received sequence is previously saved in the SN, the SN discards the message as it is repeated, otherwise the SN saves the sequence number in the SN and continue to process the message. After that, the SN checks the number of hops of the received message. If the received number of hops is larger than the saved number of hops, the SN discards the

Algorithm 1 Process the received position message from the left direction.

```
procedure CheckMsgFromLeft(recSeqNum, recHopNum, sourceIP, x_i)
    if HopsNumber == 1 then
        if recSeqNum == saveSeqNum for the same sourceIP then
            return (Repeated message)
        else
            currentNode ← borderNode
        end if
    else if HopsNumber > 1 then
        if currentNode is a Border Node then
            This is a border node.
            return (message is coming from longer route)
        else if recHopsNum > savedLeftHopsNum then
            return (message is coming from longer route)
        end if
    end if
    saveSeqNum ← recSeqNum for sourceIP
    Add x_i to the position list
    if size of position list > 2 then
        EstimateNodesPosition()
        Send message with current estimated position
    end if
end procedure
```



(a) When $\widehat{D}_{j,k} < 90$ (b) When $\widehat{D}_{j,k} > 90$

Fig. 2: Estimating Node k Using The Estimated Distance.

message as it is coming from a longer route, otherwise the SN saves the received number of hops and this hop number represents how far the SN from the edge from the received direction. The SN then saves the position of the node that sent the position message. Algorithm 1 shows the main procedure to check the message coming from the left direction. The same procedure is done for the right side.

After a SN receives two or more location packets that have the same number of hops, it estimates the three distances $d_{i,j}, d_{i,k}, d_{j,k}$ shown in Fig. 2, for each pair, where the positions of $\mathbf{x}_i$ and $\mathbf{x}_j$ are previously known (i.e. two different location for two mobile anchor nodes or normal nodes that have estimated their position in a previous step) and $\mathbf{x}_k$ is the location of node k that needs to estimate its position.

In order to estimate $\mathbf{x}_k$, we need to estimate the coordinate of point $\mathbf{x}_l$ representing the intersection between $d_{i,j}$ and the height h of triangle $d_{i,j}, d_{i,k}, d_{j,k}$. The coordinates of $\mathbf{x}_l$ is calculated as follow:

$$\begin{bmatrix} x_l \\ y_l \end{bmatrix} = \begin{cases} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \frac{l}{d_{i,j}} \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} & \text{for } \widehat{D}_{j,k} \leq 90 \\ \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \frac{l}{d_{i,j}} \begin{bmatrix} x_j - x_i \\ -(y_j - y_i) \end{bmatrix} & \text{otherwise} \end{cases} \tag{2}$$

where l is calculated as:

$$l = \sqrt{h^2 + d_{i,k}^2 - (2 \times h \times d_{i,k} \times \cos(\widehat{L}))} \tag{3}$$

In order to calculate l we need to calculate h and angle $\widehat{L}$. h is given by:

$$h = \frac{2 \times d_{i,k} \times d_{i,j} \times \sin(\widehat{D}_{j,k})}{d_{i,j}} \tag{4}$$

where the angle $\widehat{D}_{j,k}$ is calculated using:

$$\widehat{D}_{j,k} = \cos^{-1}(\frac{d_{i,j}^2 + d_{j,k}^2 - d_{i,k}^2}{2 \times d_{i,j} \times d_{j,k}}) \tag{5}$$

and the angle $\widehat{L}$ as:

$$\widehat{L} = \begin{cases} 90 - \widehat{D}_{j,k} & \text{for } \widehat{D}_{j,k} \leq 90 \\ \widehat{D}_{j,k} - 90 & \text{otherwise} \end{cases} \tag{6}$$

After estimating the coordinates of $\mathbf{x}_l$, we need to get the slope between node i and j to calculate $\mathbf{x}_k$ to consider the shift in x and y coordinate caused by the slope of the line. This allow us to estimate the position of the SN using collinear and non-collinear anchor nodes. The slope of $d_{i,j}$ is:

$$m_{d_{i,j}} = \tan^{-1} \frac{y_j - y_i}{x_j - x_i} \tag{7}$$

Node k estimates its position based on the direction of the message. Thus, if the message is coming from the left direction, Node K estimates $\tilde{\mathbf{x}}_k^{l,p}$ using the following equation:

$$\tilde{\mathbf{x}}_k^{l,p} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_l \\ y_l \end{bmatrix} + h \begin{bmatrix} sin(m_{d_{i,j}}) \\ -\cos(m_{d_{i,j}}) \end{bmatrix} \tag{8}$$

otherwise if the message is coming from the right direction, Node K estimates $\tilde{\mathbf{x}}_k^{r,q}$ using the following equation:

$$\tilde{\mathbf{x}}_k^{r,q} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_l \\ y_l \end{bmatrix} + h \begin{bmatrix} -\sin(m_{d_{i,j}}) \\ \cos(m_{d_{i,j}}) \end{bmatrix} \tag{9}$$

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. Algorithm 2 shows the procedure for estimating the node's position from the left and right directions.

After node k 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:

$$\tilde{\mathbf{x}}_k = \frac{(\tilde{\mathbf{x}}_k^{l,p} \times q) + (\tilde{\mathbf{x}}_k^{r,q} \times p)}{p + q} \tag{10}$$

Algorithm 2 Estimate the position of Node k

procedure ESTIMATENODEKPOSITION($\mathbf{x}_i$, $\mathbf{x}_j$, $d_{i,k}$, $d_{j,k}$, $messageDirection$)
  $d_{i,j} \leftarrow \sqrt{(x_j - x_i)2 + (y_j - y_i)2}$
  Calculate angle $\widehat{D}_{j,k}$ using eq 5
  Calculate the length of h using eq 4
  if $\widehat{D}_{j,k} \leq 90$ **then**
    $\widehat{L} \leftarrow 90 - \widehat{D}_{j,k}$
    Calculate l using eq. 3
    $x_l \leftarrow x_i + \frac{l}{d_{i,j}} \times (x_j - x_i)$
    $y_l \leftarrow y_i + \frac{l}{d_{i,j}} \times (y_j - y_i)$
  else
    $\widehat{L} \leftarrow \widehat{D}_{j,k} - 90$
    Calculate l using eq. 3
    $x_l \leftarrow x_i + \frac{l}{d_{i,j}} \times (x_j - x_i)$
    $y_l \leftarrow y_i + \frac{l}{d_{i,j}} \times -(y_j - y_i)$
  end if
  Calculate $m_{d_{i,j}}$ using eq 7
  if $messageDirection == left$ **then**
    $p \leftarrow savedLeftHopNum$
    estimate $\tilde{\mathbf{x}}_k^{l,p}$ using eq 8
  else if $messageDirection == right$ **then**
    $q \leftarrow savedRightHopNum$
    estimate $\tilde{\mathbf{x}}_k^{r,q}$ using eq 9
  end if
end procedure

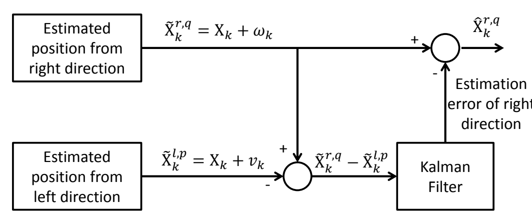


Fig. 3: The Kalman filter left/right integration for $p < q$ [17].

However, the weighted mean does not take into consideration the error gained from each hop. We propose to use Kalman Filter 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

Kalman Filter is an optimal estimation tool that enhances one measurement giving a more accurate measurement from another source using a sequential recursive algorithm [17]. We use a loosely coupled Kalman Filter, where it corrects the estimated location of the side that has larger number of hops using the information provided from the side that has smaller number of hops. This helps to estimate the error resulting from the larger number of hops. Fig. 3 shows the Kalman Filter block diagram used in this study.

In order to develop the state-space of the discrete time kalman filter equations, the system dynamic and measurement models for the SN has to be defined. The system dynamic and measurement models 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:

$$\mathbf{x}_k^{l,p} = \phi_{k,k-1}\mathbf{x}_k + Q_k^p \tag{11}$$

$$\mathbf{x}_k^{r,q} = \phi_{k,k-1}\mathbf{x}_k + R_k^q \tag{12}$$

where $\mathbf{x}_k$ is the actual position of the SN, $\phi_{k,k-1}$ is a dynamic matrix that relates $\mathbf{x}_k$ with its previous state. Since there is no change in the node state (i.e. position), 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.

Cho et al. calculate Q_k and R_k for single hop as $\frac{R_3}{2} \times I$, where R is the normal distribution of error placement for single hop [18]. In order to calculate Q_k and R_k for multihops, we expanded their proof to calculate Q_k and R_k for multiple hops[1]. Q_k^p and R_k^q are calculated in our work as follows:

$$E[\omega_k^p \ (\omega_k^p)^T] = \begin{bmatrix} \frac{\sum_{i=1}^{p} (\sigma_i^2)^3}{2} & 0 \\ 0 & \frac{\sum_{i=1}^{p} (\sigma_i^2)^3}{2} \end{bmatrix} \quad (13)$$

$$E[\nu_k^q \ (\nu_k^q)^T] = \begin{bmatrix} \frac{\sum_{i=1}^{q} (\sigma_i^2)^3}{2} & 0 \\ 0 & \frac{\sum_{i=1}^{q} (\sigma_i^2)^3}{2} \end{bmatrix} \quad (14)$$

where $\sum_{i=1}^{p} \sigma_i^2$, is the summation of the uncorrelated noise in equation 1 from hop 1 to hop p and similar for $\sum_{i=1}^{q} \sigma_i^2$.

Kalman Filter equations	
Covariance matrix initialization:	
$P_0 = E\langle \tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^T \rangle$	(15)
Priori covariance matrix:	
$P_k(-) = \phi_{k,k-1} P_{k-1}(+) \phi_{k,k-1}^T + Q_{k-1}$	(16)
Kalman gain matrix:	
$K_k = P_k(-) H_k^T (H_k P_k(-) H_k^T + R_k)^{-1}$	(17)
Update the estimated position:	
$\hat{\mathbf{x}}_k^r = \tilde{\mathbf{x}}_k^r + K_k(\tilde{\mathbf{x}}_k^l - \tilde{\mathbf{x}}_k^r(-))$	(18)
A posteriori covariance matrix:	
$P_k(+) = (I - K_k H_k) P_k(-)$	(19)

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

The Kalman Filter equations used in this study are summarized in Table I. The steps of using Kalman Filter are as follow 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 node using equation 15. After that, the SN calculates the priori covariance and Kalman gain matrices using equations 16 and 17. Then, the right position $\tilde{\mathbf{x}}_k^r$ is updated to $\hat{\mathbf{x}}_k^r$ using equation 18. Later, the SN calculates the posteriori covariance matrix using equation 19 and forward its value to the next hop SNs. 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 nodes estimate their new position using the following equation:

$$\tilde{\mathbf{x}}_k = \frac{\tilde{\mathbf{x}}_k^{l,p} + \hat{\mathbf{x}}_k^{r,q}}{2} \quad (20)$$

[1]Details are not shown for space limit.

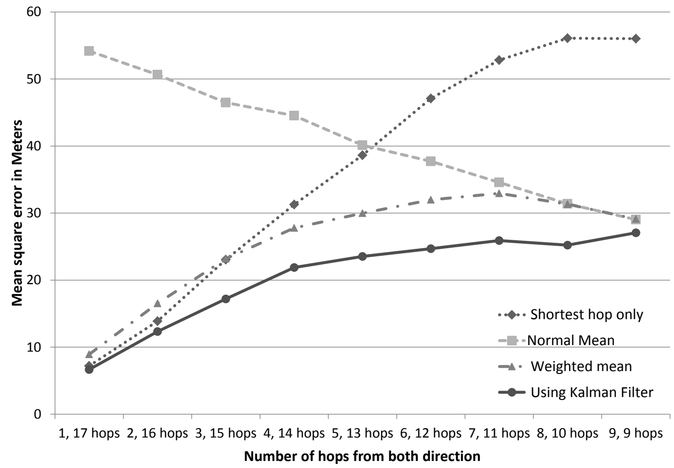


Fig. 4: Relation between localization error and hops.

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 four different estimation techniques. The four estimations are: 1) using one direction that has fewer number of hops; 2) using the mean of both sides; 3) using the weighted mean of both sides using equation 10; and 4) using Kalman Filter using equation 20.

In our simulation, using NS3, the communication range of anchor and SN is set to 30m. The range measurement noise $\varepsilon_{i,j}$ is a zero-mean white Gaussian processes with variance $\sigma_{i,j}^2 = d^2/\text{SNR}$, where SNR is the signal-to-noise ratio received by the node [16]. 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 $400m \times 100m$, 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 17.

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 gives least estimation error, while the mean estimation gives the highest estimation error. The mean estimation gives 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.

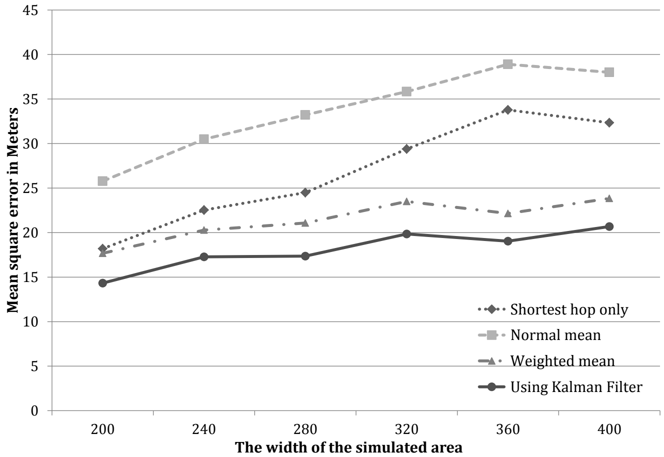


Fig. 5: Relation between localization error and width increase.

The improvement of Kalman Filter over the weighted mean is between 19% and 13% with an overall mean of 15.6%. 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). However, the estimation error increases dramatically for SNs that are 4 hops away from the edge of the network and reaches the maximum in the middle of the network to perform worse than the Kalman Filter by 51.7%.

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. Kalman filter gives a better localization accuracy than weighted mean by 15.6% on average and better than a single side by 31% on average. 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 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 propose a new localization scheme to localize SNs using two mobile anchor nodes through multihop. Our new scheme is divided into two phases. The first 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 second phase, we apply Kalman Filter to improve localization accuracy. Simulations results show that the proposed scheme, which estimates positions from both directions, gives better results than estimating the position from a single direction especially when the number of hops increases. Moreover, the results show that using Kalman Filter increases the accuracy of our scheme. The SNs use the information coming from both directions to improve the localization accuracy. To achieve a similar either to use a high density of stationary SNs, which increases the cost of the WSN, or to use a mobile anchor that have to access the isolated area, which is not feasible in some environment. Moreover, our new localization scheme estimates the position of SNs using a collinear and non-collinear mobile anchor nodes, where in previous work they have to ensure that the path of the mobile anchor is avoiding the non-collinearity of the mobile anchor node.

ACKNOWLEDGMENT

This research is funded by a grant from Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] J. Huanxiang, W. Yong, and T. Xiaoling, “Localization algorithm for mobile anchor node based on genetic algorithm in wireless sensor network,” in *International Conf. on Intelligent Computing and Integrated Systems*, 2010, pp. 40–44.

[2] D.-K. Kim, Y.-S. Kim, and J.-W. Chong, “Hybrid rss/toa wireless positioning with a mobile anchor in wireless sensor networks,” in *Proc. of the 6th Latin America Networking Conference*, 2011, pp. 25–32.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

[10] 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.

[11] P. Pathirana, N. Bulusu, A. Savkin, and S. Jha, “Node localization using mobile robots in delay-tolerant sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 285 – 296, May 2005.

[12] D. Koutsonikolas, S. M. Das, and Y. C. Hu, “Path planning of mobile landmarks for localization in wireless sensor networks,” *Computer Communication*, vol. 30, no. 13, pp. 2577–2592, Sep 2007.

[13] R. Huang and G. Zaruba, “Static path planning for mobile beacons to localize sensor networks,” in *5th International Conf. on Pervasive Computing and Communications Workshops.*, 2007, pp. 323–330.

[14] H. Wang, W. Qi, K. Wang, P. Liu, L. Wei, and Y. Zhu, “Mobile-assisted localization by stitching in wireless sensor networks,” in *IEEE International Conference on Communications*, 2011, pp. 1–5.

[15] M. D. Hestenes and R. Hill, *Algebra and Trigonometry*, 2nd ed. Prentice Hall Professional Technical Reference, 1986.

[16] H. Wei, Q. Wan, Z. Chen, and S. Ye, “A novel weighted multidimensional scaling analysis for time-of-arrival-based mobile location,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3018–3022, 2008.

[17] M. Grewal and A. Andrews, “Kalman Filtering: Theory and Practice Using MATLAB,” 2001.

[18] H. Cho, J. Lee, Y. Lee, and S. W. Kim, “Localization for a mobile node based on chrip spread spectrum ranging,” in *International Conference on Computational Science and Its Applications*, Mar. 2010, pp. 211–216.