Managing Cohort Movement of Mobile Sensors via GPS-Free and Compass–Free Node Localization

Iman Babagoli¹, Hamid Hassanpour²

Abstract- A critical problem in mobile ad hoc wireless sensor networks is that each node's awareness of its position relative to the network that is known as localization. Many measuring techniques are used for it:
1. Via Global positioning system that needs hardwares & don't cast effective.
2. Via Anchors that have globally known positions.
Today work is on small sub-divisions of mobility nodes of GPS.
Most of the usages need sensor network in areas where GPS signals are unreachable. Some errors will caves algorithm to be un-practical. So by adding mobility to network will have node's direction here we introduce 2 algorithm that don't need GPS nodes for location. First (GDL) each sensor is equipped with a digital compass & (GCDL) location is done without compass. Results show that these algorithm increase the on big network.

Index Terms— wireless sensor networks, mobility, localization, GCDL, GDL.

I. INTRODUCTION

Advancement in electronics and wireless communication leads us to make and design sensors with low consumptions, smaller size and suitable price. These make wireless sensors networks. Without localization many of sensors networks are usable localization is through 3 phases. First distance between unknown nodes to source nodes are measured. Second by these measurements, positions of unknown nodes are measured and third by examining theses better results are concluded, localizations are divided into different categories: software and hardware system, way of measuring, big & small systems, kinds of measuring distance techniques. In localization, nodes awareness of its neighbors characters is enough.

To support the movable usages, node should move in a way suitable to its neighbors. So we should know the direction of both. we introduce algorithm called As hoc to measured position and directions of nodes. First one is localization with GPS & second without GPS & compass (GCDL) and localization is down by second one. Errors don’t effect them. Analyze the errors. Examinations show errors during movements. We will discuss 2 algorithms GDL & GCDL:

GPS-Free Directed Localization Each node has a compass pointed to the North. Nodes can measure their direction to their neighbor by limited methods like TOA, TDOA, Ultra–Wideband radios, compass & measurements are influenced by wind, un level grounds, bad equipments and so on.

In GDL algorithm we use a digital compass that included 2 subs – algorithms: core localization and verification.

Core localization algorithm
here two possible situations is produced for each neighbor. Each period include 3 phases.
1) Measuring directions between neighbors, 2) individual moving of nodes 3) exchanging of directions and distance values in period between neighbors.

CORELOCALIZATION(n₁, n₂, v₁, α₁)
1: d₁ ← inter-distance(n₁, n₂)
2: Move node n₁ by v₁ and α₁
3: d₂ ← inter-distance(n₁, n₂)
4: Retrieve v₂ and α₂ from n₂
5: Calculate positions of n₂ using equations (4),(5) and(6)

- verification algorithm
an algorithm that verifies a node’s position using a third neighbor. This step is required to solve the ambiguity of two possible positions per neighbor calculated in core localization algorithm. This use a third neighbor to achier to VERIFICATION (NEIGHBOR PAIRLIST NPL)
1: for each neighbor pair (m,n) in NPL do
2: if m and n are neighbors then
3: dₘₙ measured inter-distance(m,n)
4: for each position pair {mᵢ,nⱼ | i,j=1,2} do
5: Compute Euclidean distance D between mᵢ and nⱼ
6: if D=dₘₙ then
7: mark mᵢ and nⱼ as exact positions

Because of various world disturbances and equipment errors nodes don’t always get a rigid geometry from their measurements. When the core algorithm can’t find meaningful results we reach exceptional movement

¹ Iman Babagoli is with Azad University of Qazin Faculty of IT engineering, i.babagoli@gmail.com
² Hamid Hassanpour is with Shahrood University of Technology, h_hassanpour@yahoo.com
configuration. We discuss about two configurations named equal parallel movement & excessive error configuration:
- Equal parallel movement: here, nodes move parallelly and their distance is fixed. Here geometry is not rigid and infinite solutions existed.
- Excessive error: the main sources of error occur due to distance, actuation & compass measurement inaccuracies.

II. GPS – AND COMPASS – FREE DIRECTED LOCALIZATION (GCDL) ADDITION COST.

Addition cost on hardware and unfavorable physical conditions altering magnetic field restrict the use of compass in certain hostile environments such as disaster areas. To enhance versatility, we relax the requirement for a compass and achieve localization using our algorithm that controls the mobility of the nodes. The main idea is to divide the nodes into two groups, blue (dark) and red (light), and move each group in a stepwise manner while the other remains stationary. We show that, through the use of geo-metric properties, we can localize the neighbors in such a 2-step motion for each group. After localization, nodes can agree on a common virtual north, which essentially has same effect as having a compass.

We outline our 2-step motion algorithm (fig 1) using a blue (dark) and a red (light) node. The blue node is stationary, and the red node performs a 2-step motion to localize the exact position of the blue node. Each time the red node communicates with the blue node, a virtual communication circle is formed, and the distance between the two nodes are measured using a known range measurement method. As shown in fig.1, the first virtual communication circle (C1) results in infinite possibilities for the position of the blue node. The red node can reduce the possible positions to two by calculating the intersection of the first and the second (C2) virtual communication circles. After the second step of the red node, the third (C3) virtual communication circle is formed, allowing the red node to calculate the exact location of the blue node. Therefore, in order to exactly find the position of the blue node. Three virtual communication circles are needed, by keeping track of its own movement distance, in the worst case; the red node gathers enough data to localize the blue node only after 2-step motion.

Fig.1. 2-step motion algorithm. The blue node is stationary, and the red node performs a 2-step motion to localize the exact position of the blue node.

A. Lock step movement

For applications when the mobile sensor nodes are required to move as a chart, we configure our GCDL algorithm to perform directional movement. We assign red & blue colors to nodes at each iteration to color the nodes in the swarm. When the direction of movement is known by all nodes, each group performs a zig-zag movement.

B. Selecting a common north

In GCDL, nodes coordinate their movement based on a pseudo – north, since no compass is used. When it is selected by each node, nodes have to agree on a common north with their neighbors to move as a cohort. When both group nodes complete their lock-step motion, neighbor nodes can communicate more to agree on a common north for the entire swarm. Once the skew between local coordinate systems is calculated, the nodes can use 2 methods to agree on common coordinate system 1) in proactive approach to green on a single coordinate system by using a hierarchy 2) use a reactive method & store only the variance.

III. EXPERIMENTAL EVALUATION

A. Evaluation of GDL algorithm first

We show results from the GDL algorithm under ideal conditions without counting errors and then introduce independent errors on angle & distance measurements to simulate real world disturbances.

1) Experiments under ideal conditions

Here we simulate nodes randomly placed in a 100*100 area under a uniform special distribution. Each is run for 100 epochs & the results are averaged. Each epoch nodes show random walk with random speed [0.5], random angle [0,2π] & fixed radio range of 6.speed & angle are selected by a uniform random distribution. GDL requires 2 neighbors to accurately find each other’s positions. Fig 2 shows the percentage of nodes, whose positions are not calculated for different node densities. For small densities, we see that not all nodes can be localized.
Fig. 2. shows the percentage of nodes, whose positions are not calculated for different node densities. For small densities, we see that not all nodes can be localized.

2) Introduction Measurement error

Now we relax the idea condition assumption and introduce errors on distance & angle measurements. In real-world measurements, many are quite inaccurate due to weather, terrain conditions, and equipment failure. To simulate these, we add uniform range noise to all measurements. These errors change our algorithm's behavior in one of two ways: 1) the algorithm calculates the positions with limited accuracy. 2) Excessive error configurations prevent the algorithm from localizing some of the nodes. Fig. 3 (a) shows the average position error of our algorithm for different rules of noise on angle & fig. 3 (b) we postulate that 30% noise on angle is a significant error rate for real-world conditions.

In order to evaluate the effects of movement speed and wireless range, we tested our GDL algorithm under high noise, with a fixed wireless (10 units) and variable speed values for nodes. We apply an upper limit to the speed, such that the nodes do not move a distance greater than their wireless range per epoch. In any sensor network scenario, if a node moves by a distance greater than its neighborhood will change at each step, which would make it impossible to localize. We can see in fig 4 that the localization error of our algorithm is nearly constant for increasing speed.

B. Evaluation of the GCDL algorithm

Here, we evaluate the effects of various environmental errors on the common north resolution method as the metric to measure the common north errors of the swarm, we calculate the average difference between real & calculate north's of each neighbor per epoch. We simulate 100 nodes in a 100*100 area. In random motion nodes can cover at most 5 units, 8 have a fixed range of 15 units.

We test our algorithm for two different uniform random noise levels: high and low as well as in random
motion and directed motion scenarios. High noise level is up to 30% of distance measurements and up to 2\pi/10 of angle measurements. Low noise level is up to 3% of distance measurements and up to 2\pi/100 angle measurements. In this experiment, after each epoch, nodes resolve the common north variance with their neighbors and the average error per node is calculated. This figure shows that for both random and directed movement scenarios, the average error on common north remains constant throughout the total number of epochs. This is expected as we do not store any information other than the current motion per epoch. All calculations are performed from scratch based on the new neighborhood after the 2-step motion. Thus, GCDL is free from incremental errors.

In this figure we can observe that the amount of noise on the angle and distance affect the common north error of the GCDL algorithm. The common north error is zero for environments free of error and increases linearly with the angle and distance errors. For low noise level, the common north errors are 10° & 2° for random & directed motion. For a high noise level, the errors become 34° & 14°. The common north errors are reasonable considering that we assume 1°/100 errors on angle measurements caused by node actuators for low & high noise levels.

IV. CONCLUSION

We propose two distributed algorithms to address the directional node localization problem in wireless sensor networks: GDL

Where each node has a digital compass, and GCDL which does not require a compass. Both of our directional localization algorithms enable nodes to coordinate their movement relative to their one-hop neighbors, and maintain a semi-rigid structure that results in a coherent movement of the swarm, during mobility, measurement errors or real-world disturbances tend to accumulate over time and affect the structure of the swarm, eventually disorganizing it. To avoid this undesirable effect, our algorithms perform localization in a few epochs of the node movement and work only with the data gathered within that time frame. We design our algorithms to work with local knowledge only, without the use of any global positioning infrastructure such as GPS, anchor points, and seed nodes.
REFERENCES