A New Splitting-Merging Paradigm for Distributed Localization in Wireless Sensor Networks

S. Alireza Motevallian, Lu Xia, Brian D.O Anderson

Abstract—This paper proposes a new merging (stitching) scheme for distributed localization of wireless sensor networks. In splitting-merging localization techniques, the network is first split into small sub-networks and each sub-network self-localizes itself, possibly in its own rather than a global coordinate basis. Then by using a stitching strategy they are merged back to compute the position of the nodes in a global coordinate basis. Unlike the existing techniques in which the stitchable sub-networks must have at least 3 nodes in common for the stitching to succeed, the proposed method can systematically address all possible scenarios (even when the two sub-networks are disjoint) while keeping the computational complexity fairly low. The scheme puts the idea of four-bar linkage mechanism and bilateration together in tackling the problem. Therefore, theoretically the proposed method can localize a broader class of networks which are localized by any splitting-stitching technique. Simulation comparisons with traditional trilateration and wheel graphs show a considerably higher percentage of the localized nodes in this technique. This technique in conjunction with some existing distributed splitting technique, provides a total distributed localization algorithm.

Index Terms—Sensor Networks, Distributed Localization, Merging-based localization, Bilateralation, Four-bar linkage

I. INTRODUCTION

The focus of this paper is on distributed range-based localization techniques. The network is modeled by its grounded graph where each vertex corresponds to a node and there is an edge between two vertices if the distance between the corresponding nodes is known. Generally the distributed localization techniques are categorized into two classes: (a) sequential localization techniques; (b) clustered localization techniques. In sequential localization techniques, at each step, all non-localized nodes with at least three neighbors with known positions, localize themselves by using the distance measurements to those neighbors. This is generally known as trilateration and the networks which can be localized by these techniques are called trilateration networks. Clustered localization techniques generally work in three steps: (1) (splitting) using some technique the network is split into some small clusters; (2) (local localization) each cluster is localized up to its local coordinate basis via a known localization technique; (3) (stitching or merging) the small clusters are stitched together to form the whole network by translating the location of the nodes into the global coordinate basis.

There are several clustered localization algorithms proposed in the literature ([5], [8], [6]). The main benefit of clustered localization techniques demonstrated was to limit the propagation of error while reducing the complexity of localization. Despite the difference between the localization approaches in these techniques, they all follow the same scheme when it comes to the stitching step: any pair of locally localized sub-networks can be stitched (or merged) together only when there are at least 3 nodes in common between them. This of course imposes a minimum density requirement on the network to ensure the success of the third step. Most of these techniques also require the 1-hop neighborhood of any node to be uniquely locally localizable. To ensure the 1-hop local localizability, as studied in [10], the underlying model of the network must be a wheel graph which is superior to trilateration model (Figure 1). However, they are not the only class of sub-networks that can be stitched together to form a single uniquely locally localizable sub-network. For example, consider the two sub-networks $G_1,G_2$ in Figure 1. Suppose that the nodes in $G_1$ are all localized up to the global coordinate basis, while evidently no node in $G_2$ has a wheel graph neighborhood which includes 3 nodes from $G_1$. Therefore, these graphs are marked as non-stitchable by wheel graph approach. However, as we will show later, all the nodes in $G_2$ are uniquely localizable.

In this paper, we propose a new stitching scheme that enables the localization of a broader class of networks. As is shown in [11], it is possible (in theory) to stitch two sub-networks with fewer than 3 vertices in common, given that there are enough edges joining them. By virtue of this result, the scheme proposed in this paper not only allows us to merge any pair of sub-networks with common vertices but also two disjoint sub-networks are joined with a sufficient number of edges between them.

Since in the existing clustered techniques in [5], [6], [8], each sub-network is effectively the 1-hop neighborhood of each node, the splitting step is trivial and efficient. However,
for other merging scenarios where there are less than three nodes in common between two sub-networks (e.g. when the sub-networks are all required to be disjoint), the splitting is no longer trivial. Despite the complexity of splitting a network into almost equally-sized sub-networks, there exist some fully distributed approximation techniques that can well split the network. To show the practicality of the proposed scheme we use the TASC technique ([9]) as it can split the network into disjoint sub-networks of almost equal sizes with the benefit of ensuring a minimum size for each cluster (unique localizability requires at least 4 nodes to prevent degenerate cases).

Section II introduces the background techniques, theories and notations we will be using throughout the paper. Section III gives a clear definition of the problem explaining the distinct cases that need to be addressed whereas Section IV provides merging solution for those cases. Section V and Section VI provide empirical evaluation of the technique as well as its implementation.

II. BACKGROUND

The main focus of the paper is on localization in a 2D, but wherever it is possible we provide a general definition valid for any $d$-dimensional ($d \in \{2, 3\}$) ambient space.

A. Network Abstraction and Global Rigidity

To model a network we first need to introduce the notion of frameworks. The \textit{grounded graph} of the network is a graph $G = (V, E)$, $n = |V|$ where each vertex $v \in V$ corresponds to a node and there is an edge $(v_1, v_2) \in E$ if the distance between the nodes corresponding to $v_1$ and $v_2$ is known. A \textit{configuration} $p = \{p_1, ..., p_n\}$ is a finite collection of $n$ labeled points in $\mathbb{R}^d$ where $d$ is the dimension $d \in \{2, 3\}$. A configuration is \textit{generic} if all $p_i$ are algebraically independent from each other. A \textit{framework} $F = (G, p)$ is a graph $G = (V, E)$ together with a corresponding configuration $p = \{p_1, ..., p_n\}$ in $\mathbb{R}^d$. In this framework, $p$ is the mapping $p : V \rightarrow \mathbb{R}^d$ where each vertex $v_i \in V, i = 1, ..., n$ is assumed to be located at a corresponding point $p_i \in p$. Sometimes this framework is called a realization of the graph $G$

In the context of network localization, a network of $n$ sensors therefore can be modeled by the framework $F = (G, p)$ where $G = (V, E)$ is the grounded graph of the network and $p$ is a mapping $p : V \rightarrow \mathbb{R}^d$. We use the terms framework and network interchangeably throughout the paper.

We say that two frameworks $F_1 = (G, p)$ and $F_2 = (G, q)$ are \textit{equivalent} (shown by $F_1 \equiv F_2$) if for all pairs $i, j$ where $(i, j) \in E$, $|p_i - p_j| = |q_i - q_j|$. Two configurations $p$ and $q$ are called congruent, denoted by $p \cong q$ if for all pairs $i, j \in V$, $|p_i - p_j| = |q_i - q_j|$. A framework $(G, p)$ is \textit{globally rigid} in $\mathbb{R}^d$ if $(G, p) \equiv (G, q)$ implies $p \cong q$ for any generic configuration $q$ with $|q| = |p|$. In other terms, a graph $G$ is said to be globally rigid if any set of its equivalent frameworks $(G, p)$ can be obtained by the translation, rotation or reflection of the whole framework.

A globally rigid networks is potentially locally localizable. In [3] (Theorem 1) it is stated that a network (of at least $d + 1$ nodes) modeled by the framework $(G, p)$ is uniquely localizable in $\mathbb{R}^d$ if and only if there are at least $d + 1$ anchors in it\(^1\) and its grounded graph $G$ is globally rigid. Anchors convert local localizability to localizability in reference to a global coordinate basis.

B. Merging Conditions

We now restrict attention to the case $d = 2$. Merging is the task of introducing a set of links between two locally localizable networks in order to form a single locally localizable network. This has been studied in detail in [2], [11]. In terms of global rigidity, this problem is equivalent to merging two globally rigid graphs by adding a set of edges so that the union is a single globally rigid graph. It is possible that the two networks share some nodes. However, as shown in [11], these are special cases of the situation where there is no common node. Therefore, from now on we assume that the sub-networks are node-disjoint unless explicitly specified (Subsection IV-C will later address all other cases including degenerate one).

The following important theorem discusses the necessary (from [11]) and sufficient (proved in [2]) conditions for a possible merging, in terms of global rigidity.

\textbf{Theorem 1.} Suppose that two disjoint globally rigid graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ with $|V_i| \geq 3$, are connected by a set of links, denoted by $L$. Then $G = G_1 \cup G_2 \cup G(L)$ is globally rigid if and only if (a) $|V_i \cap V(L)| \geq 3$, $i = 1, 2$ and $|L| \geq 4$, where $G(L)$ is the graph induced by the edge set $L$ and $V(L)$ is the set of end vertices of the edges in $L$.

This theorem only provides the conditions to ensure that the merging leads to a globally rigid graph but does not say how to actually accomplish the computations for merging-based localizations. We will address this by providing a distributed algorithm for merging two locally localizable sub-networks.

C. Bilateration and Trilateration

A network is a \textit{bilaterative} network if its grounded graph has a bilateration ordering. The class of trilaterative networks is a proper subset of the class of bilaterative networks.

\textbf{Definition 2.} A graph $G = (V, E)$ with $|V| \geq 3$ ($|V| \geq 4$) has a bilateration (trilateration) ordering if its vertices can be ordered as $v_1, ..., v_n$ ($n = |V|$) so that the subgraph induced by $v_1, v_2, v_3$ is a triangle and each $v_i, 3 \leq i \leq n$ ($4 \leq i \leq n$) is connected to at least two (three) vertices $v_j, j \leq i$ [4].

The \textit{bilateration operation} is the operation of localizing the position of a node up to a set of finite possibilities by considering its distances to at least two neighbors. This can be done by solving the equations obtained from these distance constraints (intersection of multiple circles each one around a neighbor) in addition to the position (global or relative) of those neighbors.

D. Four-bar linkage Mechanism

We will use the idea of four-bar linkage mechanisms (refer to [1] for details of the mechanism) later to address one of the merging scenarios which has been described in [4] as unsolved.

\(^1\)Anchors must be non-collinear in case $d = 2$, non-planar in case $d = 3$
by any existing bilateration-based localization technique. The four-bar linkage mechanism has been used in [7] for the first time in the context of localization. Figure 2 shows a typical four-bar linkage. Assume that the vertices A-E are joints and the edges between them are bars with fixed length. If we fix the position of the vertices A and B, and allow the other nodes to move freely in the plane, the position of the node D will follow a curve called a *coupler curve* [1], [7]. Assuming that joint A is the origin and joint B is on the positive side of x-axis, this curve can be explicitly expressed by the degree-six equation (1) where \((x, y)\) is the position of the vertex D [7]:

\[
a^2 ((x - k)^2 + y^2) (x^2 + y^2 + b^2 - r^2)^2 - 2ab ((x^2 + y^2 - kx) \cos \eta + ky \sin \eta) \\
\times (x^2 + y^2 + b^2 - r^2) ((x - k)^2 + y^2 + a^2 - R^2) \\
+ b^2 (x^2 + y^2) ((x - k)^2 + y^2 + a^2 - R^2)^2 \\
- 4a^2b^2 (x^2 + y^2 - kx)^2 \sin \eta - ky \cos \eta)^2 = 0
\]

As is mentioned in [1], the intersection of this equation with a circle will have a finite number of solutions (at most 12).

III. PROBLEM STATEMENT

Assume that we have a large network which has already been split into small sub-networks with the condition that each of them is uniquely localizable up to at least a local coordinate basis, i.e. the grounded graph of each sub-network is globally rigid. Also assume that each sub-network has at least 3 nodes. We are interested in a step-by-step merging process in which each sub-network is merged into one of its neighboring sub-networks to form a single post-merged locally localizable sub-network and this merging continues until the whole network is localized up to a single (possible the global) coordinate basis.

We will not discuss in this section the conditions required for a network to be splittable in the above manner. It is also fairly easy to see that trilateration networks can be treated by this technique as an special case where the merging sub-network is a single vertex. As the nodes are locally localized prior to the merging step, it is reasonable to assume that the distances between all pairs of nodes in any locally localized sub-network are known (we assume that every node in a sub-network knows the location information of the other members).

The main step of the process (merging two neighboring locally localizable sub-networks) can then be formally defined as follows:

Assume there are two sub-networks \(F_1 = (G_1, p_1)\) and \(F_2 = (G_2, p_2)\) where both have been locally (or maybe globally if there are at least three anchors in them) localized by an arbitrary localization algorithm with \(F_1\) localized in the reference basis and \(F_2\) localized in its local basis. Also assume that there are enough links (and/or common nodes) between \(F_1\) satisfying the conditions of Theorem 1. Apply a merging technique (detailed subsequently) to localize the nodes in \(F_2\) into the reference coordinate basis. Once the nodes involving in the merging are localized, the other nodes in \(F_2\) can be localized by simply computing the transformation matrix between the reference coordinate basis and the local basis of \(F_2\).

A. Possible configurations

According to Theorem 1, there must be at least four distance measurements between disjoint \(F_1\) and \(F_2\) in order for a merging algorithm to succeed. In the absence of noise, four distance measurements (\(|L| = 4\)) are enough to carry out the merging. If there are more links available, we just pick the best four and proceed with the merging. As it is necessary for each \(F_i\) to involve at least three vertices, there are 4 possible configurations if the \(F_i\) do not share a common vertex: (1) \(|V_1 \cap V(L)| = 3\); (2) \(|V_1 \cap V(L)| = 4\) and \(|V_2 \cap V(L)| = 3\); (3) \(|V_1 \cap V(L)| = 3\) and \(|V_2 \cap V(L)| = 4\); (4) \(|V_1 \cap V(L)| = 4\).

Allowing the \(F_i\) to share some nodes, results in three different scenarios detailed in [11]: (1) \(|V_1 \cap V_2| = 1\) and \(|L| = 2\); (2) \(|V_1 \cap V_2| = 2\) and \(|L| = 1\); (3) \(|V_1 \cap V_2| \geq 3\) and \(|L| = 0\).

IV. PROPOSED MERGING SCHEME

When \(|V_1 \cap V_2| = 0\), among the 4 possible configurations the first three can be addressed using bilateration while the last one requires four-bar linkage mechanism.

A. Bilateration-based cases

We explain the procedure for the case \(|V_1 \cap V(L)| = 3\) (Figure 3a) and skip the other cases as the method is the same. All the three nodes in \(F_1\) are already localized into the reference basis and acting as anchors. In this case there is always a node in \(F_2\) (\(n_3\) in Figure 3a) which is connected to exactly two nodes in \(F_1\) (\(n_1, n_2 \in V_1\)). Applying the bilateration operation (using these links) gives two possible positions for this node. Following the figure, the node \(n_4\) is connected to \(n_3\) and \(n_2\) and hence by further applying a bilateration operation, we obtain up to 4 possible positions for \(n_4\). Similar process produces 8 possible positions for \(n_5\). However, the distance between \(n_3\) and \(n_6\) is also known, and generic global rigidity of the whole network implies that only one of the 8 possibilities is consistent. This unique position of \(n_5\) is obtained from unique positions of \(n_4\) and \(n_3\) and hence we can identify the unique positions of all these nodes in \(F_2\).

B. Four-bar linkage case

Four-bar linkage mechanism is required for the case \(|V_1 \cap V(L)| = 4\) (Figure 3b) as there is no node in \(F_2\) with more than one neighbor in \(F_1\) and bilateration operation cannot be performed. Note that in the figure, the 4 vertices in each \(F_i\) form a complete graph as all the nodes in each \(F_i\) know their pair-wise distances. In [7], the authors solved a similar problem using the semi-definite programming (SDP) technique which requires a high computational capability an
can be obtained by mirroring \( K \). There is also another coupler curve, called \( C \). According to [1], the structure in Figure 4a (Figure 4b) leads to a coupler curve for node \( n_7 \) \( (n_8) \). In both cases, the nodes \( n_1, n_2 \) are the fixed bases in the four-bar linkage mechanism. Assume that \( n_1 \) is at the origin and \( n_2 \) lies on the positive side of the \( x \)-axis. The equation for the position of the node \( n_7 \), denoted by \( K(x, y) \), can be obtained by substituting \( k = l_1, r = l_1, R = l_2, a = l_6, b = l_5 \) in Equation 1. There is also another coupler curve, called \( K'(x, y) \), which can be obtained by mirroring \( K(x, y) \) against \( l_0 \) (\( K'(x, y) = K(x, -y) \)). The position of the node \( n_7 \) must also satisfy the circle equation \( C_3(x, y) : (x - x_3)^2 + (y - y_3)^2 = l_3^2 \) corresponding to link \( l_3 \) and the position of \( n_3 \).

Therefore the position of \( n_7 \) must be on the intersection of the two curves \( K(x, y) \) and \( C_3(x, y) \). The real intersections of the curves can be obtained easily by using any numerical technique (details are removed to cope with space limitation). As is explained in [1] the intersection of above equations can have at most six possible solutions (instead of 12). Intersecting \( K'(x, y) \) and \( C_3(x, y) \) also gives at most six other possible positions for \( n_7 \). Therefore, we may have a total of maximum 12 possible positions for \( n_7 \). We will denote this solution set by \( P = p_1, ..., p_12 \). By a similar procedure, we will obtain a maximum of 12 numerical values for the position of node \( n_8 \), denoted by \( Q = \{q_1, q_2, ..., q_{12} \} \). Since the distance \( l_0 \) between \( n_7 \) and \( n_8 \) is known, for each \( i \) \( \leq 12 \), we calculate \( ||p_i - q_j|| \) and compare it with \( l_0 \). If the configuration of the network is generic and the measurements are exact (no noise), then out of these \( s \times t \) possibilities for \( (i, j) \), there must be exactly one pair of positions for \( n_7 \) and \( n_8 \) consistent with the distance \( l_0 \). Therefore, the distance \( l_0 \) will resolve the unique position of nodes \( n_7 \) and \( n_8 \). If there is noise we simply choose the pair \( p_i, q_j \) which minimizes \( ||p_i - q_j|| - l_0 \). These unique positions can help to resolve the unique position of the nodes \( n_{i,j}, i = 5, 6 \) as well.

One issue with this technique is that there may be up to 144 different possible \((i, j)\) pairs which is high and the presence of any noise can lead to a wrong decision on correct positions. Therefore, it is important to see if we always get this worst-case in real scenarios. We studied the average number of solutions for each 4-bar linkage over several randomly generated networks in 2D. The unit square is split into two same-sized rectangles by a vertical line. Each side includes 4 uniformly randomly generate nodes representing \( F_i, i = 1, 2 \). Then, the coupler and circle equations are derived and solved to obtain the possible positions of \( n_7 \) and \( n_8 \). This whole process is repeated 50 times to get the average number of solutions.

According to the results, the average number of solutions for each 4-bar linkage is 4.56 (with \( \sigma = 2.21 \)). This states that the total number of possibilities for \((i, j)\) is expected to be around 20 which is far less than the theoretical worst case of 144 obtained earlier. It is also worth mentioning that in our simulations, we never encountered a case where 12 possible positions occurred for either \( n_7 \) or \( n_8 \) and the worst case encountered was 8.

C. Other cases

Figure 5a shows an example where one node \( (n_1) \) is shared between the \( F_i \). Since \( n_1 \) is in \( F_1 \), \( n_3 \) is connected to three vertices \( \{F_1, F_2, F_3\} \). Therefore, we can apply the bilateration operation to find 2 possible positions for \( n_3 \). This implies that there may only be two possible positions for \( n_3 \) and \( n_4 \) can be uniquely localized. If there is more than one vertex in common, the bilateration reduces to a simple trilateration (Figure 5b).

If there are less than three nodes in \( F_i \) (degenerate cases) the conditions of Theorem 1 are not satisfied. Without loss of generality, let us assume that \( |F_1| \geq 3 \) and \( |F_2| \leq 2 \). As is shown in [5], a necessary condition for a graph to be globally rigid is that the graph is 3-connected. This implies that \( |V_1 \cap V(L)| \geq 3 \) (at least three vertices in \( F_1 \) must be connected to the vertices in \( F_2 \)) or otherwise the the union of \( F_1 \) and \( F_2 \) is not globally rigid. If \( F_2 \) has one vertex, its position can be obtained by trilateration (\(|L| \geq 3 \)). Figure 5c shows the case where \( |V_2 \cap V(L)| = 2 \). Since the degree of each vertex \( n_i, i = 1, 2 \) in \( F_2 \) is 1, each \( n_i \) must also be connected to at least two vertices in \( F_1 \). This simply means that we can apply the bilateration operation starting from \( F_1 \) and localize the position of each \( n_i \). The unique position is then derivable by using the extra constraint between \( n_1, n_2 \) (Figure 5c).

V. PERFORMANCE EVALUATION

As mentioned above, most sequential localization techniques are based on trilateration while the clustered ones can be mainly modeled by wheel structures. To compare the benefits of the new stitching scheme, we compare it with trilateration and wheel. We generate networks of 100 uniformly distributed nodes in the unit square. We also randomly choose three neighboring nodes as the 3 anchors in the network. Figure 6 compares the percentage of the nodes marked as localizable by each scheme for different sensing radii. As can be seen the proposed merging scheme outperforms all existing techniques. For example when the sensing radius is 0.15, the proposed technique can localize around 70% of the nodes while both trilateration and wheel structures can only localize 40% percent.
graphs but they require central algorithms (Figure 7). Notice that there are still algorithms than bilateration which still can be localized by efficient algorithms exist. By introducing the four-bar linkage technique, we have proposed an even broader class of networks than bilateration which still can be localized by efficient algorithms (Figure 7). Notice that there are still algorithms that can localize all networks with globally rigid grounded graphs but they require **central** calculations.

VI. IMPLEMENTATION OF THE SCHEME

The proposed scheme in this paper is a template of three steps required in clustered localization techniques. This means that the user has a choice over the set of all possible splitting alternatives existing in the literature as well localization techniques for each sub-network. The scheme proposes different methods for the merging step based on the existing configuration. However, in order to propose an actual localization algorithm, we briefly introduce some techniques that can be used for splitting and localization steps (steps 1 and 2 in clustered localization). To perform the first step of the algorithm, we use the TASC splitting technique [9]. This algorithm splits the network into almost balanced clusters not only for isotropic networks but only with anisotropic ones which are more probable in real scenarios. This technique has several benefits: (1) it splits the networks into disjoint clusters where every node is within the 2-hop neighborhood of the cluster head (as desired by our stitching scheme); (2) the minimum size of each cluster can be guaranteed; (3) the number of nodes in each cluster is always small (which guarantees that each cluster can be localized efficiently).

Since TASC guarantees that the size of these clusters are small, most localization techniques can be used for the second step. For the localization of each sub-network, we propose the Sweeps algorithm mentioned in [4] (which can localize networks with bilateration orderings) as most small sub-networks resulted from the splitting phase, have a bilateration ordering and Sweeps is very efficient on these networks. For the stitching step we use the following algorithm based on the proposed merging scheme:

For each pair of neighboring (disjoint) clusters, they cooperatively decide if they can be merged into a single sub-network. This can be done in a distributed manner as all the proposed configurations in Section IV are within 2-hop neighborhood of the involving nodes (Figure 3). Then using either bilateration or 4-bar linkage mechanisms the networks merge themselves into a single sub-network. These steps repeat until no further change is possible.

VII. CONCLUSION AND FUTURE WORK

In this paper we proposed a new splitting stitching scheme for distributed localization of wireless sensor networks. Unlike the existing techniques in which any pair of split clusters must have at least 3 nodes in common in order to be merged, the proposed method uses used the bilateration and four-bar linkage mechanisms to address all mergeable scenarios (even when the clusters are disjoint) while keeping the computational complexity fairly low. Comparing to the underlying models of the existing splitting-merging techniques, namely wheel and trilateration structures, this scheme can localize a considerably higher percentage of the nodes when the network is sparse, a claim validated through simulations as well. To propose a total localization algorithm, we also provided a distributed splitting algorithm and a typical localization technique to localize the sub-networks. These techniques have existed for several years. However, this is the first time they are put together in the context of clustered localization techniques.

We are currently investigating the error propagation in this scheme and the actual implementation of the proposed localization algorithm. This involves further comparison with the accuracy of state-of-the-art techniques in presence of noise.

REFERENCES