ETOC: Obtaining Robustness in Component-based Localization

Xiaoping Wang[1] Yunhao Liu[2,3] Zheng Yang[2,3] Junliang Liu[2] Jun Luo[1]

[1]National University of Defense Technology, China

[2]Hong Kong University of Science and Technology, Hong Kong

[3]TNLIST, School of Software, Tsinghua University, China

xiaopingwang@nudt.edu.cn liu@cse.ust.hk yangzh@cse.ust.hk junll@cse.ust.hk junluo@nudt.edu.cn

Abstract — **Accurate localization is crucial for wireless ad-hoc and sensor networks. Among the localization schemes, component-based approaches specialize in localization performance, which can properly conquer network sparseness and anchor sparseness. However, such design is sensitive to measurement errors. Existing robust localization methods focus on eliminating the positioning error of a single node. Indeed, a single node has two dimensions of freedom in 2D space and only suffers from one type of transformation: translation. As a rigid 2D structure, a component suffers from three possible transformations: translation, rotation, and reflection. A high degree of freedom brings about complicated cases of error productions and difficulties on error controlling. This study is the first work addressing how to deal with ranging noises for component-based methods. By exploiting a set of robust patterns, we present an Error-TOlerant Component-based algorithm (ETOC) that not only inherits the high-performance characteristic of component-based methods, but also achieves robustness of the result. We evaluate ETOC through a real-world sensor network consisting of 120 TelosB motes as well as extensive large-scale simulations. Experiment results show that, comparing with the-state-of-the-art designs, ETOC can work properly in sparse networks and provide more accurate localization results.**

Keywords - component-based; localization; location ambiguity; robust localization; structural error tolerance

1. Introduction

Location awareness is highly critical for wireless ad-hoc and sensor networks. Due to the constraints on hardware cost and energy consumption, however, it is unfeasible to equip all nodes with positioning hardware (e.g., GPS receivers). Instead, only a few nodes are configured with location information in the network setup phase, called anchors. Other nodes then locate themselves by the inter-node distance measurements. We call this procedure network localization.

Recently, there are increasing literatures on network localization algorithms [1], falling into two categories: node-based design [2-4] and component-based design [5]. Node-based localization algorithms try to locate the entire network by individually locating each non-anchor node, which is also called sequential localization. Researches on this subject [6] show that the Sweeps algorithm [2] is able to localize all the sequentially localizable networks. That is, Sweeps achieves the utmost performance of the node-based design. Nevertheless, there is still a performance gap between Sweeps and the theoretical limit [7, 8]. Overall, node-based design suffers the inherent limitation that it requires each located node to be self-localizable, which is not necessary for network localization. To make it clearer, we illustrate the development of high-performance localization algorithms in Figure 1.

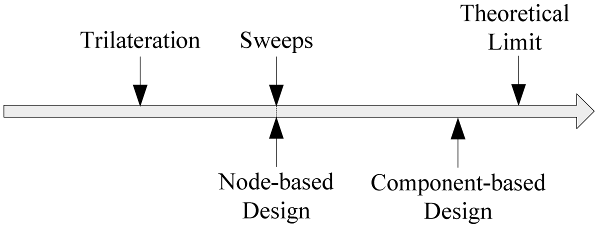


Fig. 1. Performance of localization algorithms

To narrow the gap of localization performance, component-based localization is proposed [5]. A component is defined as a set of nodes that forms a rigid structure. As components are rigid, they can be located as a whole. Hence, besides each single node, component is another basic unit for localization. After integrated into components, nodes can collaborate with each other, thus to facilitate network localization. As a result, component-based designs can locate more nodes, i.e. have higher performance, than the node-based ones, especially in sparse networks.

Existing component-based algorithms [5], however, are based on the idealized model that assumes accurate distance measurements between neighboring nodes. Nevertheless, measurement errors are inevitable in practice. Based on the noisy measurements, the localization result may suffer structural deformation, which locates a component by an incorrect embedding in the physical coordinate system. Once a structural deformation is triggered, the error of the result is not determined by the measurement errors, but by the moved distances caused

978-1-4244-8645-8/10/$26.00 ©2010 IEEE

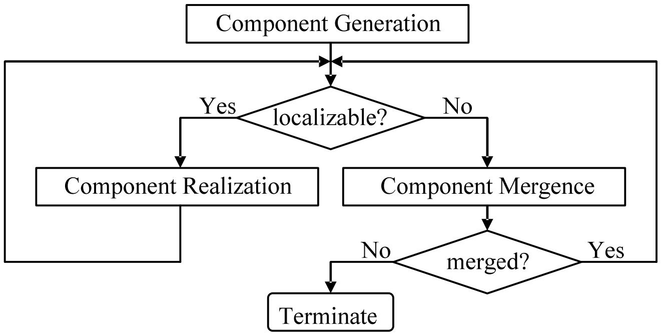


Fig. 2. The flowchart of component-based algorithms

by the deformation. Hence, the structural deformation may lead to huge errors in the result set.

Unfortunately, existing designs can only achieve robustness in node-based localization schemes [3, 9]. As we know, a single node has two dimensions of freedom in 2D space (e.g., X-Y coordinates) and only suffers from one type of transformation: translation. In contrast, a component suffers from three possible transformations: translation, rotation, and reflection. A high level of freedom obviously brings about complicated cases of error producing and difficulties on error controlling. For component-based approaches, to the best of our knowledge, there is no related work on how to deal with measurement noises before this study.

To conquer such difficulties, we analyze how noisy ranging destroys the robustness of component realization. To quantify the impact of noises, an error tolerance for a component is defined as the upper bound of ranging errors under which the component can be localized uniquely (without deformation). By the concept of error tolerance, we present an Error-TOlerant Component-based algorithm (ETOC) that can achieve guaranteed robustness of the localization result. In addition, as a component-based design, ETOC inherits the property of high performance, where performance refers to the number of successfully localized nodes out of all non-anchor nodes. We evaluate ETOC through a real-world sensor network. The network consists of 120 TelosB motes distributed in an area of 126×145m^2. Experimental results show that ETOC can localize the network entirely in spite of the network sparseness. Extensive simulations also show that, comparing with the state-of -the-art design, Robust Quadrilaterals [9], ETOC succeeds in both localization performance and error control.

The rest of this paper is organized as follows. We present the preliminary knowledge in Section 2. In Section 3, we introduce the robust patterns for ETOC. Experimental studies are presented in Section 4. We introduce related work in section 5. Finally, we conclude the work in Section 6.

2. Preliminary

In this section, we briefly introduce the basic idea of the component-based localization algorithm. Then, we analyze the key issue of achieving robustness in component-based scheme.

2.1 Terminology

We use a *distance graph* $G=(V, E)$ to present a given network, in which each vertex in V denotes a node in the wireless network and each edge $(i, j)\in E$ denotes a distance measurement between node pair (i, j). Associated with each edge, we use a function $d(i,j){:}E\rightarrow R$ to denote the distance value, also abbreviated as d_{ij}. We suppose a small portion of nodes, called *anchors*, know their locations in advance. Without loss of generality, m anchors are labeled from 1 to m, together with n-m ordinary nodes labeled from m+1 to n, where n denotes the total number of nodes in the network. The ground truth position of each node is denoted by p_i, $1\leq i\leq n$. We uniformly use node to present a wireless device and use edge to present the distance measurement of two nodes. We also use the terms in graph theory to present certain special topology in the distance graph. For example, a triangle means three nodes can measure distances to each other.

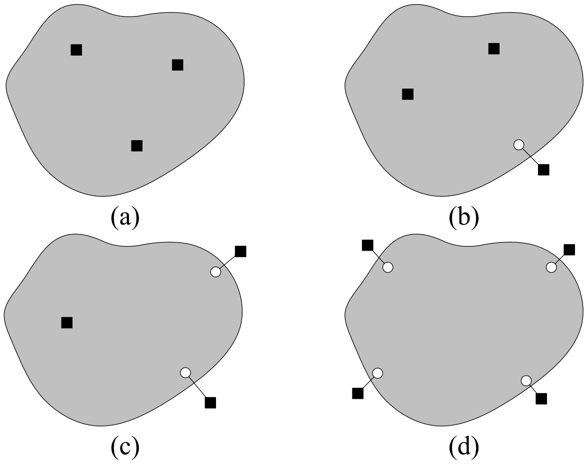


Fig. 3. Realization patterns of components

2.2 Overview of Component-based Localization

Component-based localization algorithm contains three main procedures: component generation, component mergence, and component realization. We show the flowchart of the localization process in Figure 2. The details of each step are as follows:

1. Component generation partitions the network into a set of components. A component is initialized by a triangle in the network for constructing a local coordinate system. Then, other nodes join the newly generated component by trilateration.

2. Component realization localizes a component as a whole, which converts the local coordinate system of a component to that of the physical coordinate system.

3. Component mergence stitches two components to generate a larger component. Merging components is to covert the local coordinate system of one component to that of the other one.

As component mergence and component realization are both based on converting the local coordinate system, the key of component-based localization is the coordinate system conversion. In contrast to the traditional local maps [9, 10], components convert their local coordinate systems by both the common nodes and the inter-coordinate-system measurements. We take component realization as an example, where the target coordinate system is the physical coordinate system defined by the anchors. We first differentiate the role of anchors. If an

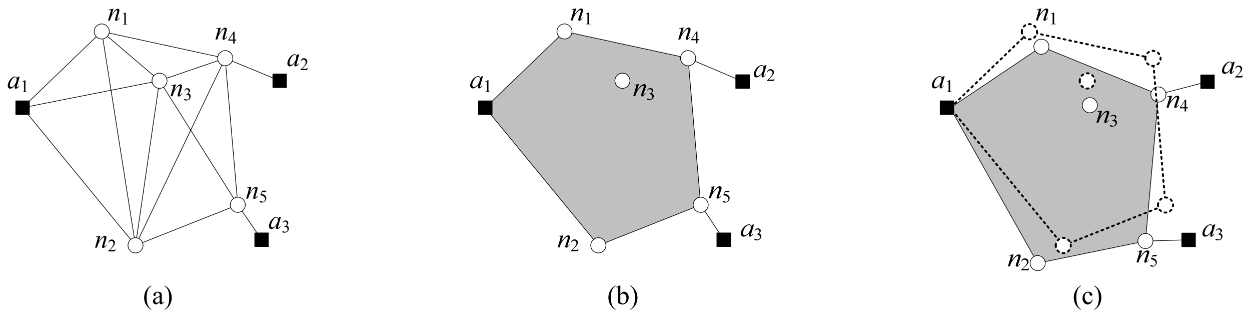


Fig. 4. Component-based localization

anchor belongs to the target component, we say the anchor is an *internal anchor*; otherwise, it is an *external anchor*. By enumerating the number of internal anchors, we conclude the complete patterns for component realization. All patterns are illustrated in Figure 3, where the shaded area denotes a component, the solid squares denote anchors, the soft circles denote non-anchor nodes, and edges denote distance measurements between nodes. If there is an edge connecting an in-component node and an external anchor, we say the component has an edge linked to an anchor. Then, we list all the patterns as follows:

(a) the target component has three or more internal anchors;

(b) the target component has two internal anchors and at least one edge linked to an anchor;

(c) the target component has one internal anchor and at least two edges linked to two distinct anchors;

(d) the target component has four edges linked to at least three distinct anchors, where the number nodes associated with these edges is at least three in the component.

Once a component matches one of the patterns, we can convert the local coordinate system by solving simultaneous polynomial equations [5].

Figure 4 demonstrates component-based localization. Figure 4(a) shows the distance graph of the network. Under the view of each single node, none of them can be localized, because each node has only one neighboring anchor. In contrast to locating each single node, we can integrate the nodes into a component, as shown by the shaded area in Figure 4(b). By pattern(c), all-non-anchor nodes can be located simultaneously.

2.3 Ambiguity Issue in Component-based Localization

When the distance measurements are noisy, the two coordinate systems may not be precisely aligned. Such misalignment may produce ambiguities in localization result. Clearly, if we locate the component to the ground truth location in the physical coordinate system, the embedding will accept the measurement errors, where the embedded distance falls in an acceptable range of the measured distance. However, there may also exist alternative embeddings that match the embedded distances with the measured distances.

For the example shown in Figure 4(a), if the measured distance between a_2 and n_4 is a bit shorter than the accurate value, the embedding of the component will be "rotated" to the place shown in Figure 4(c). We also plot the ground truth embedding of the component by the dashed lines and circles in this figure. If we evaluate the embedding by the discrepancy between the embedded distance and the measured distance, the rotated embedding matches the measured distance better and become the final result. Comparing the two embeddings, we observe huge errors for all located nodes. In a word, if there exist multiple embeddings for a component that accept all of the noisy distance measurements, we say the localization result of the component is *ambiguous*. Hence, we define a localization algorithm is *robust*, if it can properly avoid the ambiguities.

3. Robust Coordinate System Conversion Patterns

In this section, we exploit robust patterns for the coordinate system conversion without ambiguities. We also use component realization to show the robust coordinate system conversion patterns, for the sake of better differentiating the source and target coordinate system. Clearly, these robust patterns can be directly used for component mergence. As shown in Figure 3, there are only four patterns to do the conversion, so that we discuss each pattern in each of the following sub-sections. In addition, we assume that the in-component positions of the nodes are computed by robust node-based algorithms [3, 9] without ambiguities.

3.1 Robust Realization with Three or More Internal Anchors

As three nodes are enough to determine a coordinate system, the conversion can accomplished by purely using the alignment of the internal anchors. To uniformly number the patterns, we rewrite this pattern as follow. Case 1: the component contains three or more internal anchors.

The in-component coordinates of the anchors may contain errors because of the noisy distance measurements. Hence, it is unlikely that the anchors will precisely align in both of the coordinate systems. For this issue, there is a closed-form and least-square optimal method, called coordinate system registration [11].

Let $x_{l,i}$ and $x_{p,i}$ be the known positions of k anchors $i = 1, \dots, k$, $k \geq 3$ in the local coordinate system and the physical coordinate system, respectively. The goal of registration is to find a translation vector t and rotation matrix (with possible reflection) R that transform a point x_l in the local coordinate system to the equivalent point x_p in the physical coordinate system using the formula: $x_p = Rx_l + t$.

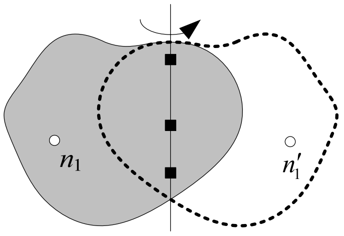


Fig. 5. Global flip

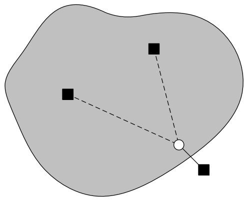


Fig. 6. Illustration of Case 2

By the squared error, this problem is solved by minimizing the alignment error of the anchor nodes:

$$(R,t) = \arg\min_{R,t} \sum_{i=1}^{k} \| x_{p,i} - Rx_{l,i} - t \|^2 .$$

Define the centroids of $x_{l,i}$ and $x_{p,i}$:

$$\bar{x}_l = \sum_{i=1}^{k} x_{l,i} \quad \bar{x}_p = \sum_{i=1}^{k} x_{p,i} .$$

Then, compute the matrix:

$$M = \sum_{i=1}^{k} (x_{p,i} - \bar{x}_p)(x_{l,i} - \bar{x}_l)^T .$$

The result of the optimization problem is expressed as [11]:

$$\begin{cases} R = M((M^T M)^{1/2})^{-1} \\ t = \bar{x}_p - R\bar{x}_l \end{cases} .$$

Nevertheless, this conversion only considers the alignment of the anchors. When the anchors are approximately collinear, the conversion may flip the non-anchor nodes to incorrect positions. As shown in Figure 5, if the result is flipped, node n_1 will be localized to the position n_1', which is far from its ground truth position. To avoid this problem, ETOC requires that width of the anchors must be large enough according to the measurement errors. When the requirement is not satisfied, ETOC will try to realize this component through other patterns.

3.2 Robust Realization with Two Internal Anchors

Two internal anchors are not enough to determine a coordinate system, because the coordinate system may flip against the axis of the two anchors. Hence, we need at least one additional constraint to make this problem solvable, described as follow. Case 2: the component contains two anchors and at least one edge linked to an external anchor.

As shown in Figure 6, for each node linked to an external anchor, it can obtain at least three distance estimates with anchors: one is the direct distance measurement with the external anchor and the other two are the in-component distances to the internal anchors. Then, the physical location of the node can be computed by trilateration. Then, we obtain at least three nodes know their physical locations, so that we can convert the coordinate system by following Case 1.

3.3 Robust Realization with One Internal Anchor

When a component contains only one anchor, we cannot directly adopt the coordinate system registration. In this section, we present a novel robust mechanism to address this problem.

The internal anchor can only prevent translation of the component. To make the coordinate system conversion problem solvable, two edges (distance measurements) are required to eliminate possible rotation and reflection, described as follow. Case 3: the component contains one anchor and two distinct non-anchor nodes connecting with two distinct anchors.

As illustrated in Figure 7, the shaded area denotes a component C, and it contains an anchor a_1 and two nodes n_1, n_2 sharing two edges with two external anchors a_2, a_3. Let α, θ, and φ denote the angle values of $\angle a_2a_1a_3$, $\angle n_1a_1a_2$, and $\angle n_2a_1a_3$, respectively. Since all the distances are known, we can compute the values of these angles from $\Delta a_2a_1a_3$, $\Delta n_1a_1a_2$, and $\Delta n_2a_1a_3$, respectively. Define $S=\{\alpha+\theta+\varphi, \alpha+\theta-\varphi, \alpha-\theta+\varphi, \alpha-\theta-\varphi\}$. Let δ denote $\min\{|\cos\beta_1 - \cos\beta_2|\}$ for all $\beta_1, \beta_2 \in S$ and l_1, l_2 denote the in-component distances of node pair (a_1,n_1), (a_1,n_2), respectively. Then, the upper bound of ranging errors under which the component can be realized uniquely (defined as the error tolerance of the component) can be expressed as:

$$T_C = \frac{1}{2}\frac{l_1 l_2}{l_1 + l_2}\delta .$$

We leave the proof of this proposition in Appendix. By comparing error tolerance T_C with the measurement error, the worst-case probability of ambiguities is bounded. If the error is lower than T_C, ETOC can realize this component robustly.

ETOC converts the coordinate system by the following two steps. First, ETOC computes the physical locations of node n_1 and node n_2. As shown in Figure 7, by the distance constraints, node n_1 may flip against axis a_1a_2, thus to have two candidate locations denoted by n_1 and n_1' in the figure. Similarly, node n_2 may flip against axis a_1a_3, thus to have two candidate locations denoted by n_2 and n_2'. Hence, the number of candidate distances between nodes n_1 and n_2 is four, i.e. the distances between the location pairs (n_1, n_2), (n_1, n_2'), (n_1', n_2), and (n_1', n_2'). Then, ETOC selects the candidate distance with minimum difference from the in-component distance between nodes n_1 and n_2. Note that each candidate distance uniquely determines a location pair of nodes n_1 and n_2. As a result, nodes n_1 and n_2 are located simultaneously. Together with anchor a_1, we obtain three nodes that know their physical coordinates. Then, following Case 1, ETOC converts the coordinate system.

3.4 Robust Realization with No Internal Anchor

When the distance measurements are accurate, four interconnected edges are able to form a system of overdetermined simultaneous equations for converting a coordinate system [5]. However, such mechanism is sensitive to the measurement

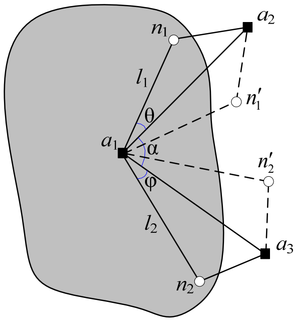
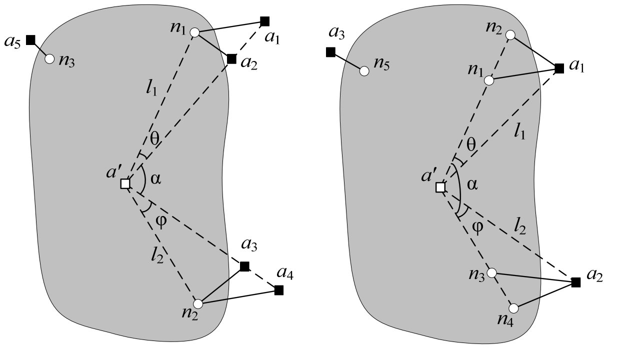


Fig. 7. Illustration of Case 3



(a) (b)

Fig. 8. Illustration of Case 4

errors. To avoid the ambiguities, we solve this case by five interconnected edges, which is fairly close to the optimal result. Case 4: there are five edges connecting the component with anchors forming one of the structures shown in Figure 8.

As shown in Figure 8(a), there are three distinct nodes n_1, n_2, and n_3 in the component, where node n_1 and node n_2 share two edges with two external anchors respectively, and node n_3 shares an edge with another external anchor. We label the anchors by a_1, a_2, a_3, a_4, and a_5. Though we use different notations for these anchors, we only demand that they map to at least three distinct anchors. We draw two lines through a_1, a_2, and a_3, a_4, and they intersect at the point a'. Then, the physical location of point a' is known. As shown in Figure 8(a), the angle value of α can be computed from $\Delta a'a_1a_4$. Further, as $\Delta n_1a_1a_2$ and length $a'a_2$ are known, we can compute angle value θ and distance l_1 from $\Delta a'n_1a_1$. Similarly, we can compute the angle value φ and distance l_2 from $\Delta a'n_2a_4$. As we already obtain the angle values of α, θ, and φ as well as the distances of l_1 and l_2, we can evaluate the robustness of this structure by error tolerance. If it is robust, we get the physical positions of node n_1 and node n_2 by Case 3. Taking nodes n_1 and n_2 as two anchors, we can convert the coordinate system by the case with two internal anchors, i.e. Case 2. According to case 2, an additional edge (n_3, a_5) is required. Finally, after computing the location of node n_3, the whole component is located.

As coordinate system conversion is symmetric, this method can also work when we exchange anchors and the internal nodes, as shown in Figure 8(b). In addition, there is a special case that the two lines a_1a_2 and a_3a_4 are parallel (i.e. the point a' does not exist). Then, we can testify that the error tolerance of this structure is zero. Hence, this special case is non-robust.

Further, we conclude the completeness of ETOC. Compared with the complete pattern set shown in Figure 3, ETOC completely solves three out of four patterns. Specifically, cases 1-3 of ETOC solve patterns (a)-(c), respectively. Case 4 in ETOC partially solves pattern (d). Case 4 requires five edges, while the optimal result of pattern (d) requires four edges. In a word, ETOC can provide robustness for most patterns of original component-based localization. Finally, we show the details of ETOC implementation in Algorithm 1.

Though we describe ETOC in the centralized way for clarity, ETOC can be implemented in a distributed way by geographic hash table (GHT) [12]. Using GHT, a component can integrate the distance measurements with another component (or the anchors) to a node at a specified in-component position. Thus, the specified node can determine whether the component can be localized or merged with another component.

***Algorithm 1** ETOC*

```
Input: the ranging graph G, the corresponding measured distance of each edge, the anchor node set Anchors, and corresponding physical positions of anchors.
Output: the localized node set and the position estimates
/* Invoke component generation process to partition G into components and isolated nodes. */
C = GenerateComponents(G).
/* Set anchors as localized nodes*/
Localized = Anchors
while $C \neq \emptyset$ do
    /*Find and localize the components or isolated nodes*/
    C = RobustRealizeComponents(C)
    /*Merge the components or isolated nodes*/
    C = RobustMergeComponents(C)
    if C does not change in this loop then
        break
return Localized and the corresponding positions
RobustConvert(local, target)
switch pattern of
    Case 1:
        if the width of the aligning nodes in the local coordinate system or in the target coordinate system is below the threshold then
            return FALSE
        else
            Compute the conversion tuple (R, t)
            break
    Case 2:
        if the trilateration is non-robust then
            return FALSE
        else
            Compute the position of the node in the target coordinate system by trilateration
            goto Case 1
```

```
Case 3:
    if the error tolerance is below the maximum error then
        return FALSE
    else
        Compute the positions of the two nodes in the target coordinate system
        goto Case 1
Case 4:
    if the error tolerance is below the maximum error then
        return FALSE
    else
        Compute the positions of the two nodes in the target coordinate system
        goto Case 2
/* When it passes all test, the result is robust */
return TRUE and the conversion tuple ($(R, t)$)
GenerateComponents($G$)
$C = \varnothing$
while $G$ contains unchecked triangles $t$ do
    if the width of $t$ is beyond the threshold then
        /* Create the component $c_t$ */
        Construct a local coordinate system from the distance measurements of $t$
        while existing a node $n$ that can perform trilateration with component $c_t$ do
            if the trilateration is robust then
                $c_t = c_t \cup n$
        $G = G \backslash c_t$
        $C = C \cup c_t$
/* The remainder nodes in $G$ are isolated nodes */
  $C = C \cup G$
return $C$
RobustRealizeComponents($c$)
while existing an isolated node $n$ that can process trilateration to nodes in Localized do
    if the trilateration is robust then
        $C = C \backslash n$
        $Localized = Localized \cup n$
while existing a component $c$ that matches the realization patterns do
    if RobustConvert($c$, Anchors) then
        $C = C \backslash c$
        $Localized = Localized \cup c$
return $C$
RobustMergeComponents($C$)
while existing an isolated node $n$ that can process trilateration to a component $c$ do
    if the trilateration is robust then
        $C = C \backslash c$, $C = C \backslash n$, $c = c \cup n$
        $C = C \cup c$
while existing two components $c_1$ and $c_2$ that match the patterns do
    if RobustConvert($c_1$, $c_2$) then
        $c = c_1 \cup c_2$, $C = C \backslash \{c_1, c_2\}$
        $C = C \cup c$
return $C$
```

4. Experiments

We evaluate ETOC by both a real-world system and extensive simulations.

4.1 Experiment Setup

To validate the robustness and the performance, we evaluate ETOC on a real-world sensor network for forest monitoring (the GreenOrbs project [13, 14]). Currently, the network consists of 120 TelosB nodes that are deployed in an area of 126×145m^2. Based on the Time of Arrival (ToA) of acoustic signals, the inter-node distances are measured with maximum error of 0.3m. The average degree of the distance graph is 7.2. That is, the network is sparse for localization, considering the fact that trilateration requires the average degree to be beyond 10 for entirely locating a network [9]. Anchor nodes are manually configured on the boundary of the deployed region. We use this network to show the performance of ETOC, i.e. conquering network sparseness and anchor sparseness. Further, we compared the localization result with that of BCALL [5] to show the accuracy of ETOC.

Besides the real-world system test, we also conduct extensive simulations to evaluate ETOC under a wide range of parameters. The scenario is a square normalized region $[0,1]^2$ with randomly distributed 200 nodes, and we select a certain percentage of the nodes as anchors. We adopt distance measurement range r to control the density of the network. The distance information between neighboring nodes is corrupted by zero mean additive Gaussian noise [3], $N(0, \sigma^2)$. We take three deviations, i.e. 3σ, as the maximum error, so that the threshold for determining robustness is also 3σ. For each set of simulations, we take multiple runs and report the average.

We use three metrics in our simulations: performance, accuracy, and cost. The proportion of robustly located nodes out of all non-anchor nodes shows the performance of each algorithm. The standardized position estimate error (SPEE) indicates the localization accuracy of each algorithm, defined as the ratio of the mean position estimate error to the measurement range:

$$SPEE = \frac{1}{nr}\sum_{i=1}^{n} \| p_i - \hat{p}_i \| \times 100\%$$

where n is the total number of located nodes, r is the range of distance measurement, p_i and $\hat{p}_i$ are the ground truth and estimated position of node i, respectively. We evaluate the cost by the mean number of trilaterations (a.k.a. multilaterations) needed for locating a node. When no nodes are localized from a network instance, i.e. n=0, we define the SPEE and cost to be not-a-number and do not count these data in the final statistic. Moreover, we conduct the experiments by controlling the following parameters: the standard deviation of the ranging errors and the mean degree of the network.

We compare ETOC with the state-of-the-art design, robust quadrilaterals (RQ) [9], in the simulations. RQ is a node-based design for guaranteed robustness, i.e. structural uniqueness of the localization result. The basic localization unit of RQ is four-node local maps. RQ sets a bound on the geometric element to avoid flip ambiguity, which is the only structural

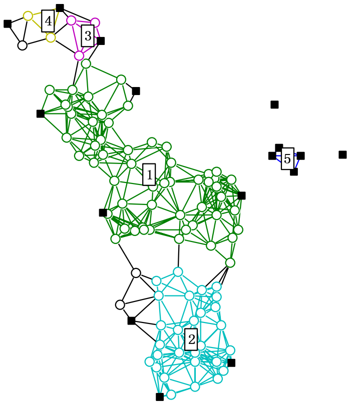


Fig. 9. Distance graph of the network

deformation of such local maps. Consequently, the result of RQ is proved to be robust under noisy distance measurements. We implement full RQ algorithm with cluster optimization to mitigate error accumulation.

4.2 Real-World System Test

We compare ETOC with an extended version of BCALL [5]. For BCALL, we first finitely localize the components by solving simultaneous polynomial equations. Then, we select the candidate result with minimum stress, which is defined as the squared discrepancy between the localized inter-node distances and the measured distances. As they are both component based algorithms, they follow the same execution procedure. Figure 9 shows the state of the network after component generation, in which the solid squares denote anchors and the soft circles denote non-anchor nodes. The details of each step are as follows:

1. After component generation step, this network is partitioned into five components and eight isolated nodes, as shown in Figure 9.

2. Then, we use the patterns of ETOC and BCALL to realize the components. First, component 1 can be localized through Case 1 (Pattern (a)), and component 2 can be localized through Case 2 (Pattern (b)).

3. After component 1 is localized, component 3 can be localized through Case 3 (Pattern (c)).

4. Next, component 4 is localized through Case 3 (Pattern (c)).

5. Finally, all isolated nodes can be localized by trilateration.

We show the average error of each component and the isolated nodes (the last column) in Figure 10. The error is standardized to be the percentage of the maximum distance measurement range. From this figure, we conclude that ETOC achieves lower localization error than BCALL. Further, the error for each component is quite stable, while BCALL will lead to error accumulation. Clearly, ETOC performs better in error control. Then, we conduct extensive simulations to evaluate ETOC under various configurations.

4.3 The Impact of Measurement Errors

In this section, we evaluate performance, accuracy and cost of ETOC, when the standard deviation of the errors varies. We fix the anchor proportion to 10% and set the average degree to be 25. The maximum proportion of error in distance measurement varies from 0% to 20% with step length 1%. We report the mean result of 50 network instances in Figure 11.

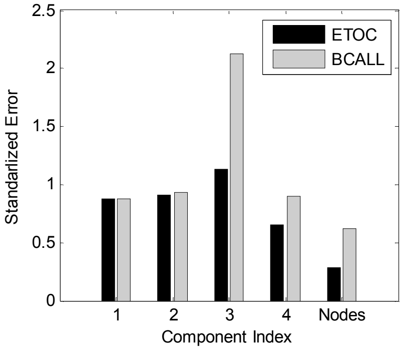


Fig. 10. The average error for each component

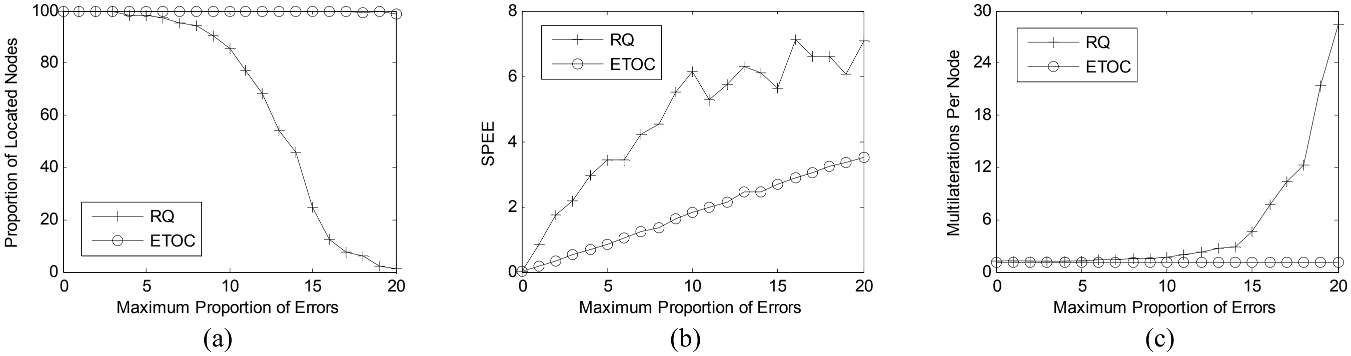
Figure 11(a) plots the proportion of successfully located nodes against the error magnitude. The performance of ETOC decreases slightly with the increase of error magnitude, because the increased errors may make some components fail to pass the robust test. In contrast, the performance of RQ decreases sharply when the error magnitude exceeds 15%. RQ relies on generating uniformly overlapped local maps to produce a global map. With the increase of ranging errors, the robust test of RQ will drop more local maps, so that the performance decreases sharply when RQ fails to generate adequate overlapped local maps.

Figure 11(b) plots the SPEE against error magnitude. The SPEE of each algorithm is approximately linear with the ranging errors. As we do not include the instances that no node is localized in the final statistics, the valid instances for RQ become less with the increase of error magnitude. Hence, the SPEE of RQ fluctuates more when the maximum error exceeds 10%. Over all the tested range, the SPEE of RQ is always higher than that of ETOC. RQ always uses the four-node local maps as basic localization units. Hence, nodes can only use three distance measurements at a time, no matter how many neighbors are available. In contrary, ETOC takes advantage of all measurements with neighboring nodes. Using more measurements can clearly diminish the error of the result.

Figure 11(c) shows the number of trilaterations needed to locate a node for each algorithm. The cost of ETOC is near optimal over the tested range. In contrast, the cost of RQ first increases slightly for less than 15% errors. Then, the cost increases sharply when the error further enlarges. Both of the algorithms need to generate all components or local maps in the generation step, no matter whether they are finally localized or not. As a result, the total cost is approximately a constant for each algorithm. Consequently, the average cost is reversely proportional to the number of successfully localized nodes.

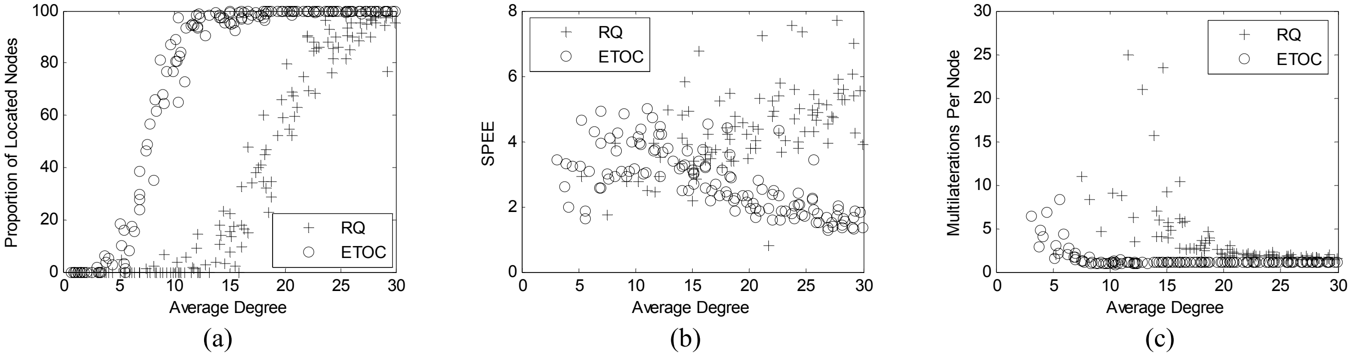
4.4 The Impact of Network Density

In this section, we evaluate the performance, accuracy and cost of ETOC, when the network density varies. We fix the anchor proportion to 10% and set the errors to be at most 10% of



(a) (b) (c)

Fig. 11. The impact of error magnitude



(a) (b) (c)

Fig. 12. The impact of average degree

distance measurement values. We adopt an empirical formula to control the average degree of the network instances linearly. The average degree of each network instance varies from 0 to 30 with step length about 0.2.

Figure 12(a) plots the proportion of robustly localized nodes against average degree. When the network density enlarges, both of the algorithms locate more nodes. ETOC can locate the entire network when the average degree is beyond 10. Being a component-based algorithm, ETOC inherits the characteristic of high performance. Component-based localization can integrate information on the granularity of components, thus can work well with low network density as well as low anchor density. In contrast, RQ demands the average degree to be over 25 for entirely localizing a network. Indeed, to guarantee robustness, RQ drops a large proportion of local maps in the local-map generation step, so that RQ needs high network density to compensate this.

Figure 12(b) plots the SPEE against average degree. The SPEE of ETOC decreases when the average degree increases. In contrast, the SPEE of RQ keeps in the same level over all the tested range. RQ can hardly benefit from the increase of average degree, because RQ can only use three distance measurements at a time when generating a local map. In contrary, ETOC can utilize all measurements with neighboring nodes. Using more measurements can clearly diminish the error of the result, especially when the average degree is high.

Figure 12(c) plots the mean number of trilaterations needed for locating a node by each algorithm. With the increase of network density, the cost of each algorithm decreases. As we have discussed, the mean cost is reversely proportional to the total number of successfully localized nodes. The cost of RQ is higher than that of ETOC, because ETOC locates more nodes than RQ does.

5. Related Work

Localization in wireless networks has been attracted significant research interest. Many researchers model the localization problem as a weighted graph realization problem and employ rigidity theory to analyze the corresponding problems [2, 7, 15-18]. This model is also widely used for theoretical research [19] and algorithm design [2, 5]. These researches provide valuable insight in localization problem. Nevertheless, this model assumes that the distance measurements between neighboring nodes are accurate, which is over-idealized for current ranging techniques [20]. Compared with these designs, ETOC guarantees the robustness of the result even under noisy distance measurements, which is more practical and realistic.

To handle the noisy ranging measurements, many researchers propose to minimize the impact of ranging errors in localization [21-29]. Moore et al. introduce the concept of Robust Quadrilaterals to avoid flip ambiguity [9]. They point out that flip ambiguity is the only structural deformation for locating a single node, so that the result is proved to be robust. However, the experimental results show that this design suffers low performance, working properly only in dense networks. In contrast, ETOC inherits the high localization performance from component based localization.

Basu et al. propose to locate wireless networks with both distance and angle information [30]. They relax the related problem to a convex form and solve it by linear programming. This design can provide a clear region estimate for each localized node, thus can properly avoid possible structural deformation. However, this design relies on the knowledge of both distance and angle measurements, which is not always available in practice.

Liu et al. [3] and Yang et al. [4] propose to enhance traditional trilateration with error management. They track the error in each step to minimize the expected error of the final localization result. Their methods perform well on diminishing the overall localization error of the result. However, they cannot provide any robustness guarantee. As structural deformation is topology-sensitive, it can be triggered by any tiny errors in special case. Hence, purely tracking errors cannot completely avoid potential structural deformations.

In addition, some researchers use the Cramer-Rao lower bound (CRLB) to characterize the error of network localization [31-34]. CRLB provides a lower bound on the variance achievable of an unbiased location estimator [33]. The same as error management, purely investigating the errors cannot guarantee robustness of the localization result.

6. Conclusions

We propose the concept of error tolerance for component based approaches. By exploiting a set of patterns, we design a robust localization algorithm, ETOC, which is the first work to address ranging noises for component based localization. Compared with existing works, ETOC obtains higher localization performance and better error control. We evaluate ETOC through a real-world system and extensive simulations. The experimental results show that ETOC works properly in sparse networks and achieves more accurate results.

Acknowledgment

This work is supported in part by NSFC/RGC Joint Research Scheme N_HKUST602/08, National Basic Research Program of China (973 Program) under Grants No. 2011CB302705 and No. 2010CB328004, and NSF China 60970118 and 60903224.

References

[1] Y. Liu, Z. Yang, X. Wang, L. Jian, Location, Localization, and Localizability, *Journal of Computer Science and Technology (JCST),* vol. 25(2), pp. 274-297, 2010.

[2] D. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. S. Morse, Y. R. Yang, Localization in sparse networks using Sweeps, in *ACM MobiCom*, 2006.

[3] J. Liu, Y. Zhang, F. Zhao, Robust Distributed Node Localization with Error Management, in *ACM MobiHoc*, 2006.

[4] Z. Yang, Y. Liu, Quality of Trilateration: Confidence-Based Iterative Localization, *IEEE Transactions on Parallel and Distributed Systems (TPDS),* vol. 21, pp. 631 - 640 2010.

[5] X. Wang, J. Luo, S. Li, D. Dong, W. Cheng, Component Based Localization in Sparse Wireless Ad Hoc and Sensor Networks, in *IEEE ICNP*, 2008, pp. 288-297.

[6] J. Fang, M. Cao, A. S. Morse, B. Anderson, Sequential Localization of Sensor Networks, *SIAM Journal on Control and Optimization,* vol. 48, pp. 321–350, 2009.

[7] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Yang Richard Yang, B. D. O. Anderson, P. N. Belhumeur, A Theory of Network Localization, *IEEE Transactions on Mobile Computing,* vol. 5(12), pp. 1-15, 2006.

[8] Z. Yang, Y. Liu, Understanding Node Localizability of Wireless Ad-hoc Networks, in *IEEE INFOCOM*, 2010.

[9] D. Moore, J. Leonard, D. Rus, S. J. Teller, Robust distributed network localization with noisy range measurements, in *ACM SenSys*, 2004.

[10] Y. Shang, W. Ruml, Improved MDS-based localization, in *IEEE INFOCOM*, 2004.

[11] B. K. P. Horn, H. M. Hilden, S. Negahdaripour, Closed-form solution of absolute orientation using orthonormal matrices, *Journal of the Optical Society of America A,* vol. 5, pp. 1127-1135, 1988.

[12] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, S. Shenker, GHT: A geographic hash table for data-centric storage, in *ACM international workshop on Wireless sensor networks and applications (WSNA)* Atlanta, Georgia, USA, 2002, pp. 78 - 87.

[13] GreenOrbs http://greenorbs.org/.

[14] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, Canopy closure estimates with GreenOrbs: Long-term large-scale sensing in the forest, in *ACM SenSys* Berkeley, California, USA, 2009, pp. 99-112.

[15] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, S. Teller, Mobile-Assisted Localization in Wireless Sensor Networks, in *IEEE Infocom*, 2005, pp. 172 - 183.

[16] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, P. N. Belhumeur, Rigidity, computation, and randomization in network localization, in *IEEE INFOCOM*, 2004.

[17]D. K. Goldenberg, A. Krishnamurthy, W. C. Maness, Y. R. Yang, A. Young, Network localization in partially localizable networks, in *IEEE INFOCOM* Miami, FL, 2005, pp. 313 - 326

[18] Z. Yang, Y. Liu, X.-Y. Li, Beyond trilateration: On the localizability of wireless ad-hoc networks, in *IEEE INFOCOM* Rio de Janeiro, Brazil, 2009, pp. 2392 - 2400.

[19] J. Aspnes, D. Goldenberg, Y. R. Yang, On the Computational Complexity of Sensor Network Localization, in *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop (ALGOSENSORS)*, 2004, pp. 32–44.

[20] C. Peng, G. Shen, Y. Zhang, Y. Li, K. Tan, BeepBeep: A high accuracy acoustic ranging system using COTS mobile devices, in *ACM SenSys* Sydney, Australia, 2007, pp. 1 - 14.

[21] A. Srinivasan, J. Wu, A Survey on Secure Localization in Wireless Sensor Networks, *Encyclopedia of Wireless and Mobile Communications,* 2008.

[22] H. Song, L. Xie, S. Zhu, G. Cao, Sensor Node Compromise Detection: The Location Perspective, in *IEEE International Wireless Communications and Mobile Computing Conference*, 2007.

[23] F. Wang, L. Qiu, S. Lam, Probablistic Region-based Localization for Wireless Networks, in *ACM Mobile Computing and Communications Review Special Issue on Localization*, 2007.

[24] A. P. Subramanian, P. Deshpande, J. Gao, S. R. Das, Drive-by Localization of Roadside WiFi Networks, in *IEEE INFOCOM*, 2008.

[25] A. Uchiyama, S. Fujii, K. Maeda, T. Umedu, H. Yamaguchi, T. Higashino, Ad-hoc Localization in Urban District, in *IEEE Conf. on Computer Communications*, 2007.

[26] M. Li, Y. Liu, Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes, *IEEE/ACM Transactions on Networking (TON),* vol. 18 (1), pp. 320-332, 2010.

[27] J. Jeong, S. Guo, T. He, D. Du, APL: Autonomous Passive Localization for Wireless Sensors Deployed in Road Networks, in *IEEE INFOCOM*, 2008, pp. 583-591.

[28] Z. Zhong, T. He, Achieving Range-Free Localization Beyond Connectivity, in *ACM SenSys*, 2009, pp. 281-294.

[29] T. He, C. Huang, B. M. Blum, J. A. Stankovic, T. F. Abdelzaher, Range-free localization and its impact on large scale sensor networks, *ACM Transactions on Embedded Computing Systems (TECS),* vol. 4, pp. 877 - 906, 2005.

[30] A. Basu, J. Gao, J. Mitchell, G. Sabhnani, Distributed Localization Using Noisy Distance and Angle Information, in *ACM MobiHoc*, 2006.

[31] D. Niculescu, B. Nath, Error characteristics of ad hoc positioning systems (APS), in *ACM MobiHoc*, 2004.

[32] A. Savvides, W. Garber, S. Adlakha, R. Moses, M. B. Srivastava, On the error characteristics of multihop node localization in ad-hoc sensor networks, in *ACM/IEEE IPSN*, 2003, pp. 317-332.

[33] N. Patwari, J. N. Ash, S. Kyperountas, A. O. H. III, R. L. Moses, N. S. Correal, Locating the Nodes: Cooperative localization in wireless sensor networks, *IEEE Signal Processing Magazine,* vol. 22(4), pp. 54-69, 2005.

[34] A. Savvides, W. Garber, R. Moses, M. B. Srivastava, An analysis of error inducing parameters in multihop sensor node localization, *IEEE Transactions on Mobile Computing,* vol. 4, pp. 567-577, 2005.

Appendix

Proof of the Structural Error Tolerance

We prove the structural error tolerance and analyze the bound of the error tolerance estimation, which is the basis of ETOC design.

Theorem 1. Given a component, it contains an anchor a_1 and two distinct non-anchor nodes n_1 and n_2 sharing two edges with two distinct anchors a_2 and a_3, respectively. Let α, θ, and φ denote the angle separations of $\angle a_2a_1a_3$, $\angle n_1a_1a_2$, and $\angle n_2a_1a_3$, respectively. Define set S as follow, $S=\{\alpha+\theta+\varphi,\ \alpha+\theta-\varphi,\ \alpha-\theta+\varphi,\ \alpha-\theta-\varphi\}$. Let l_1 and l_2 denote the in-component distances of node pair (a_1, n_1) and (a_1, n_2), respectively, and δ denote $\min\{|\cos\beta_1 - \cos\beta_2|\}$ for all $\beta_1, \beta_2 \in S$. Then, the lower bound of the structural error tolerance for node pair (n_1,n_2) is:

$$T_C = \frac{1}{2}\frac{l_1 l_2}{l_1 + l_2}\delta \, .$$

Proof: As shown in Figure 13, by the measurements to anchors a_1 and a_2, node n_1 has two candidate positions, n_1 and n_1'. Similarly, node n_2 has two candidate positions, n_2 and n_2'. Then, there are four possible distances of node pair (n_1,n_2), uniformly denoted by a set $D=\{\sqrt{l_1^2 + l_1^2 - 2l_1l_2\cos\beta} \mid \beta \in S\}$. If we determine the positions of node n_1 and n_2 by selecting the distance in D that has the lest absolute difference with the distance of node pair (n_1,n_2), then the robustness condition must guarantee that other candidate distances will not corrupt this selection procedure. That is, the distances in D must be differentiable to each other even when error occurs. Consequently, the structural error tolerance is denoted by $\min\{|d_1-d_2|/2\}$ for all $d_1, d_2 \in D$.

For any $\beta_1, \beta_2 \in S$, they will select two distances in D, denoted by d_1 and d_2. Then, the error tolerance of this two distances is $t_{1,2}=|d_1-d_2|/2$. Hence, we obtain:

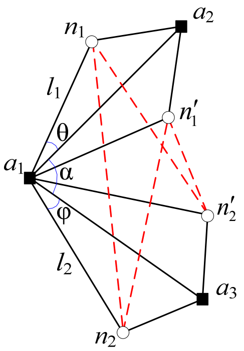


Fig. 13. Proof of robustness

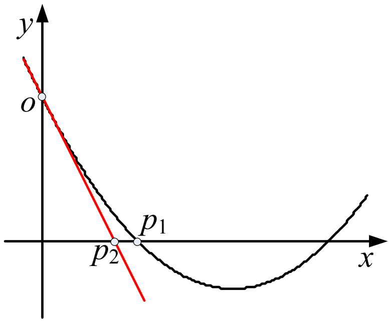


Fig. 14. The lower bound of $T_{1,2}$

$$t_{1,2}^2 = b[a-(\cos\beta_1+\cos\beta_2)-\sqrt{a^2-2a(\cos\beta_1+\cos\beta_2)+4\cos\beta_1\cos\beta_2}\,],$$

where $a=(l_1{}^2+l_2{}^2)/l_1l_2$, $b=l_1l_2/2$.

We construct the following conic section:

$$f(x) = x^2 - 2[a - (\cos\beta_1 + \cos\beta_2)]x + (\cos\beta_1 - \cos\beta_2)^2.$$

Define $T_{1,2} = t_{1,2}^2 / b$. Then, $T_{1,2}$ is the smaller root of the equation $f(x) = 0$. Clearly, $T_{1,2}$ lies in the interval $[0, a - (\cos\beta_1 + \cos\beta_2)]$. We illustrate the curve of $f(x)$ in Figure 14, where p_1 denotes the value of $T_{1,2}$. Considering the line op_2, which is the tangent of $f(x)$ at the position $(0, (\cos\beta_1 - \cos\beta_2)^2)$, it intersects the x-coordinate at the point p_2.

Hence, the value of p_2 is a lower bound of $T_{1,2}$. That is,

$$T_{1,2} \ge \frac{1}{2}\frac{(\cos\beta_1 - \cos\beta_2)^2}{a-(\cos\beta_1+\cos\beta_2)},$$

and we use lb_1 to denote this lower bound. Considering $\cos\beta_1 + \cos\beta_2 \ge -2$, we obtain another lower bound of $T_{1,2}$:

$$T_{1,2} \ge \frac{1}{2}\frac{(\cos\beta_1 - \cos\beta_2)^2}{a+2},$$

denoted by lb_2. Substituting a and b, we obtain a lower bound of $t_{1,2}$, that is,

$$t_{1,2} \ge \frac{1}{2}\frac{l_1l_2}{l_1+l_2}|\cos\beta_1 - \cos\beta_2| \, .$$

By this conclusion, for arbitrarily selected $\beta_1, \beta_2 \in S$, we obtain the lower bound of the structural error tolerance for node pair (n_1,n_2):

$$T_C = \frac{1}{2}\frac{l_1l_2}{l_1+l_2}\delta \, . \blacksquare$$