Target Localization using Proximity Binary Sensors

Qiang Le
Department of Electrical Engineering
Hampton University
Hampton, VA 23668
qiang.le@hamptonu.edu

Lance M. Kaplan
US Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197
lkaplan@ieee.org

Abstract—This work presents the maximum likelihood localization (ML) algorithm for multi-target localization using proximity-based sensor networks. Proximity sensors simply report a single binary value indicating whether or not a target is near. The ML approach requires a hill climbing algorithm to find the peak, and its ability to find the global peak is determined by the initial estimates for the target locations. This paper investigates three methods to initialize the ML algorithm: 1) centroid of \( k \)-means clustering, 2) centroid of clique clustering, and 3) peak in the 1-target likelihood surface. To provide a performance bound for the initialization methods, the paper also considers the ground truth target positions as initial estimates. Simulations compare the ability of these methods to resolve and localize two targets. The simulations demonstrate that the clique clustering technique outperforms \( k \)-means clustering and is nearly as effective as the 1-target likelihood peak methods at a fraction of the computational cost.

TABLE OF CONTENTS

1 INTRODUCTION ...................................... 1
2 SENSOR MODEL ..................................... 2
3 MAXIMUM LIKELIHOOD POSITION ESTIMATOR .... 2
4 ML INITIALIZATION ............................... 3
5 EXPERIMENTS ...................................... 5
6 CONCLUSIONS ..................................... 7
REFERENCES ......................................... 8

1. INTRODUCTION

Wireless sensor networks represent an inexpensive pervasive technology for situational awareness. The simplest sensor that one can conceive is a proximity sensor that acts as a tripwire, i.e., it reports a detection when a target close by triggers it. Examples of sensor modalities that are suitable for low cost proximity nodes include seismic, acoustic, passive infrared, etc. For many types of sensors, it is possible to implement simple energy detection algorithms where the signal, i.e., voltage squared, is integrated over a processing interval and compared to a threshold. The sensor node reports a "1" for detection (exceeds the threshold) or "0" for no detection. Despite the minimal information provided by a binary sensor, a network of such sensors can potentially be used to localize and track targets that trip the sensors. Previous target location algorithms apply to the single target case [1], [2], [3], [4], [5], [6]. Our work here focuses on multiple target localization. The ability to detect and track targets is governed by operating parameters such as sensor density, the threshold settings, and the spacing between the targets. The ultimate goal of this research is to explore fundamental performance bounds that determine how well a network of proximity sensors can resolve and localize multiple targets as a function of the operating parameters.

This paper extends the single target maximum likelihood (ML) proximity sensor localization method proposed in [2] for multiple targets when number of targets is known. The more realistic scenario where the number of targets is unknown \textit{a priori} will be investigated in future work incorporating well known and obvious techniques such as Akaike’s information criteria (AIC) [7] and minimum description length (MDL) [8]. The ML approach is implemented using a hill climbing algorithm to find the peak. The ability to find the global peak depends on how the ML approach is initialized. The paper considers three practical initialization approaches: 1) centroid of \( k \)-means, 2) centroid of largest two maximal cliques, and 3) peak in 1-target likelihood surface. To provide a performance bound, the paper also considers the ground truth target positions to initialize the hill climbing ML approach.

The first two initialization methods employ clustering approaches; namely, \( k \)-means clustering and clique clustering. In general, both methods cluster the sensor detections in \( K \) classes associated to the known \( K \) number of targets. The first approach employs the \( k \)-means algorithm by clustering based on the geographic positions of the nodes reporting detections and returns the \( K \) centroids as the initial target position estimates. On the other hand, the maximal clique approach first creates a graph of the nodes returning detections where edges exist between nodes when they are both able to detect a common target at sufficient \( P_d \). Then, it finds \( K \) maximal cliques under the assumption that each clique is generated from a different target. Finally, it computes the centroid
associated to the node positions from the $K$ cliques as the initial estimates of the target position.

Unlike the first two methods, the 1-target likelihood surface does not assume any association between the sensor detections and the targets. However, it is computationally complex. It first assumes one target and calculates the likelihood of the target position over a uniform grid of points. The top $K$ peaks of the likelihood map are the initial estimates of the target position.

The paper investigates the ability of the different initialization approaches to resolve and localize targets as function of sensor density and target separation. Namely, the root mean squared (RMS) position errors and the ability to discern two targets are reported as a function of the operating parameters. The paper focus on the two target case. Finally, the paper also considers the unbiased Cramer-Rao lower bound (CRLB) as a means to understand the performance limitations of networks of proximity sensors.

The paper is as organized as follows. Section 2 provides details about the sensor model, and Section 3 presents the ML estimator. Section 4 introduces the various initialization methods for the ML estimator. The initialization approaches are compared via simulations in Section 5. Finally, Section 6 provides concluding remarks.

2. SENSOR MODEL

This section provides the sensor model for proximity sensors that has been incorporated in earlier work [6], [9]. The received power $p^t_i$, that is, the power measurement of the $i$-th sensor at time $t$, is given by

$$ p^t_i = \sum_{k=1}^{K} p^t_{0,k} \left( \frac{r_{0,k}}{r_{i,k}} \right)^{\alpha} + v^t_i, \quad (1) $$

where $p^t_{0,k}$ is the power measured at a reference distance $r_{0,k}$ for the $k$-th target, $K$ is the total number of targets, $r_{i,k}$ is the relative distance between the $i$-th sensor and the $k$-th target, $\alpha$ is the attenuation parameter that depends on the transmission medium, and $v^t_i \sim N(\mu_v, \sigma_v^2)$.

The mean and variance of the error $v^t_i$ is derived from the zero mean measurement noise of variance $\sigma^2$ for the case that the measured power is the result of integrating the square of the measurements over $L$ samples. As shown in [6], [9], $\mu_v = \sigma^2$ and $\sigma_v^2 = 2\sigma^4/L$. This paper assumes the references $r_{0,k} = r_0$ and $p_{0,k} = p_0$ for all $K$ targets.

The $i$-th sensor measures the received power $p^t_i$, processes it locally and reports a single binary digit: ‘1’ for the presence of one or more targets or ‘0’ for the absence of any target. The decision follows the rule:

$$ d^t_i = \begin{cases} 1 & p^t_i > \lambda \\ 0 & p^t_i \leq \lambda \end{cases} \quad (2) $$

Given a probability of false alarm $P_{fa}$, the threshold $\lambda$ is computed as:

$$ \lambda = \sigma_v \sqrt{2} \text{erf}^{-1} (1 - 2P_{fa}) + \mu_v, \quad (3) $$

where erf is the standard error function

$$ \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. $$

Furthermore, the probability of detection $P_d$ at the $i$-th sensor, for a given $P_{fa}$, as a function of distance to the $k$ targets, is

$$ P_d(r^t_i) = \text{Prob}(d^t_i = 1|\text{targets}), \quad (4) $$

$$ = \text{Prob}(p^t_i > \lambda|\text{targets}), $$

$$ = \frac{1}{2} - \frac{1}{2} \text{erf} \left( \frac{\lambda - \sum_{k=1}^{K} p^t_{0,k} \left( \frac{r^t_{0,k}}{r^t_{i,k}} \right)^{\alpha} - \mu_v}{\sqrt{2}\sigma_v} \right), $$

where $r^t_{i,k} = \left[ r^t_{i,0}, \ldots, r^t_{i,K} \right]$. Fig. 1 illustrates $P_d$ as a function of the distance of a single target to a sensor. The figure demonstrates that $P_d$ as given by (4) is a sigmoid function that steps down from a value near one to a value near $P_{fa}$ over a transition region of about 200 meters.

3. MAXIMUM LIKELIHOOD POSITION ESTIMATOR

The single target ML localization algorithm for binary sensors was devised in [2] and further explored in [4]. The likelihood is the probability of the observed measurements for the $N$ sensors, i.e. $d^t_i$ for $i = 1, \ldots, N$, conditioned on the location of the targets $x_k$ for $k = 1, \ldots, K$ and the location of the sensor nodes $S_i$ for $i = 1, \ldots, N$. Thus, the likelihood function is given by

$$ \prod_{i=1}^{N} P_d(r^t_i)^{d^t_i} \left( 1 - P_d(r^t_i) \right)^{(1-d^t_i)}. \quad (5) $$

In general, the $K$ target localization algorithm is

$$ \{ \hat{x}_1, \ldots, \hat{x}_K \} = \quad (6) $$
The unbiased CRLB is the inverse of the Fisher information matrix (FIM), i.e.,

\[ C = J^{-1}, \]

and the FIM \( J \) is the expected covariance of the gradient of the log-likelihood evaluated at the true target state parameter values. Given the likelihood \( (5) \) for the case that \( K = 1 \), the FIM can be expressed as

\[
J = \sum_{i=1}^{N_x} J_i,
\]

where

\[
J_i = \frac{1}{P_{d_i}(1-P_{d_i})} \left[ \left( \frac{dP_{d_i}}{dx} \right)^2 \left( \frac{dP_{d_i}}{dy} \right)^2 \right],
\]

\[
dP_{d_i} = \frac{-\alpha \rho \sigma^* \cos(\theta_i)}{\sigma^* \alpha^+} \frac{1}{\sqrt{2\pi}} e^{-\frac{r^2}{2}},
\]

\[
\frac{dP_{d_i}}{dy} = \frac{-\alpha \rho \sigma^* \sin(\theta_i)}{\sigma^* \alpha^+} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}},
\]

and by \( (4) \) for \( K = 1 \),

\[
P_{d_i} = P_{d}(r_{i,1}) = \frac{1}{2} \left[ 1 - \frac{1}{2} \text{erf} \left( \frac{\lambda - p_{0,1} \left( \frac{r_{i,1}}{\sigma_v} \right)^\alpha - \mu_v}{\sqrt{2}\sigma_v} \right) \right].
\]

**4. ML INITIALIZATION**

The ML solution is estimated via an optimization search algorithm that relies on initial estimates of the target positions. Noting that the ML search could be easily trapped in the local optimum due to a “bad” initial guess, performance of the ML depends on the algorithm that determines the initial estimates of the target locations. This paper considers three practical methods: 1) centroid of \( k \)-means, 2) centroid of largest two maximal cliques, and 3) peak in 1-target likelihood surface. The two clustering methods first associate nodes to the various \( K \) targets. Then, the centroid of the clusters are used as the initial estimates for the target locations. The peak in 1-target likelihood surface looks for peaks by gridding the likelihood surface under the assumption that \( K = 1 \). The clustering methods, centroid estimation, and 1-target likelihood methods are described below in further detail.

**Figure 2.** An undirected graph with 6 nodes and 10 edges.

**K-means Clustering**

The \( k \)-means clustering is a method of cluster analysis which aims to partition \( n \) observations into \( K \) clusters in which each observation belongs to the cluster with the nearest mean. Specifically, given a set of observations \( \{y_1, y_2, \ldots, y_n\} \), where each observation \( y_i \) is a \( d \)-dimensional real vector, then \( k \)-means clustering aims to partition the \( n \) observations into \( K \) sets \( C = (C_1, C_2, \ldots, C_k) \) so as to minimize the within-cluster sum of squares (WCSS):

\[
\arg \min_{C} \sum_{i=1}^{k} \sum_{y_j \in C_i} ||y_j - u_k||^2,
\]

where \( u_k \) is the mean of \( C_k \). Here we use the \( k \)-means function in Matlab where the number of clusters is equal to the number of targets \( K \), and the set of observations consist of the 2-D locations of the sensor nodes that detected the target.

**Clique Clustering**

**Graph Theory Preliminary**—In an undirected graph \( G \), two vertices \( u \) and \( v \) are called connected if \( G \) contains a path from \( u \) to \( v \). Otherwise, they are called disconnected. Vertices connected by an edge, i.e., by path of length 1, are called adjacent. A graph is considered connected if every pair of distinct vertices in the graph can be connected through some path. A complete graph with \( n \) vertices is a connected graph where each vertex has \( n-1 \) adjacent neighbors. A maximal complete subgraph, also called a clique, is a complete subgraph that is not contained in any other complete subgraph. For example, in an undirected graph with 6 nodes and 10 edges shown in Fig. 2, there are 4 cliques: 1) nodes \((1,2,4)\), 2) nodes \((2,4,5)\), 3) node \((1,3,6)\), and 4) nodes \((1,4,6)\).

**Clique Clustering**—To use the clique clustering for the proximity-based localization, one needs to define an undirected graph. The vertices in the graph are the proximity nodes that report detections, and the edge between the nodes should address the chance of any two proximity nodes reporting the presence of a common target. To this end, the clustering approach builds an adjacency matrix \( A \) where the element at the \( i \)-th row and \( j \)-th column is one, that is, \( A_{i,j} = 1 \) when
with high probability the $i$-th and $j$-th nodes can respond to a common target. In other words,
\[
A_{i,j} = \begin{cases} 
1 & P_d(\frac{r_{i,j}}{\tau_d}) \geq \tau \\
0 & P_d(\frac{r_{i,j}}{\tau_d}) < \tau,
\end{cases}
\]
where $\tau$ is a user defined threshold and $r_{i,j}$ is the distance between the $i$-th and $j$-th nodes. In our work, the threshold is set to $\tau = 0.5$.

After the adjacency matrix is computed, clique clustering computes the set of maximal cliques in the graph of nodes reporting detections. To this end, we consider two algorithms to compute the set of possible maximal cliques: the Bron-Kerbrosch (BK) and the Greedy algorithms. The BK algorithm uses recursion and pruning to find all maximal cliques [10], [11]. As described in Fig. 3, it maintains three sets: $C$, $P$, and $X$. The set $C$ represents the currently growing clique; the set $P$ represents the prospective nodes which are candidates for the extension of the set $C$; and the set $X$ contains the nodes that have already been processed and should not be considered as the extension for the current clique $C$. Naturally, since the BK works recursively, it still exhibits a NP computation complexity.

```plaintext
BK initialization: BK(∅, V, ∅) where V is the list of all nodes in the graph. 
BK(C, P, X) subroutine
if P = ∅ and X = ∅,
   Report C as a maximal clique
else
   for n ∈ P
      P = P − n
      N(n) = the neighbors of the node n
      P_{new} = P ∩ N(n)
      X_{new} = X ∩ N(n)
      C_{new} = C ∪ n
      BK(C_{new}, P_{new}, X_{new})
      X = X ∪ n
   endfor
endif
```

Figure 3. Bron-Kerbrosch search algorithm.

The Greedy search avoids the recursion in the BK algorithm in order to find the maximal cliques. However, it does not exhaust all possible maximal cliques. Fig. 4 provides the Matlab pseudo codes for the Greedy algorithm. The following ideas motivate the use of the Greedy search:

1. The indices of nonzero elements in each row of the adjacent matrix $A$ construct a candidate for the clique, which means one can stop the search when all the rows are visited.

2. To find the maximal clique that includes the $i$-th node, one can use the Greedy idea by adding one node at a time. In other words, the algorithm adds its unexplored adjacent node one by one to the currently growing clique until all the adjacent nodes are visited\(^3\). If the proposed new node is connected to all nodes in the current clique, then the clique is augmented by the proposed node.

The BK search provides all maximal cliques of a graph while the Greedy search gives a subset of maximal cliques. This greedy method could be justified by the fact that not all of the maximal cliques associate to actual targets. For the case that the number of targets $K$ is known, we need to choose $K$ cliques that are associated with the targets. To this end, the clique selection method in Fig. 5 chooses the $K$ farthest maximal cliques according to their Euclidean distance to represent the $K$ clusters that are responding to each of the $K$ targets. For example given the four cliques shown in Fig.2 and $K = 2$, two cliques represented by nodes (2,4,5) and (1,3,6) are the farthest ones according to the Euclidean distance. Fig. 5 provides the Matlab codes to find $K = 2$ cliques that the algorithm assumes should be associated with $K = 2$ targets. Different clique clustering methods (BK and the Greedy) must be combined with the clique selection methods to provide localization performances. In this sense, the Greedy clustering method may provide comparable localization accuracy as the BK clustering method when combined with clique selection methods.

```plaintext
// Input:
// A: the N × N adjacency matrix where N is the number of nodes in the graph and the entries of A are either 0 or 1.
// Output Variables:
// C: the N × N matrix where the indices of the nonzero elements in the row i denote a clique including this node;
// C = zeros(N, N);
// for i ∈ {1, 2, …, N}
//   r_i = A(i,:);
//   discovery = [i : N, 1 : i − 1];
//   r_i = r_i(discovery);
//   //arrange the neighbors of the node i in the discovery order
//   neighbor = discovery(find(r_i == 1));
//   //add one node at a time to find the clique for the node i
//   for j ∈ neighbor
//      adj = find(C(i,:) == 1);
//      new = [adj, j];
//      if length(find(A(new, new) == 0)) == 0
//         C(i, j) = 1;
//      end
//   end
// C = unique(C, 'rows');
```

Figure 4. Greedy search algorithm.

Centroid Position Estimator—After the cliques or clusters are determined from $k$-means clustering or clique clustering, the

\(^3\)The order that is chosen to add nodes one by one can give different suboptimal answers.
**dist**: the $N_c \times N_c$ matrix that records the pairwise Euclidean distance of $C$, and $N_c$ is the total number of distinct cliques;

**$F$**: the $2 \times N_c$ matrix where the indices of the nonzero elements in each row denote a clique.

// find the two farthest cliques

\[
\text{dist} = \text{triu}(\text{squareform}(	ext{pdist}(C, 'euclidean')));
\]

\[
[v, v_1] = \max(\max(\text{dist}, [], 1));
\]

\[
[v, v_2] = \max(\text{dist}(v_1, [], 2));
\]

\[
F = C([v_1, v_2], :)
\]

**Figure 5.** Clique selection method.

The centroids of the clusters are used as the initial estimates for the target locations. Namely,

\[
x_k^* = \frac{\sum_{i \in C_k} S_i}{|C_k|},
\]

for $k = 1, \ldots, K$.

**Peak 1-Target Likelihood Surface**

Unlike the clustering methods, the peak 1-target likelihood surface method does not associate measurements to targets, but it is computationally complex. It first assumes one target and calculates the likelihood of the target position over a uniform grid of points by (5). Figs. 6(b) and (d) show the likelihood maps for two targets separated by 600m and 100m, respectively. The shaded areas are either due to the possible target locations or no sensor coverage, and the dark blue disk regions are due to sensors reporting 'no targets'. Note that there exist multiple peaks in the map. For the case that the number of targets $K$ is known, the method picks the $K$ top peaks of the likelihood map as the initial estimates of the target positions. For the case that $K = 1$, this method is essentially calculating ML localization given that the sampling on the grid is fine enough.

**5. EXPERIMENTS**

The first set of simulations considers a single target traveling along a straight line at a speed of 10m/s in a proximity sensor network field of size $2\text{km} \times 2\text{km}$ where the number of nodes varies from 100 to 400 in an increment of 50. Fig. 7 illustrates the sensor/target geometry for $N = 200$ sensors. The time interval for the target estimation is 1 second. We observe the localization for 100 seconds. For each of $N$ number of sensors, we ran 100 simulations (10 random network configurations $\times$ 10 Monte Carlo runs per configuration). Fig. 8 shows the localization performances for different ML methods as a function of the number of nodes. Note that localization performance refers to the RMS position error of the $K$-target ML when initialized by one of the four localization techniques. Also, the clique clustering is implemented via the Greedy search as motivated by the final set of simulations described in this section (see Table 1). The grid-based ML method using the largest peak in the 2D ML surface mostly approaches the ML performance bound since it locates the
largest peak of the likelihood solution over the support of the grid, and this peak should be near the global maxima. The figure also provides the CRLB as computed by (7). The CRLB curve demonstrates that the performance of the localization methods can outperform the unbiased CRLB because the localization methods are biased. Fig. 9 shows the bias for each ML method

$$b = \|\mathbf{E}(\hat{x}) - x\| = \left\| \frac{1}{M} \sum_{i=1}^{M} (\hat{x}_i - x) \right\|,$$

where $x$ is the true 2D target location, and $M$ is the number of simulations. Clearly, the bias of the ML method using the ground truth is smallest but still is non zero when $N$ is as large as 400. The ML methods provides a mapping from the different $2^N$ possible sensor outputs to estimated target locations, akin to a Voronoi diagram. The bias is due the finiteness of the possible estimated target locations. The bias decreases as the size of the Voronoi-like cells decreases. The size of these cells is dictated by operational parameters such as the sensor density (or $N$), $P_{fa}$, etc. When the $P_{fa}$ is set up appropriately, it is expected that the bias decrease to zero as the number of the proximity sensors approaches infinity.

The next set of simulations considers the case of two targets separated by 600m traveling through a sensor network field of size 2km×2km. In this case, the number of proximity sensors varies from 100 to 400 in increments of 50. For each case of $N$ sensors, we ran 5000 simulations (100 random network configurations × 50 Monte Carlo runs per configuration). Fig. 10 shows the localization performance versus $N$ curves for the four ML initialization techniques over this two-target scenario. The figure demonstrates that the ground truth initialization does serve as a performance bound. When the density of the sensors is not too small (greater than 200 nodes per 4km²), the clique and 1-target peak methods clearly outperform the $k$-means method. Furthermore, as the $N$ gets large, the localization performance of the clique and 1-target peak methods is approaching the performance bound using ground truth initialization. For small density values, the three practical initialization methods demonstrate similar localization performance. This may be due to the fact that none of the methods can distinguish, i.e., resolve, two targets from the set of nodes detecting the targets.

Clearly, the ability to resolve two targets is dictated by the sensor density and target separation. Figure 6 demonstrates that for the given sensor density at a target separation of 600m, the 1-target likelihood map can resolve the two targets, but for a target separation of 100m, the targets are not resolved. To better understand the role of sensor density and target separation for target resolution, we ran a third set of simulations over a sensor network field of size 2km×2km. In these simulations, the distance between the two targets varied from 500m to 1000m and the number of sensors varied from 100 to 600. For each parameter value of target separation and number of sensor nodes, we computed the 1-target likelihood map over 5000 simulations (100 random network configurations × 50 Monte Carlo runs per configuration). For each likelihood map, the two targets are considered resolved when the two peaks closest to the ground truth target positions are
the two largest peaks in the likelihood map. Otherwise, the second peak likely originated from noise rather than from an actual target. Fig. 11 plots the curves for the necessary number of nodes for a given target separation so that $x\%$ of the simulations are able to resolve the two targets. The figure demonstrates the critical density, i.e., the number of nodes over the 4km$^2$ sensor field, to achieve two target resolution as a functions of target separation. For instance, it is shown that when targets are separated by 500m, one should expect to resolve 80% of the targets when $N = 260$ and 90% of the targets if $N = 400$. For a target separation of 600m, there is hardly any target resolution when $N = 100$, which is the point where all practical initialization techniques demonstrate similar performance in Fig. 10. Once the probability of resolving the two targets is high, the clique and 1-target likelihood peak methods demonstrate similar localization performance as ground truth initialization.

Table 1 compares the average processing time and RMS position errors for the ML initialization methods when the two targets are separated by 600m over 5000 simulations (100 random network configurations $\times$ 50 Monte Carlo runs per configuration). The grid spacing for the 1-target peak likelihood method is 50m in a 2km$^2$ region. The localization performance of the clique initialization via Greedy or BK search are virtually identical. On the other hand, the Greedy search is significantly faster than the BK search for identifying the two maximal cliques. The computationally efficiency of the Greedy search over the BK search improves as the number of nodes increases. Therefore, the Greedy search is used to implement the clique initialization in practice. The Greedy implementation of clique clustering is comparable to $k$-means clustering in implementation time, but clique clustering provides significantly better localization performance. Furthermore, the clique clustering provides comparable localization performance to 1-target peak likelihood at about four times the speed.

6. CONCLUSIONS

This work presents the ML localization method for the proximity-based sensor networks. The likelihood is based upon the probability of detection of each sensor node. Furthermore, the probability of detection is a function of the location of the targets relative to the sensor node. This paper investigated the performance of the ML localization method when initialized by various techniques: 1) centroid of $k$-means clustering, 2) centroid of clique clustering, 3) peak in 1-target likelihood surface, and 4) ground truth target positions. The ground truth initialization technique serves as a performance bound for the other three more practical techniques. For the single target case, the 1-target likelihood methods outperforms the other two practical techniques as expected. For the two-target case, the the 1-target likelihood method demonstrates similar localization performance as clique clustering. Both of these techniques approaches the performance bound when then the number of proximity sensors is large and it is easy to resolve the two targets. In addition, the implementation of the clique clustering can be accomplished much faster via a suboptimal Greedy search at similar localization performance as the optimal Bron-Kerbrosch search. Furthermore, the Greedy implementation of the clique clustering is about four times faster than the 1-target localization method. As a result, the clique clustering appears be a viable initialization technique for 2-
future localization methods for $K > 2$ and the effects of threshold settings on the proximity sensors to localization performance. Furthermore, when the the practical initialization techniques are unable to resolve the two targets, there is a gap between the realized localization performance and the performance bound. One must study whether or not it is possible for a practical initialization technique to achieve better performance than the ones investigated in this paper. Finally, we will investigate the performance of $K$-target ML localization when $K$ is not known a priori.

**References**


