

Non-Interactive Location Surveying for Sensor Networks with Mobility-Differentiated ToA

Jun Luo Hersh V. Shukla Jean-Pierre Hubaux

School of Computer and Communication Sciences

EPFL (Swiss Federal Institute of Technology at Lausanne), CH-1015 Lausanne, Switzerland

Email: {jun.luo, hershvardhan.shukla, jean-pierre.hubaux}@epfl.ch

Abstract—Location-awareness is crucial to many applications of sensor networks. Existing location surveying approaches either rely on an inflexible infrastructure or suffer from high computation and communication load. In this paper, we present Non-intERactive lOcation Surveying (NEOS) to address certain deficiencies in the existing approaches. The key contribution of NEOS is twofold: (i) it employs a mobile beacon to introduce *mobility-differentiated time-of-arrival* (MDToA) observations, a special form of *time difference of arrival* (TDoA), at the node side and (ii) it involves simple computations and entails no node-to-node communication. MDToA enables us to devise flexible and robust positioning algorithms; the resulting computational load fully obeys the processing constraints of sensor nodes. Furthermore, the non-interactive feature of NEOS allows of a substantial reduction on the nodes' energy consumption. We have implemented a preliminary prototype of NEOS using CricketMotes. Our experiments with this prototype demonstrate a location accuracy within 2cm in a 16m² area.

Index Terms—Location surveying, Sensor networks, Mobile beacons, MDToA, TDoA.

I. INTRODUCTION

Location-awareness is becoming increasingly important in sensor networks. Its necessity stems from applications such as guiding navigation (e.g., [1]), source localization (e.g., [2]), efficient routing with geographical information (e.g., [3]), and coverage awareness (e.g., [4]). In response to this necessity, a large number of *location surveying*¹ techniques have been proposed (e.g., [6]–[23]). These techniques can be roughly categorized into two classes, namely *non-interactive* (NI) and *interactive* (IT), based on whether the communications involved are unidirectional or bidirectional. A NI approach

usually relies on external infrastructures, such as GPS and beacon/sensor arrays [6], [24], [12]. An IT approach allows nodes in a network to build location information autonomously. It may either require mere assistance from a few *anchors* whose locations are known (e.g. [7], [8], [10], [13], [16], [21]), or be anchor-free (e.g., [9], [11], [15], [19]).

In this paper, we are interested in large scale wireless sensor networks that perform monitoring of, for example, wildlife habitat, avalanche/landslide-prone areas, forests (e.g., fire detection), and underwater zones (e.g., tsunami alert). Nodes involved in such networks are resource-constrained (in terms of both computation and communication), whereas the applications (e.g., event localization) require **fine-grained** location information for each node. In this context, a NI approach that relies on either GPS-equipped nodes or beacon/sensor arrays (e.g., [25], [6], [26], [12]) might not scale to large geographical areas, due to the additional costs that can become huge for a large network size. The IT approach, though designed to improve scalability, has its own disadvantages. First, the ranging errors inherent in the applied ranging technology² can propagate and thus lead to even larger errors in coordinate estimations. Secondly, the energy consumed for ranging and (location and distance) information exchanging can be significant ($\mathcal{O}(m^2)$ with m being the maximum node degree [19]). Thirdly, translating mutual distances into coordinate assignments may involve intensive computations (e.g, [8], [7]). Finally, networks can only be partially localizable even with relatively high node degrees [23].

The situation can be improved if a small set of **mobile** nodes [27] are applicable to a sensor deployment area. On one hand, a NI system benefits from the multiple measurements obtained at distinct positions, which are made possible by the mobile nodes, to improve its location accuracy [17], [18], [20]. On the other hand,

¹The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322. (<http://www.terminodes.org>)

¹The term, borrowed from [5], refers to the process of specifying the locations of the network components.

²In fact, it is already questionable whether we can afford to equip every node with the ranging ability.

mobile nodes can also assist an IT system to build a globally rigid structure by adding distance constraints [22]. In general, node mobility brings the same benefit, i.e., refining distance measurements, to both NI and IT approaches; this seems to suggest that, aided by mobile nodes with location-awareness, a NI approach could be more efficient than an IT one. The reason is simple: the former approach needs only one phase (i.e., ranging) whereas the latter approach requires at least two phases (e.g., ranging+multilateration [22]). Unfortunately, the existing mobile-aided NI location systems, which rely on either RSSI-based ranging mechanisms [17], [20] or proximity information [18], do not scale well with increasing network sizes. For example, intensive computation must be performed in a centralized way if sophisticated estimation algorithms are used [20], and a substantial number of distance measurements have to be obtained for each node in order to compensate the inaccuracy of location estimations [17], [18].

The aforementioned observations motivate us to revisit the mobile-aided NI approach. Our Non-intERactive lOcation Surveying (NEOS) makes use of *mobility-differentiated time of arrival* (MDToA) to allow nodes to localize themselves by passively listening to a mobile beacon. MDToA, as a special form of *time difference of arrival* (TDoA), introduces a difference between two arrival times through the movements of a mobile beacon. The main contributions of our work are as follows:

- We exploit mobility to differentiate the *time of arrival* (ToA) by which location information can be inferred. This technique makes use of signal propagation time rather than attenuation as an indicator of distances, which is much less error-prone. In addition, it avoids the need for synchronization (which is a chicken-and-egg dilemma with localization).
- We require only the mobile node to send beacon signals; other nodes simply listen to the signals and make observations about the differentiated ToAs in order to find their own locations. This allows all (static) nodes to compute their coordinate estimations without interfering with each other.
- We devise algorithms with low computational load for coordinate estimations. We also propose simple traces for the mobile beacon, which provide adequate observations for all nodes at the same time.

In addition, NEOS guarantees location privacy [24] thanks to the localized computation. Finally, NEOS is not meant to replace any IT system. In fact, NEOS can be **integrated** with existing IT systems; it provides fine-grained localization for nodes that either serve as anchors or cannot be localized due to scarce connectivity.

This paper is organized as follows: In the next section, we provide some background on ranging techniques and the principles of MDToA. Section III describes the algorithms based on MDToA for localization. Section IV proposes proper mobility strategies, taken by a mobile beacon, that enable the MDToA-based localization mechanism. The experiment results are reported in Section V. Section VI surveys related work. Finally, Section VII concludes the paper.

II. BACKGROUND ON RANGING TECHNIQUES AND PRINCIPLES OF MDToA

In this section, we briefly examine the essence of different ranging techniques, including RSSI, TDoA, AoA and our MDToA. By comparing these techniques, we highlight the benefit of MDToA.

A. Receiver Signal Strength Indicator (RSSI)

RSSI, initially used for power control in wireless networks, can also serve as a tool for distance estimation. The basic idea is that, given a predefined transmission power as well as a signal propagation model that maps a transmission power and a distance to a received power, one can estimate the distance from a receiver to a sender by checking the strength of the received power. The benefit of using RSSI for localization in large systems such as sensor networks is obvious: trilateration can be achieved for all nodes with only three beacons, and nodes only perform passive listening³. Unfortunately, the existing signal propagation models are far from perfect (as a result of, for instance, obstructions, reflections, and non-uniform spreading.) [29]. This imperfection significantly limits the accuracy of localization; a receiver usually needs to perform sophisticated algorithms to synthesize the RSSIs from multiple ($\gg 1$) senders to achieve adequate accuracy [17], [20].

B. Time Difference of Arrival (TDoA)

TDoA makes use of signal propagation speed, which is more robust than the signal attenuation feature used by RSSI. Ideally, if a sender and a receiver are precisely synchronized, the measurement of *time-of-flight* (ToF) is already enough to indicate the distance between them. However, synchronization whose precision can match the signal (in particular RF) speed is hard to achieve. The TDoA mechanism is commonly used in cellular

³Although many proposed localization systems (e.g., RADAR [28]) require nodes to transmit signals and make localization computation at the fixed receiver side, we consider the reverse (e.g., [17]) to be more scalable in large systems, in terms of both energy efficiency and interference avoidance.

networks for localizing a handset. Since it requires only the difference between arrival times observed at several receivers (base stations), TDoA removes the need for synchronization between the handset and receivers, but the receivers must still be synchronized.

TDoA has an alternative definition in recent literature [26], [8]: the difference between arrival times of two signals (in particular, RF and ultrasound). This mechanism, although termed TDoA, actually applies ToF; it measures the propagation time of one signal (ultrasound in the case of Cricket) and makes use of another signal (whose propagation speed is much faster and thus the propagation time can be neglected) to perform time synchronization. Using different types of signal for ranging has a limitation: one of them might not work properly in an environment that favors another (e.g., the transmission range of acoustic/ultrasound is short in the air, while RF propagates poorly underwater). Therefore, ranging mechanisms relying on only one signal could be desirable for sensor location surveying.

C. Angle of Arrival (AoA)

AoA, again a localization mechanism used initially in cellular networks, requires each receiver to be equipped with additional gear (e.g., an antenna array) to detect the bearing from which the sender's signal is arriving. Sensor nodes to be localized are usually very small; it is not realistic to apply this mechanism because of the limitation on size and power consumption of a node. Although devices such as Cricket Compass [30] may be used to provide AoA [10] and thus perform localization, a node equipped with 4 to 5 ultrasound receivers [30] might still break the constraints on size and energy consumption in many scenarios.⁴

D. Mobility-Differentiated Time-of-Arrival (MDToA)

Let us first look at a “reversed” TDoA system, illustrated in Fig. 1, before elaborating the principle of MDToA. Assume that (i) the three beacons are synchronized, (ii) they are aware of their locations, and (iii) they transmit signals that convey their locations and that are timestamped with the *sending time* t_1 , t_2 , and t_3 (Appendix A provides an **index of symbols** used in this paper), respectively. Upon receiving the first signal (say the one from beacon 1), the node records the location information and “aligns” its local clock with the timestamp (as shown by the aligned time line in in

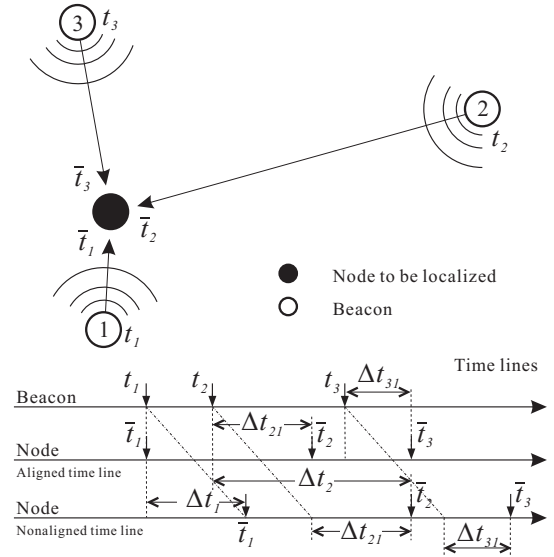


Fig. 1. Principles of “reversed” TDoA. An event time on a time line is defined according to the local clock of a beacon or of the node.

Fig. 1).⁵ When the signals from beacon 2 and 3 arrive at the node, the quantities $\bar{t}_i - t_i$, $i = 2, 3$ (where \bar{t}_i stands for an *arrival time*) characterize the difference in distances from beacons to the node. With enough measures of $\bar{t}_i - t_i$, The node is able to infer its own location (we refer to Section III for details).

There are two severe weaknesses for the above mechanism. First, in a large computing environment where nodes are deployed in an ad hoc fashion, it might not be reasonable to assume the existence of a precisely synchronized beacon group. Secondly, if obstacles exist in between a node and certain beacons, this node might not obtain enough time difference measurements to estimate its coordinates. By introducing a **mobile** beacon to replace multiple static beacons, we avoid the above weaknesses altogether. The beacon, on one hand, is bound to “synchronize” with itself, and, on the other hand, can always find a position to bypass the obstacles unless a node is totally isolated. Since this mechanism differentiates the arrival times through beacon mobility, we term it *mobility-differentiated time-of-arrival* (MDToA).

Exploiting the difference of arrival times (of the same type of signal), MDToA brings two additional benefits. First, this mechanism is more efficient than other mobile beacon approaches based on RSSI (e.g., [17], [20]), because, as shown in Section III, only a few observations and simple calculations are necessary to obtain the location. Secondly, it does not need more than

⁴Using a ring of charge-coupled devices (CCDs) that detects light at certain frequencies could be a cheaper way to estimate AoA [16], but how to engineer such a system is still an open question.

⁵As far as there is a non-zero distance between beacon 1 and the node, this clock actually lags behind the clock of beacon 1.

one transceiver, which facilitates the low-power design of sensor nodes.

Now, we formally define the way to measure time differences and the related errors. We take an alternative approach without clock alignment (see the nonaligned time line in Fig. 1); this approach makes it easy to construct estimators. If there is no clock drift and the ToA measurements are perfect, a time difference Δt_{i1} is defined as:

$$\Delta t_{i1} \equiv \Delta t_i - \Delta t_1 = (\bar{t}_i - t_i) - (\bar{t}_1 - t_1) \quad (1)$$

for $i \neq 1$. In reality, the frequency differences of the crystals used by sensor nodes and beacons introduce non-negligible clock drift (e.g., up to $40\mu\text{s}$ per second for Mica2 motes [31]), and the ToA measurements always involve random errors. Therefore, two error terms are introduced if (1) is used to compute a time difference:

$$\widetilde{\Delta t_{i1}} - \Delta t_{i1} = (\bar{t}_i - \bar{t}_1)t_{drift} + \epsilon_i \quad (2)$$

where $\widetilde{\Delta t_{i1}}$ is the observed value computed with (1) and Δt_{i1} stands for the real value. The first term includes an unknown constant t_{drift} given a pair of node and beacon, and the second term is assumed to be independent Gaussian random variable $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ that represents a ToA measurement error.

If we require the mobile beacon to send a sequence of N signals from a given position i (with $t_i[n]$ and $\bar{t}_i[n]$ being a sending time and an arrival time, respectively, for $n = 0, 1, \dots, N-1$), we have:

$$\widetilde{\Delta t_{i1}}[n] = \Delta t_{i1} + (\bar{t}_i[n] - \bar{t}_1[n])t_{drift} + \epsilon_i \quad (3)$$

Unfortunately, jointly estimating Δt_{i1} and t_{drift} is impractical; we prove this in Appendix B by showing that the achievable *Cramer-Rao lower bound* (CRLB, the lower bound of estimation errors) [32] is too large. As a result, we have to estimate Δt_{i1} through two steps:

- i) Estimating t_{drift} using a sequence of N signals from the **same** beacon position. Since t_{drift} leads to different signal intervals between a node and the beacon, we have:

$$\bar{t}_{i\Delta}[n] - t_{i\Delta}[n] = t_{i\Delta}[n]t_{drift} + \epsilon_i \quad (4)$$

where $\bar{t}_{i\Delta}[n] \equiv \bar{t}_i[n] - \bar{t}_i[0]$ and $t_{i\Delta}[n] \equiv t_i[n] - t_i[0]$. The following efficient estimator can be used:

$$\widehat{t_{drift}} = \frac{\sum_{n=1}^{N-1} (\bar{t}_{i\Delta}[n] - t_{i\Delta}[n])t_{i\Delta}[n]}{\sum_{n=1}^{N-1} (t_{i\Delta}[n])^2} \quad (5)$$

with $\text{var}(\widehat{t_{drift}}) = [\sum_{n=1}^{N-1} (t_{i\Delta}[n]/\sigma_i)^2]^{-1}$. This estimation can be refined at each beacon position.

- ii) Estimating Δt_{i1} with (3) by considering that $t_{drift} \sim \mathcal{N}(\widehat{t_{drift}}, \text{var}(\widehat{t_{drift}}))$ and t_{drift} is independent of ϵ_i .

III. MDToA-BASED COORDINATES ASSIGNMENT

Based on the principles of MDToA described in Section II-D, we propose two sets of simple algorithms for nodes (which receive signals from the mobile beacon) to estimate their coordinates. The first set, which we call *differentiated distance constraints* (DDC), is “oblivious” to the beacon trace; i.e., it only needs a few MDToA measurements but involves somewhat intensive computations.⁶ The second set requires a node to keep track of the changes of the MDToA measurements, such that it can catch the *closest point of approach* (CPA) and identify its coordinates. The latter set of algorithms trade more memory usage for lower computational load. As mentioned in Section II-D, the mobile beacon, upon transmitting a signal, also piggybacks its own coordinates and a timestamp with the transmission. We assume that the mobile beacon always uses a relative coordinate system. As a result, the coordinates estimated by a node are unique only up to an arbitrary translation and rotation. In addition, we assume, in all the discussions hereafter, a 2D space; the extension to 3D space is straightforward.

A. Differentiated Distance Constraints (DDC)

Let us first consider the measurements that a node obtains as a consequence of three subsequent positions of the mobile beacon. As illustrated in Fig. 2 (a), the

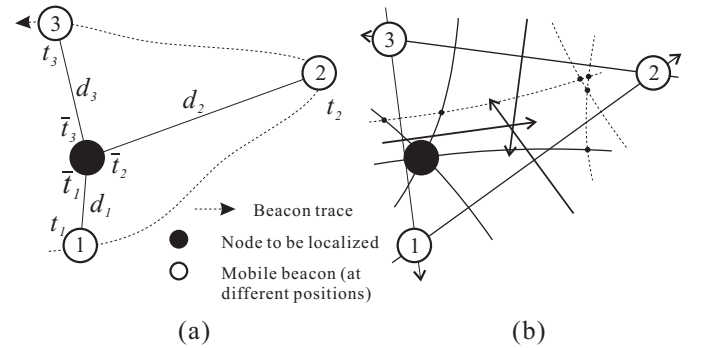


Fig. 2. Differentiated distances introduced by beacon mobility (a) and the corresponding hyperbolic equation system (b).

node can obtain three differentiated distances, which are $r_{21} = d_2 - d_1$, $r_{32} = d_3 - d_2$, and $r_{31} = d_3 - d_1$, with three different positions of the mobile beacon; they are computed as the products of the known signal propagation speed and the estimated time differences Δt_{21} , Δt_{32} , and Δt_{31} . Now, by solving the following

⁶Actually, only several square root computations are needed in the worst case. However, this might be intensive for sensor nodes that have only microcontrollers (e.g., ATmega128 [33]).

system of polynomial equations, the node can obtain its coordinates (x_n, y_n) :

$$\begin{aligned} r_{ij} &= d_i - d_j \\ &= \sqrt{(x_n - x_i)^2 + (y_n - y_i)^2} \\ &\quad - \sqrt{(x_n - x_j)^2 + (y_n - y_j)^2} \end{aligned} \quad (6)$$

where $(x_i \text{ or } j, y_i \text{ or } j)$ are the coordinates of the mobile beacon at position i or j . As illustrated in Fig. 2 (b), the problem of finding coordinates of a point, whose differentiated distances to several fixed points are known, is actually a hyperbolic location problem; a rich body of work (e.g. [34], [35], [36], [37], [38]) has been devoted to this topic. Here we are interested in solutions that require low computational load, because the load is to be taken by sensor nodes themselves. We hereby describe two algorithms, namely *hyperbolae linearization* (HL) and *kite location fix* (KLF), that produce good estimations with simple computations.

1) *Hyperbolae Linearization (HL)*: This approach is based on the idea that two polynomial equations in a form of (6) determine a straight line [37], and each new beacon position adds another (independent) linear equation. Therefore, the node can compute an unambiguous coordinate assignment with 4 beacon positions, and extra information from more beacon positions can be accommodated by computing a *least square* fitting.

Given M beacon coordinates $\mathbf{c}_i = [x_i, y_i]^T$: $i = 1, 2, \dots, M$ and the unknown coordinates $\mathbf{c}_n = [x_n, y_n]^T$ of a sensor node n , we have the following equations according to the *law of Cosines*:

$$2\mathbf{c}_i^T \mathbf{c}_n = R_i^2 + R_n^2 - d_i^2 \quad (7)$$

where $R_i = \|\mathbf{c}_i\|$ and $d_i = \|\mathbf{c}_i - \mathbf{c}_n\|$. Now if we choose beacons 1 (\mathbf{c}_1) and 2 (\mathbf{c}_2) as reference points and consider an arbitrary $i : i \neq 1, 2$, we have the following equations by subtracting (7) of 2 and i to that of 1 and by also taking into account that $r_{ij} = d_i - d_j$:

$$\begin{aligned} \mathbf{C}_{\Delta i} \mathbf{c}_n &= \mathbf{R}_{\Delta i} - d_1 \mathbf{r}_i \\ \mathbf{C}_{\Delta i} &= \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_i - x_1 & y_i - y_1 \end{bmatrix} \\ \mathbf{R}_{\Delta i} &= \frac{1}{2} \cdot \begin{bmatrix} R_2^2 - R_1^2 - r_{21}^2 \\ R_i^2 - R_1^2 - r_{i1}^2 \end{bmatrix} \\ \mathbf{r}_i &= \begin{bmatrix} r_{21} \\ r_{i1} \end{bmatrix} \end{aligned} \quad (8)$$

Eliminating d_1 from (8) for $i = 3, 4, \dots, M - 1$, we obtain a linear system of $M - 2$ equations in \mathbf{c}_n

$$[\mathbf{A} \mid \mathbf{\Omega}] \mathbf{c}_n = \mathbf{B} \quad (9)$$

where

$$\begin{aligned} \mathbf{A}[i] &= r_{i1}(x_2 - x_1) - r_{21}(x_i - x_1) \\ \mathbf{\Omega}[i] &= r_{i1}(y_2 - y_1) - r_{21}(y_i - y_1) \\ \mathbf{B}[i] &= \frac{1}{2} \cdot [r_{i1}(R_2^2 - R_1^2 - r_{21}^2) - r_{21}(R_i^2 - R_1^2 - r_{i1}^2)] \end{aligned}$$

The least square solution of (9) is

$$\widehat{\mathbf{c}}_n = ([\mathbf{A} \mid \mathbf{\Omega}]^T [\mathbf{A} \mid \mathbf{\Omega}])^{-1} [\mathbf{A} \mid \mathbf{\Omega}]^T \mathbf{B} \quad (10)$$

A special case is $M = 4$, where $[\mathbf{A} \mid \mathbf{\Omega}]$ becomes nonsingular and thus $[\mathbf{A} \mid \mathbf{\Omega}]^{-1} \mathbf{B}$ gives the solution. Note that the least square solution can be computed sequentially [32], i.e., the estimation $\widehat{\mathbf{c}}_n$ can be updated for each new beacon position \mathbf{c}_i without having to resolve (10).

2) *Kite Location Fix (KLF)*: Basically, solving a polynomial equation system (6) with 2 equations gives an estimation of \mathbf{c}_n (we continue with the same notation as in III-A.1). However, the accuracy of the estimation depends heavily on the positions of the beacons. The errors of the estimation are determined not only by the errors of MDToA estimations but also by the *geometric dilution of precision* (GDOP) introduced by the geometries of the beacon positions. The larger the GDOP, the more sensitive the estimation accuracy is to the input (i.e., MDToA estimations) errors. Intuitively, the GDOP is minimized if the two hyperbolae are orthogonal at the intersection points and it increases with a decreasing angle between the two tangents. For example, in Fig. 3 (a), the two right node locations have much larger GDOPs than the two on the left. We omit the detailed analysis, which can be done by investigating the CRLB. Usually,

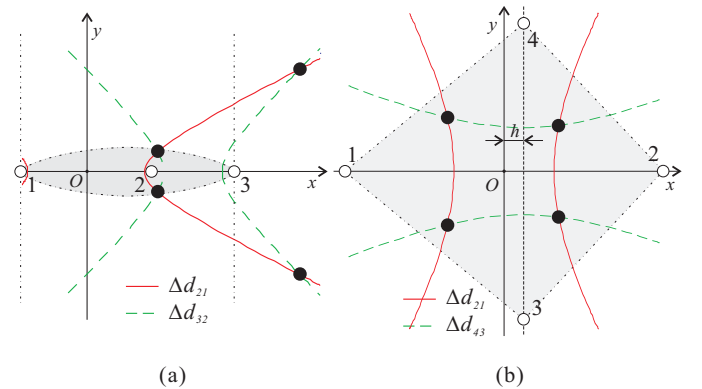


Fig. 3. Hyperbola intersection problem of two hyperbolae. (a) For 3 collinear beacon positions, the effective region is a lens belt centered around the line. (b) With 4 beacons at the vertices of a kite, the effective region is the kite.

we would like to have a large *effective region* (the region where GDOP is minimized) with a small number of beacon positions. In this work, we propose to have the

beacon positions at the vertices of a *kite*, as shown in Fig. 3 (b). By choosing a proper coordinate system, the 2 measurements r_{21} and r_{43} along with the coordinates of the 4 beacon positions define two hyperbola equations:

$$\frac{x_n^2}{r_{21}^2} - \frac{y_n^2}{d_{21}^2 - r_{21}^2} = 1 \quad (11)$$

$$\frac{y_n^2}{r_{43}^2} - \frac{(x_n - h)^2}{d_{43}^2 - r_{43}^2} = 1 \quad (12)$$

where d_{ij} is the distance between position i and j . Combining the above two equations and solving a quadratic equation, the node obtains the following coordinate assignments:

$$\widehat{x}_n = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (13)$$

$$\widehat{y}_n = \pm \sqrt{(d_{21}^2 - r_{21}^2) \left(\frac{\widehat{x}_n^2}{r_{21}^2} - 1 \right)} \quad (14)$$

$$\text{where } A = \left(\frac{d_{21}^2 - r_{21}^2}{r_{21}^2} - \frac{r_{43}^2}{d_{43}^2 - r_{43}^2} \right)$$

$$B = 2h \frac{r_{43}^2}{d_{43}^2 - r_{43}^2}$$

$$C = -r_{43}^2 \left(\frac{h^2}{d_{43}^2 - r_{43}^2} + 1 \right) - (d_{21}^2 - r_{21}^2)$$

The signs of \widehat{x}_n and \widehat{y}_n are determined by the signs of r_{21} and r_{43} .

B. Closest Point of Approach (CPA)

Given a curve and an off-curve point, the *closest point of approach* (CPA) on the curve is a point whose distance to the off-curve point is minimized. We can also define the *farthest point of approach* (FPA) in a similar way. A node that receives signals sent from a mobile beacon moving on such a curve can detect a CPA (or FPA) by keeping track of the MDToA estimations, thanks to the monotonic changes of the distance. For some specific curves (which we term *CPA-compatible*), nodes that capture a certain number of CPAs (or FPAs) can immediately identify their coordinates with virtually no computation. As shown in Fig. 4, moving a beacon along two adjacent sides of a rectangle leads to two CPAs for a given static node (a) and moving a beacon along a circle results in one CPA and one FPA (b). The former provides the coordinates $(x_n, y_n) = (x_{CPA_x}, y_{CPA_y})$ of the node under Cartesian coordinates, the latter indicates the coordinates $(d_n, \theta_n) = (\frac{1}{2}r_{FPA,CPA}, \theta_{CPA})$ of the node under polar coordinates.⁷ In this work, we only

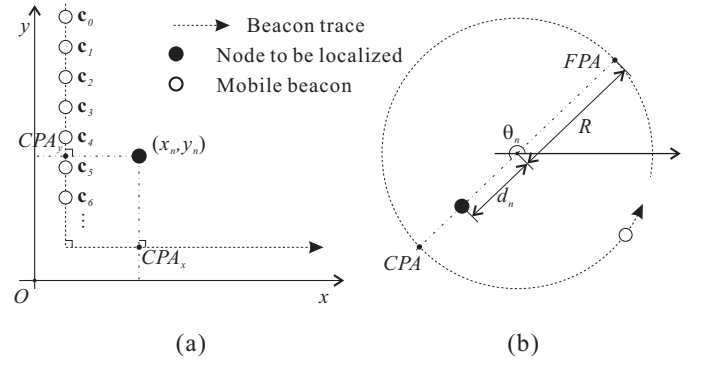


Fig. 4. Techniques relying on *closest point of approach* (CPA). (a) Under Cartesian coordinates, moving a beacon along two adjacent sides of a rectangle leads to two CPAs. (b) Under polar coordinates, moving a beacon along a circle results in one CPA and one *farthest point of approach* (FPA).

consider the case with Cartesian coordinates and directly choose the curve illustrated in Fig. 4 (a).

There is a problem with this seemingly simple solution. Although the mobile beacon can keep moving in a continuous manner, the beacon signals cannot be transmitted continuously (otherwise the receiver could not distinguish between consecutive beacons). As a result, the beacon trace is “chopped” into discrete beacon points. Now a CPA for a given node could be introduced between two beacon points with high probability, which introduces an error into the coordinates assigned to the node. We hereby discuss several estimation algorithms and their corresponding errors, assuming that a node begins to track the MDToA estimations with respect to a reference beacon position (say c_0 in Fig. 5).

- *MaxDiff*: Take the beacon position with the largest MDToA estimation as the CPA (e.g., c_k). The magnitude of the error is determined by the length of the beacon interval close to the CPA (i.e., between c_j and c_k).
- *MinDiff*: This method searches for a pair of *quasi-symmetric* (in terms of the CPA) beacon positions. The criterion is that the difference of their MDToA

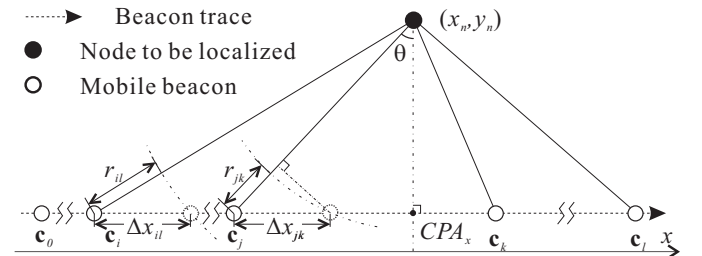


Fig. 5. Illustrations of CPA algorithms.

estimations are minimized. Let us consider such

⁷ $r_{FPA,CPA} = d_{FPA} - d_{CPA} = (R + d_n) - (R - d_n) = 2d_n$

a pair \mathbf{c}_j and \mathbf{c}_k in Fig. 5. If we use r_{jk} to approximate Δx_{jk} , we have $\widehat{x}_n \approx \frac{x_j + r_{jk} + x_k}{2}$. We could also consider several quasi-symmetric pairs (e.g., \mathbf{c}_i and \mathbf{c}_l is another pair). It is straightforward to see that the estimation error decreases with an increasing distance between the nodes of the pairs, because, for example, $\Delta x_{il} - r_{il} < \Delta x_{jk} - r_{jk}$.

- *Symmetrical Interpolation*: This method refines the estimations of MinDiff by finding the mirror position (in terms of the CPA) of the MaxDiff position and taking an average of them. As shown in Fig. 5, the mirror position can be estimated with the known MDToA estimation (which results in r_{jk}) and the MinDiff estimations of $(\widehat{x}_n, \widehat{y}_n)$ that are used to compute θ . The estimation can be refined iteratively.

In most cases, simple calculations (MaxDiff or MinDiff) can already lead to satisfactory accuracy. We refer to Section V-C for examples.

IV. NEOS-MDToA: ENABLING LOCATION SURVEYING WITH MOBILITY

Equipped with the MDToA-based localization algorithms, we are now ready to present our Non-intERactive lOcation Surveying (NEOS) approach. We consider a mobile node that is able to emit **one** analogue signal. The signal can be one of the following: RF, infrared, acoustic, and ultrasound; the choice depends on the requirement of applications (e.g., RF for terrestrial networks and acoustic for underwater facilities). For the static nodes to be localized, we assume that they are equipped with a clock whose resolution matches the propagation speed of the signal sent from the mobile beacon.⁸ The computational capabilities of the static nodes are heterogeneous; they independently choose computationally affordable algorithms for their coordinate estimations. The problem we are facing is to select a proper trace for the mobile beacon such that all nodes can obtain adequate MDToA estimations to compute their coordinate assignments, regardless of the estimation algorithms they choose. In the following subsections, we first identify the beacon positions critical to the DDC algorithms (Section III-A). Then we design, for the mobile beacon, proper moving traces that go through these critical positions and are also CPA-compatible (Section III-B). We also discuss practical issues at the end.

⁸The resolution should be in the order of nanosecond if the RF signal is used. According to PinPoint's 3D-iD system [39], a 40MHz clock rate (25ns resolution) may lead to a systematic error up to 7.5m in our case. We expect future technology to provide sensor nodes with an adequate level of timing resolution.

A. Pinpoint Critical Positions with CRLB

Critical beacon positions are those, when being added, that can significantly improve the coverage of the effective region (defined in Section III-A.2) or the estimation accuracy within the effective region. Given an estimation algorithm, its error characteristics (as a function of beacon positions) indicate the critical positions. However, the error characteristics of the *ad hoc* algorithms that we described in Section III, even if obtained through laborious inductions, could be difficult to interpret, due to the complicated correlations between inputs. Therefore, we turn to the CRLB (Cramer-Rao lower bound). Since CRLB bounds from below the estimation variance given the same inputs to our algorithms, it can indicate the critical positions for the optimistic case. We only report the results that are obtained from the analysis; detailed inductions on CRLB are given in Appendix C.

Inspired by the work of Savvides et al. [40], we first put four beacon positions at the vertices of a square, then we investigate the effect of adding beacon positions, as well as moving the four original positions. The contours of the *root mean square* (RMS) errors under the same input to the DDC-HL approach (Section III-A.1) are drawn in Fig. 6. We have also obtained results similar to Fig. 6 (a) and (f) for the DDC-KLF approach (Section III-A.2). Based on this analysis, we make the following observations:

- The effective region is roughly within the convex hull;⁹ reducing the effective region by moving beacon positions closer to each other leads to larger errors (comparing (a) with (b)). This result is similar to what was obtained in [40], but the inputs to the problem are quite different.
- The initial four beacon positions are critical; adding extra positions can gradually improve the accuracy, but removing one of the initial positions dramatically deteriorates the estimation results (e.g., (e)). Actually, the other (not plotted in Fig. 6) results for only three beacon positions show that no meaningful estimation can be made.
- Shifting the beacon positions without changing the area size within the convex hull may only result in slight differences. For example, both (f) and (g) exhibit somewhat enlarged effective regions but with reduced accuracy.

Note that, although increasing distances between beacon positions leads to better estimation accuracy and a larger effective region, these distances are also constrained by the transmission range of the beacon.

⁹Although the variances within the convex hull are not constant, they vary negligibly. So we consider them all minimum.

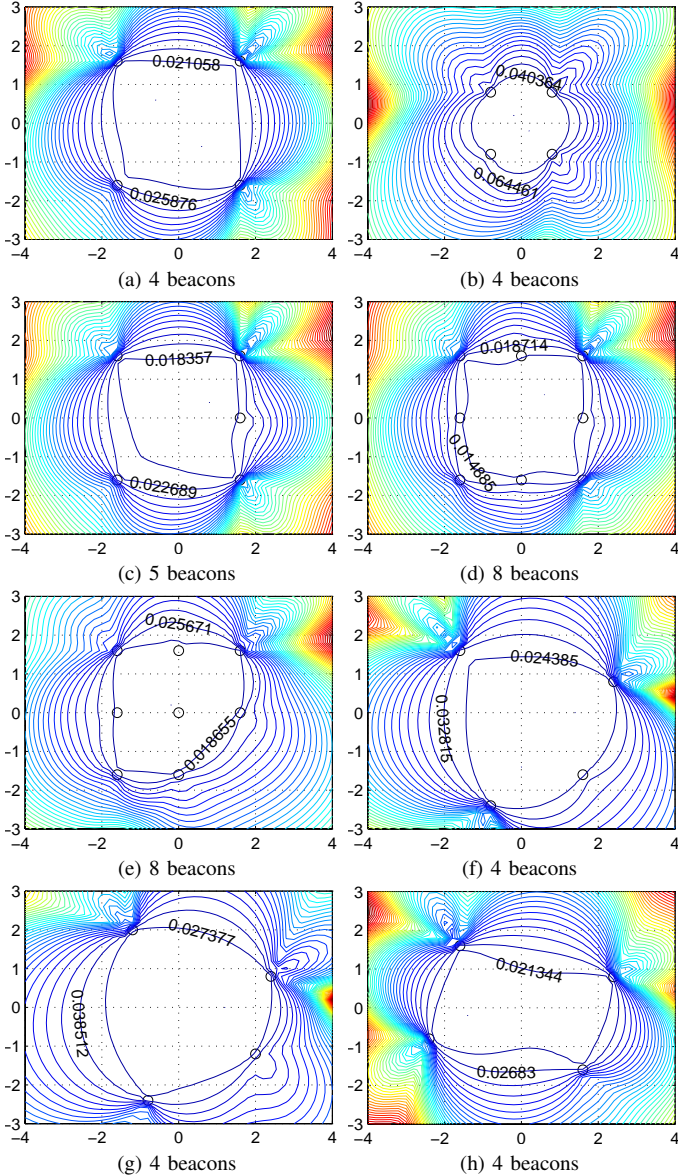


Fig. 6. The contours of root mean square (RMS) errors resulting from our CRLB analysis (assuming a RMS error of 60ns for all MDToA estimations). The dimension of each figure, as well as errors, are measured in meter. (a) 4 beacon positions at the vertices of a square. (b) Reducing the square size. (c) and (d) Adding beacon positions at the midpoints of the square edges. (e) Moving one vertex position to the center. (f) Shifting one diagonal along the other. (g) Shifting both diagonals. (h) Shifting two diagonal positions along opposite directions (i.e., 4 beacons at the vertices of a parallelogram).

B. Mobility Strategies

Given a known¹⁰ sensor deployment area as shown in Fig. 7 (a), the trace of a mobile beacon is subject to the following constraints imposed by different algorithms:

- It should pass all the critical beacon positions. According to Section IV-A, these positions are vertices of quadrilaterals that are shifted versions of a square.

¹⁰The geographical information does not need to be exact.

- In order to be CPA-compatible, the trace should parallel the axes of a given Cartesian coordinate system, as shown in Section III-B.

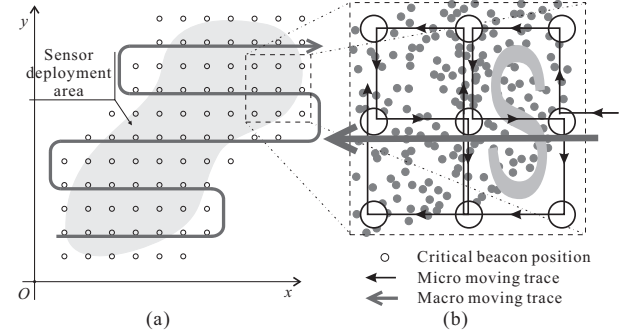


Fig. 7. Mobility strategies. (a) Sensor deployment area and macro-“S” beacon trace. (b) micro-“S” trace in detail.

Based on the above constraints, we suggest an “S” beacon trace illustrated in Fig. 7. A given deployment area is first divided into small squares defined by several critical beacon positions (a). The area size of each square is determined by the transmission range of the mobile beacon. In the extreme case where the whole deployment area is within the transmission range, one square is enough to cover the area. The mobile beacon winds around two adjacent squares following the micro trace (b) to enable observations of MDToA; the trace is designed to shorten $\bar{t}_i[n] - \bar{t}_1[n]$ in (3) (Section II-D) for higher estimation accuracy. When moving along the trace, the mobile beacon sends beacon signals periodically, with the signal sent at a critical position being specially tagged. Note that nodes that take the DDC-KLF approach have to perform a (45°) axis rotation before and after their coordinate computations.

C. Implementation Issues

We now discuss four practical aspects related to the implementation of NEOS: location awareness, energy saving, adaptive mobility control, and integration with interactive (IT) location algorithms.

1) *Location-Aware Mobile Beacon*: NEOS requires the mobile beacon be location-aware. We can meet this need by relying on a location system (e.g., GPS) and accurate odometer+compass. While the location system provides the beacon with coordinates of important positions (e.g., the starting point), the odometer and compass take care of the rest. The choice of a specific location system may vary with different application scenarios. For example, a GPS receiver does not work for underwater devices. The location information acquired through the above combination may not be perfect; we leave it for

future work to address the coordinate estimation errors introduced by the uncertainty of beacon positions.

2) *Energy Saving*: The non-interactive feature of NEOS requires no information to be sent by sensor nodes; this can reduce the energy consumption of nodes. However, since the receiving power can be as significant as transmission power, a node has to switch off its transceiver whenever appropriate in order to achieve energy saving. Nodes that apply DDC algorithms may duty cycle their transceivers through periodical sleeps and wakes. The CPA algorithms require the nodes to stay awake during certain periods, at any other time the nodes can totally shut down.

3) *Adaptive Mobility Control*: The information we have on the sensor deployment area may be too coarse in certain scenarios. This would require the mobile beacon to be navigated. If we relax the non-interactive constraint on NEOS, the already localized nodes can guide the mobile beacon to further localize other nodes. Since the solution involves other aspects of sensor networks (e.g., navigation guide [1]) that are beyond the scope of our paper, we refer to the corresponding literature for details.

4) *Integration with IT*: There could be cases where the beacon trace cannot cover the whole network, because of, for instance, hostile conditions. A solution could be the combination of NEOS and an IT algorithm: NEOS tries to localize as many nodes as possible; the IT will do the rest using nodes with known coordinates as anchors.

V. EXPERIMENTS

In this section, we report our preliminary experiment results. These results are preliminary because we have not implemented a mobile beacon that moves automatically (we actually move it manually). We make use of an existing device (i.e., CricketMote [41], a Crossbow[®] implementation of MIT Cricket [26]), to emulate the mobile beacon we have in mind. The reason we consider CricketMote is twofold: (i) it is one of the few commercial off-the-shelf sensor nodes that are equipped with ultrasound transceiver; ultrasound has a propagation speed that matches the microsecond clock resolution achieved by similar devices and (ii) we need the RF transceiver of CricketMote to send coordinates and timestamps of the mobile beacon; otherwise a special modem would be necessary to transmit information using ultrasound, which is beyond the scope of this paper.

Our experiment setup, shown in Fig. 8, consists of a mobile beacon (a CricketMote) and a listener (the node to be localized). The listener comprises 4 CricketMotes bound together. This provides us with an omnidirectional ultrasound receiver that mimics UCLA *Medusa* nodes

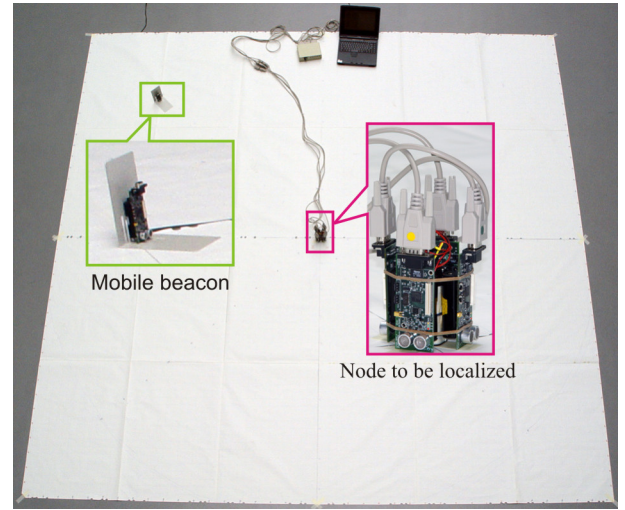


Fig. 8. Experiment setup. Both the mobile beacon and the node to be localized are amplified to show the details.

[8]. A side effect of this design is that we cannot require the listener to estimate its coordinates; we have to retrieve the MDToA readings from the listener through serial cables and compute the coordinates in a laptop. Although we have only one node to be localized, we put it at different positions to test our algorithms. Since, due to the non-interactive feature of our algorithms, the node density does not matter, our experiments perfectly emulate the case where several nodes at different positions estimate their coordinates concurrently.

We perform our experiments in a square area of $380\text{cm} \times 380\text{cm}$ (the white area shown in Fig. 8), and we assume a coordinate system with its origin at the square center and its axes paralleling the square sides. For the DDC approaches, we have 4 critical beacon positions at $(180, 180)$, $(180, -180)$, $(-180, -180)$, and $(-180, 180)$ (which work for both DDC-HL and DDC-KLF) and 4 accessory positions at $(180, 0)$, $(0, -180)$, $(-180, 0)$, and $(0, 180)$ (which help to refine the estimations of DDC-HL). For the CPA approaches, we move the beacon along two sides of the square and set the interval between two consecutive beacon positions to be 20cm. Due to the symmetry of the beacon positions, it makes sense to test our algorithms only in quadrant 1 (where both x and y coordinates are positive). Within quadrant 1, we put the listener at the following 12 positions (which are marked as “real coordinates” in the later plots): $(160, 160)$, $(0, 80)$, $(12.5, 160)$, $(13.5, 15)$, $(37.5, 50)$, $(42.5, 127.5)$, $(70, 145)$, $(75, 90)$, $(80, 12.5)$, $(117.5, 65)$, $(160, 0)$, $(160, 80)$, and compare with them the estimated coordinates resulting from different algorithms. The root mean square (RMS) error of MDToA estimations obtained in the experiments is 60ns (equivalent to a 2.2cm RMS error in differentiated distance measurements).

A. Hyperbolae Linearization

The experiment results of using DDC-HL are shown in Fig. 9. Only one critical beacon position (180, 180) is plotted, because the figure shows quadrant 1. An accessory beacon position at (180, 0) is used to refine the estimation. As we can see from the Fig. 9, five beacon positions are already enough to obtain a very good estimation (see the marked location errors).

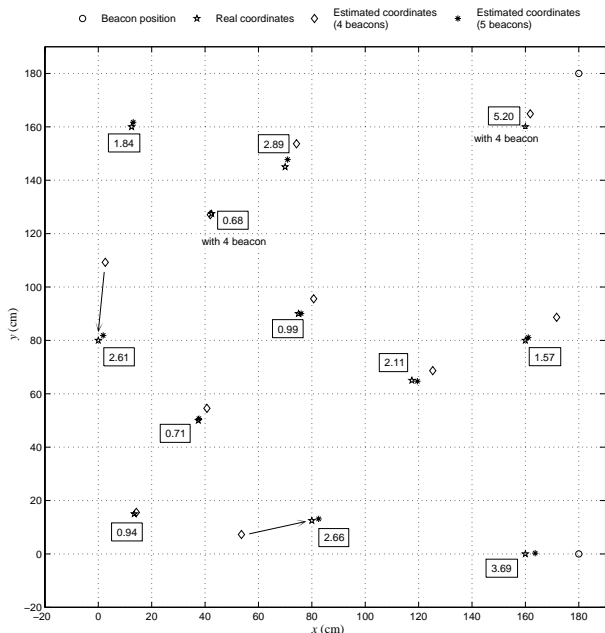


Fig. 9. Coordinate estimations using DDC-HL for 12 nodes with four or five beacon positions. Location errors are marked in centimeter.

In general, the estimations with only four beacon positions have relatively large errors. There are some positions whose 4-beacon estimations are not plotted, because they are simply out of quadrant 1. The main reason for this is that linearization dilutes information (e.g., the sign of the differentiated distances). Fortunately, adding only a single beacon rectifies the errors sufficiently. DDC-HL has the advantage of accommodating additional information, which makes it more suitable for the case where MDTa errors are large.

B. Kite Location Fix

Fig. 10 shows the results of using DDC-KLF. With only four critical beacon positions, DDC-KLF already achieves an accuracy very close to what DDC-HL obtains with five beacons. The reason is twofold: (i) DDC-KLF maintains the non-linearity of the original problem without diluting information and (ii) it maximizes the information extracted from the four beacons by making use of the differentiated distances between opposite corners. However, the algorithm requires the

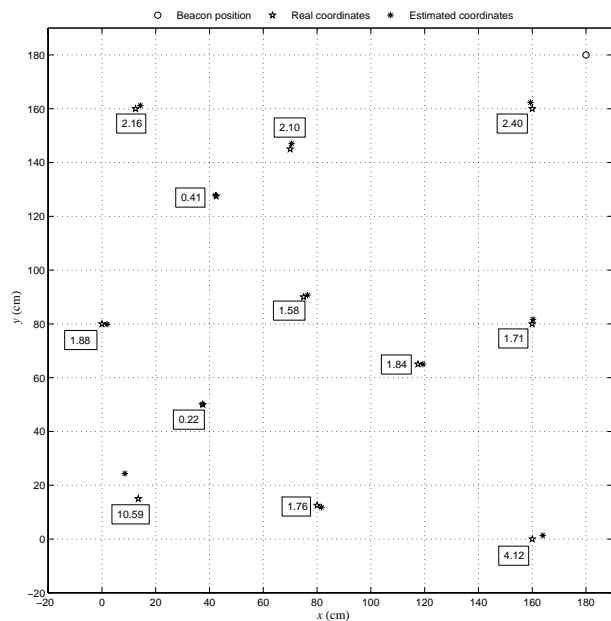


Fig. 10. Coordinate estimations using DDC-KLF for 12 nodes with four beacon positions. Location errors are marked in centimeter.

computation of square roots, which can be undesirable for some sensor nodes. In addition, DDC-KLF cannot accommodate additional information.

C. Closest Point of Approach

As shown by Fig. 11, the CPA algorithms provide less

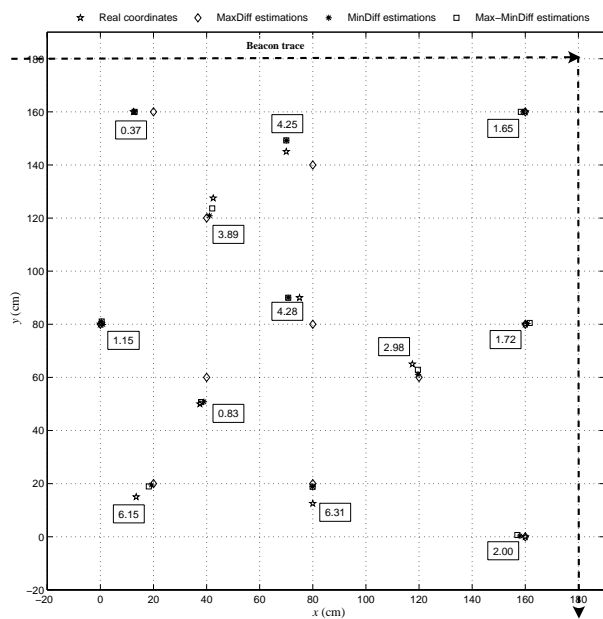


Fig. 11. Coordinate estimations using CPA for 12 nodes. Location errors are marked in centimeter for the Max-MinDiff.

accurate estimations than the DDC algorithms. There is no surprise here because CPA algorithms involve barely any computation. We could have obtained better results

with the symmetrical interpolation algorithm, but the strength of a CPA algorithm is its computation-free property. The Max-MinDiff algorithm, whose performance is also reported in Fig. 11, is a special case of MinDiff: the quasi-symmetric pair (see Section III-B) that is the farthest apart is chosen. If the coordinates of a node coincide with those of certain beacons, MaxDiff gives the perfect estimations, while MinDiff and Max-MinDiff bring small biases. In other cases, Max-MinDiff always perform better than others (actually, MinDiff does not differ from it too much). The worst case happens when the node is far from the beacon trace (e.g., the one at (13.5, 15)), because the distance changes become less sensitive to the beacon mobility and thus more sensitive to MDToA errors.

VI. RELATED WORK

We discuss, in this section, related topics that are not covered in Section I and II.

Location surveying approaches for sensor networks can also be categorized according to whether ranging techniques are used. Range-based approaches (e.g., [8], [9], [10], [12], [16], [17], [19], [20], [22]) usually apply one of the ranging techniques discussed in Section II. Their errors are measured in absolute units (e.g. cm). Range-free approaches (e.g. [6], [7], [11], [13], [14], [15], [18], [21], [23]) rely on proximity information such as radio connectivity. Relative units (e.g., percentage of the radio range) are commonly used for error measurement. Range-free approaches are often approximations to their counterparts; dedicated analyses on such approximations are also reported in literature (e.g., [42], [43]).

NEOS is also related to source localization using sensor networks with known sensor coordinates (e.g., [44], [2]). In particular, Li et al. [2] apply TDoA-based target tracking technique. NEOS can be considered as the reverse of their work: we use a mobile object with known coordinates to survey the sensor coordinates. Galstyan et al. [45] present a joint location survey and target tracking approach. Their proposal needs assistance from anchors and its performance is significantly improved with a high percentage of anchors. Source localization algorithms (apart from those assuming a priori knowledge about the optimal sensor positions [46]) can use the coordinate estimations from NEOS as their inputs.

A large body of work has investigated hyperbolic localization. Existing solutions are either iterative methods [47], [48] or in closed-form [34], [35], [36], [37], [38]. The DDC algorithms used by NEOS follow the line of closed-form solutions. Our approach differs from previous solutions in that we propose simple algorithms that are computationally affordable by sensor nodes.

VII. CONCLUSION

In this paper, we have presented a location surveying system for sensor networks. Our NEOS (Non-intEreactive Iocation Surveying) approach relies on a mobile beacon; the beacon provides static sensor nodes with MDToA (Mobility-Differentiated Time of Arrival) observations, and thus enables them to estimate their coordinates without interactions with either the beacon or other nodes. Exploiting the multiple MDToA (a special form of TDoA) observations acquired by a node, we have proposed simple algorithms to perform coordinate estimations. These algorithms either impose a modest computation load or are almost computation-free. Nodes in a network, whose processing powers are probably heterogeneous, can choose algorithms according to their processing constraints. We have also implemented a preliminary prototype of NEOS. Experiments with this prototype demonstrate promising location accuracy.

Compared with the existing mobile-aided approaches, our NEOS has the advantages of (i) higher accuracy, due to the use of a more robust signal propagation property (i.e., TDoA instead of RSSI) and (ii) less complexity and thus less energy consumption, due to its non-interactive feature. In the networks where location-aware mobile beacons are applicable, NEOS can perform better than existing interactive (IT) approaches, again due to its lower demand in terms of both computation and communication resources.

We plan to further improve our prototype and make it autonomous. Full-scale field tests with the prototype would be useful to confirm the strength of NEOS. We will also explore the possibility of integrating NEOS with IT location systems for higher scalability.

REFERENCES

- [1] Q. Li, M.D. Rosa, and D. Rus, "Distributed Algorithms for Guiding Navigation across a Sensor Network," in *Proc. of the 9th ACM MobiCom*, 2003.
- [2] T. Li, A. Ekpenyong, and Y.-F. Huang, "A Location System Using Asynchronous Distributed Sensors," in *Proc. of the 23rd IEEE INFOCOM*, 2004.
- [3] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proc. of the 6th ACM MobiCom*, 2000.
- [4] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in Wireless Ad-Hoc Sensor Networks," in *Proc. of the 7th ACM MobiCom*, 2001.
- [5] J. Scott and M. Hazas, "User-Friendly Surveying Techniques for Location-Aware Systems," in *Proc. of the 5th UbiComp*, 2003.
- [6] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization For Very Small Devices," *IEEE Personal Communication*, vol. 7, no. 5, pp. 28–34, 2000.
- [7] L. Doherty, K.S.J. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," in *Proc. of the 20th IEEE INFOCOM*, 2001.

- [8] A. Savvides, C. Han, and M.B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proc. of the 7th ACM MobiCom*, 2001.
- [9] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free Position in Mobile Ad-Hoc Networks," *Kluwer Cluster Computing Journal*, vol. 5, no. 2, pp. 157–167, 2002.
- [10] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in *Proc. of the 22nd IEEE INFOCOM*, 2003.
- [11] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Networks," in *Proc. of the 2nd IPSN*, 2003.
- [12] K. Römer, "The Lighthouse Location System for Smart Dust," in *Proc. of the 1st ACM MobiSys*, 2003.
- [13] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-free Localization Schemes for Large Scale Sensor Networks," in *Proc. of the 9th ACM MobiCom*, 2003.
- [14] N. Bulusu, J. Heidemann, D. Estrin, and T. Tran, "Self-Configuring Localization Systems: Design and Experimental Evaluation," *ACM Trans. on Embedded Computing Systems*, vol. 3, no. 1, pp. 24–60, 2004.
- [15] Y. Shang and W. Ruml, "Improved MDS-Based Localization," in *Proc. of the 23rd IEEE INFOCOM*, 2004.
- [16] K.K. Chintalapudi, A. Dhariwal, R. Govindan, and G. Sukhatme, "Ad-Hoc Localization Using Ranging and Sectoring," in *Proc. of the 23rd IEEE INFOCOM*, 2004.
- [17] M. Sichertiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," in *Proc. of the 1st IEEE MASS*, 2004.
- [18] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," in *Proc. of the 10th ACM MobiCom*, 2004.
- [19] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," in *Proc. of the 2nd ACM SenSys*, 2004.
- [20] P.N. Pathirana, N. Bulusu, A.V. Savkin, and S. Jha, "Node Localization Using Mobile Robots in Delay-Tolerant Sensor Networks," *IEEE Trans. on Mobile Computing*, vol. 4, no. 3, pp. 285–296, 2005.
- [21] H. Lim and J.C. Hou, "Localization for Anisotropic Sensor networks," in *Proc. of the 24th IEEE INFOCOM*, 2005.
- [22] N.B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," in *Proc. of the 24th IEEE INFOCOM*, 2005.
- [23] D. Goldenberg, A. Krishnamurthy, W.C. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B.D.O. Anderson, "Network Localization in Partially Localizable Networks," in *Proc. of the 24th IEEE INFOCOM*, 2005.
- [24] J. Hightower and G. Borriello, "A Survey and Taxonomy of Location Systems for Ubiquitous Computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [25] A. Harter, A. Hopper, P. Steggle, A. Ward, and P. Webster, "The anatomy of a context-aware application," in *Proc. of the 5th ACM MobiCom*, 1999.
- [26] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proc. of the 6th ACM MobiCom*, 2000.
- [27] A. LaMarca, W. Brunette, D. Koizumi, M. Lease, S. Sigurdsson, K. Sikorski, D. Fox, and G. Borriello, "Making Sensor Networks Practical with Robots," in *LNCS*, F. Mattern and M. Naghshineh, Eds., pp. 152–166. Springer-Verlag, 2002.
- [28] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," in *Proc. of the 19th IEEE INFOCOM*, 2000.
- [29] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 2002.
- [30] N.B. Priyantha, A. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," in *Proc. of the 7th ACM MobiCom*, 2001.
- [31] "MPR/MIB User's Manual," http://www.xbow.com/Products/Wireless_Sensor_Networks.htm.
- [32] S.M. Kay, *Fundamentals of Statistical Signal Processing - Estimation Theory*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [33] "8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash," <http://www.atmel.com/products/avr/>.
- [34] M.A. Spirito, "Accuracy of Hyperbolic Mobile Station Location in Cellular Networks," *IEE Electronics Letters*, vol. 37, no. 11, pp. 708–710, 2001.
- [35] Y.T. Chan and K.C. Ho, "A Simple and Efficient Estimator for Hyperbolic Location," *IEEE Trans. on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, 1994.
- [36] B.T. Fang, "Simple Solution for Hyperbolic and Related Position Fixes," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 26, no. 5, pp. 748–753, 1990.
- [37] B. Friedlander, "A Passive Localization Algorithm and Its Accuracy Analysis," *IEEE Journal of Oceanic engineering*, vol. 12, no. 1, pp. 234–245, 1987.
- [38] H.C. Schau and A.Z. Robinson, "Passive Source Localization Employing Intersecting Spherical Surfaces from Time-of-Arrival Differences," *IEEE Trans. on Acoust., Speech, Signal Processing*, vol. 34, pp. 1223–1225, 1987.
- [39] J. Werb and C. Lanzi, "Designing a Positioning System for Finding Things and People indoors," *IEEE Spectrum*, vol. 35, no. 9, pp. 71–78, 1998.
- [40] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M.B. Srivastava, "On the Error Characteristics of Multihop Node Localization in Ad-Hoc Sensor Networks," in *Proc. of the 2nd IPSN*, 2003.
- [41] "Cricket User's Manual," http://www.xbow.com/Support/Support_pdf_files/MIT_Cricket_Manual_7430-0335-01_A.pdf.
- [42] R. Bischoff and R. Wattenhofer, "Analyzing Connectivity-Based Multi-Hop Ad-hoc Positioning," in *Proc. of the 2nd IEEE PerCom*, 2004.
- [43] S. Vural and E. Ekici, "Analysis of Hop-Distance Relationship in Spatially Random Sensor Networks," in *Proc. of the 6th ACM MobiHoc*, 2005.
- [44] J.C. Chen, L. Yip, J. Elson, H. Wang, D. Maniezzo, R.E. Hudson, K. Yao, and D. Estrin, "Coherent Acoustic Array Processing and Localization on Wireless Sensor Networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1154–1162, 2003.
- [45] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattern, "Distributed Online Localization in Sensor Networks Using a Moving Target," in *Proc. of the 2nd IPSN*, 2003.
- [46] S. Ray, R. Ungrangsi, F. de Pellegrini, A. Trachtenberg, and D. Starobinski, "Robust Location Detection in Emergency Sensor Networks," in *Proc. of the 22nd IEEE INFOCOM*, 2003.
- [47] W.H. Foy, "Position-location solutions by Taylor-series estimation," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 12, no. 2, pp. 187–194, 1976.
- [48] D.J. Torrieri, "Statistical theory of passive location systems," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 20, no. 2, pp. 183–198, 1984.

APPENDIX

A. Symbol Index

t_i, \bar{t}_i	Beacon's and node's clock reading for position i .
Δt_i	$\Delta t_i = \bar{t}_i - t_i$
Δt_{ij}	The MDT to A between to position i and j .
$\widetilde{\Delta t}_{ij}$	The observed value of Δt_{ij} .

\hat{x}	Estimated value of a quantity x .
$x[n]$	A quantity related to the n th signal sent from the same beacon position.
$x_\Delta[n]$	$x_\Delta[n] \equiv x[n] - x[0]$.
r_{ij}	r_{ij} = signal speed $\times \Delta t_{ij}$ is the differentiated distance between beacon positions i and j .
d_i	The distance between a node and the beacon position i .
d_{ij}	The distance between beacon positions i and j .
\mathbf{c}	$\mathbf{c} = [x, y]^T$ is a vector representing the coordinates of a position (\mathbf{c}_i for a beacon position i and \mathbf{c}_n for a node).

B. CRLB for the Joint Estimation of Δt_{i1} and t_{drift}

The likelihood function for the given observation vector $\mathbf{x} = [\Delta \widetilde{t}_{i1}[0], \Delta \widetilde{t}_{i1}[1], \dots, \Delta \widetilde{t}_{i1}[N-1]]^T$ and parameter vector $\boldsymbol{\theta} = [\Delta t_{i1}, t_{drift}]^T$ is

$$p(\mathbf{x}; \boldsymbol{\theta}) = \frac{1}{(2\pi)^{N/2} |\mathbf{C}|^{1/2}} \times \exp\left\{-\frac{1}{2}[\mathbf{x} - \boldsymbol{\mu}(\boldsymbol{\theta})]^T \mathbf{C}^{-1}[\mathbf{x} - \boldsymbol{\mu}(\boldsymbol{\theta})]\right\} \quad (15)$$

where $\mathbf{C} = \sigma_i^2 \mathbf{I}$ is the covariance matrix of ϵ_i (with \mathbf{I} being the identity matrix), $|\cdot|$ denotes the determinant of a matrix, and

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & \bar{t}_i[0] - \bar{t}_1[0] \\ 1 & \bar{t}_i[1] - \bar{t}_1[1] \\ \vdots & \vdots \\ 1 & \bar{t}_i[N-1] - \bar{t}_1[N-1] \end{bmatrix} \begin{bmatrix} \Delta t_{i1} \\ t_{drift} \end{bmatrix} \\ = [\mathbf{U} \mid \mathbf{V}] \boldsymbol{\theta}$$

The first and second-order derivatives of $p(\mathbf{x}; \boldsymbol{\theta})$ are

$$\frac{\partial \ln p(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \mathbf{U}^T \mathbf{C}^{-1}[\mathbf{x} - \boldsymbol{\mu}(\boldsymbol{\theta})] \\ \mathbf{V}^T \mathbf{C}^{-1}[\mathbf{x} - \boldsymbol{\mu}(\boldsymbol{\theta})] \end{bmatrix} \\ = \frac{1}{\sigma_i^2} \begin{bmatrix} \sum_{n=0}^{N-1} (\Delta \widetilde{t}_{i1}[n] - \Delta t_{i1} - V[n]t_{drift}) \\ \sum_{n=0}^{N-1} (\Delta \widetilde{t}_{i1}[n] - \Delta t_{i1} - V[n]t_{drift}) V[n] \end{bmatrix} \quad (16)$$

and

$$\frac{\partial^2 \ln p(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2} = \begin{bmatrix} \mathbf{U}^T \mathbf{C}^{-1} \mathbf{U} & \mathbf{U}^T \mathbf{C}^{-1} \mathbf{V} \\ \mathbf{V}^T \mathbf{C}^{-1} \mathbf{U} & \mathbf{V}^T \mathbf{C}^{-1} \mathbf{V} \end{bmatrix} \\ = -\frac{1}{\sigma_i^2} \begin{bmatrix} N & \sum_{n=0}^{N-1} V[n] \\ \sum_{n=0}^{N-1} V[n] & \sum_{n=0}^{N-1} V[n]^2 \end{bmatrix} \quad (17)$$

Therefore, the Fisher information matrix is given by

$$\mathbf{I}(\boldsymbol{\theta}) = -E\left[\frac{\partial^2 \ln p(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] \\ = \frac{1}{\sigma_i^2} \begin{bmatrix} N & \sum_{n=0}^{N-1} V[n] \\ \sum_{n=0}^{N-1} V[n] & \sum_{n=0}^{N-1} V[n]^2 \end{bmatrix} \quad (18)$$

Now by solving the equation

$$\frac{\partial \ln p(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbf{I}(\boldsymbol{\theta})(\mathbf{g}(\mathbf{x}) - \boldsymbol{\theta})$$

we have

$$\hat{\boldsymbol{\theta}} = \mathbf{g}(\mathbf{x}) \\ = \begin{bmatrix} \frac{\sum V[n]^2 \times \sum \Delta \widetilde{t}_{i1}[n] - \sum V[n] \times \sum V[n] \Delta \widetilde{t}_{i1}[n]}{N \sum V[n]^2 - (\sum V[n])^2} \\ \frac{N \sum V[n] \Delta \widetilde{t}_{i1}[n] - \sum V[n] \times \sum \Delta \widetilde{t}_{i1}[n]}{N \sum V[n]^2 - (\sum V[n])^2} \end{bmatrix} \quad (19)$$

Although this estimator achieves CRLB, it does not produce accurate results because

$$\text{var}(\widehat{\Delta t_{i1}}) = [\mathbf{I}^{-1}(\boldsymbol{\theta})]_{11} \\ = \frac{\sigma_i^2 \sum_{n=0}^{N-1} V[n]^2}{N \sum_{n=0}^{N-1} V[n]^2 - (\sum_{n=0}^{N-1} V[n])^2} \quad (20)$$

which can be arbitrarily large. In the worst case $\text{var}(\widehat{\Delta t_{i1}}) \rightarrow \infty$ if $V[n] = \text{const}$.

C. CRLB of the MDToA-based Coordinate Estimations

The error vector derived from (6) is

$$\boldsymbol{\psi} = \mathbf{r} - \boldsymbol{\mu}(x_n, y_n) \quad (21) \\ \mathbf{r} = [r_{21}, r_{31}, \dots, r_{M-1,1}]^T \\ \boldsymbol{\mu}(x_n, y_n) = \begin{bmatrix} \|\mathbf{c}_n - \mathbf{c}_2\| - \|\mathbf{c}_n - \mathbf{c}_1\| \\ \|\mathbf{c}_n - \mathbf{c}_3\| - \|\mathbf{c}_n - \mathbf{c}_1\| \\ \vdots \\ \|\mathbf{c}_n - \mathbf{c}_{M-1}\| - \|\mathbf{c}_n - \mathbf{c}_1\| \end{bmatrix}$$

The likelihood function for the given observation vector $\mathbf{x} = \mathbf{r}$ and parameter vector $\boldsymbol{\theta} = [x_n, y_n]^T$ is the same as (15) but with $\mathbf{C} = \text{diag}\{\sigma_{2+}^2, \sigma_{3+}^2, \dots, \sigma_{M-1+}^2\}$ being the covariance matrix of $\boldsymbol{\psi}$ (where σ_{i+}^2 is determined by the estimation variance of $\widehat{\Delta t_{i1}}$ in Section II-D). We can simplify the case by considering $\sigma_{2+}^2 = \sigma_{3+}^2 = \dots = \sigma_{M-1+}^2 = \sigma^2$. As a result, the Fisher information matrix is computed as follows:

$$\mathbf{I}(\boldsymbol{\theta}) = -E\left[\frac{\partial^2 \ln p(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right] = \frac{1}{\sigma^2} \mathbf{D}^T(\boldsymbol{\theta}) \mathbf{D}(\boldsymbol{\theta}) \quad (22) \\ \mathbf{D}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{x_n - x_2}{\|\mathbf{c}_n - \mathbf{c}_2\|} - \frac{x_n - x_1}{\|\mathbf{c}_n - \mathbf{c}_1\|} & \frac{y_n - y_2}{\|\mathbf{c}_n - \mathbf{c}_2\|} - \frac{y_n - y_1}{\|\mathbf{c}_n - \mathbf{c}_1\|} \\ \frac{x_n - x_3}{\|\mathbf{c}_n - \mathbf{c}_3\|} - \frac{x_n - x_1}{\|\mathbf{c}_n - \mathbf{c}_1\|} & \frac{y_n - y_3}{\|\mathbf{c}_n - \mathbf{c}_3\|} - \frac{y_n - y_1}{\|\mathbf{c}_n - \mathbf{c}_1\|} \\ \vdots & \vdots \\ \frac{x_n - x_{M-1}}{\|\mathbf{c}_n - \mathbf{c}_{M-1}\|} - \frac{x_n - x_1}{\|\mathbf{c}_n - \mathbf{c}_1\|} & \frac{y_n - y_{M-1}}{\|\mathbf{c}_n - \mathbf{c}_{M-1}\|} - \frac{y_n - y_1}{\|\mathbf{c}_n - \mathbf{c}_1\|} \end{bmatrix}$$

and the CRLB is given by $\mathbf{C}_{\text{CRLB}} = \mathbf{I}^{-1}(\boldsymbol{\theta})$. When investigating the error characteristics with \mathbf{C}_{CRLB} , the *root mean square* (RMS) error is usually used as a scalar representative:

$$E_{\text{RMS}} = \sqrt{\sum_i \lambda_i} \quad (23)$$

where $\{\lambda_i\}$ are the eigenvalues of \mathbf{C}_{CRLB} .