Contemporary Engineering Sciences, Vol. 3, 2010, no. 7, 339 - 363

A Receiver/Transmitter Localization Algorithm on the Plane

Konstantinos Drakakis, Aishwarya Moni, Scott Rickard, and Ken Taylor[1]

UCD CASL
University College Dublin
Belfield, Dublin 4
Ireland

Abstract

We propose a receiver/transmitter localization algorithm, disassociating the problems of determining the optimal sensor topology and the optimal processing of the measurements. Accordingly, we test the algorithm under various planar topologies in order to determine the optimal one. We then compare its performance against some top performing previously published algorithms.

Mathematics Subject Classification: 62P30, 97M50

Keywords: source localization, Costas arrays

1 Introduction

Receiver localization refers to techniques used by a receiving device to determine its own position in space based on the transmissions it receives from several transmitters. Similarly, transmitter localization refers to the techniques used by a group of receiving devices to determine the position of a transmitter in space based on the transmissions they receive from it. Neither technique should be confused with RADAR/SONAR technologies, whereby a device that acts both as a receiver and a transmitter is used to gather information on (but

[1]The authors are also affiliated with the School of Electronic, Electrical & Mechanical Engineering, University College Dublin, Ireland. K. Drakakis and S. Rickard are additionally affiliated with the Claude Shannon Institute for Discrete Mathematics, Coding Theory, and Cryptography. **Email:** {Konstantinos.Drakakis, Scott.Rickard, Ken.Taylor}@ucd.ie, aishwaryamoni@gmail.com

not necessarily determine completely) the position of a passive object in space, namely one that is required neither to receive nor to transmit. Perhaps the most famous application of receiver localization today is GPS [42].

Localization problems can be naturally divided into two disjoint subproblems. To begin with, given an arrangement of transmitters/receivers in a noisy environment, what is the algorithm that optimizes localization? Subsequently, assuming such an algorithm has been determined (note that the actual optimal algorithm may depend on the nature of noise), what is the optimal location of transmitters/receivers in space?

In this work, we study and make original contributions to both subproblems in 2D. We first present a novel algorithm, dubbed Best Estimate out of a Sequence of Transmissions (BEST), which, given the formation of transmitters/receivers, suggests possible locations for the receiver/transmitter, and then we fine-tune its parameters to optimize its performance. We subsequently show that, for this algorithm, the circular arrangement of the transmitters/receivers outperforms several other arrangements, including random placement. Given that some (random) placements lead to very poor results indeed (such as the linear placement), this fact suggests a novel, simple, systematic, and extremely effective arrangement of transmitters/receivers on the plane, which confirms and extends previously published results [49–52].

Localization problems are conventionally classified into "near-field" and "far-field", according to whether both transmitter(s) and receiver(s) are collocated in approximately the same area, or the two groups lie far from each other, respectively. We present evidence below that confirms the superiority of the circular formation in both cases.

2 Description of the problem and the BEST algorithm

In this section we attempt a high-level description of the localization problem, as well as of the BEST algorithm we propose. We will be using $[n]$ as a notational shorthand for the set of the first n positive integers $\{1, 2, \ldots, n\}$.

2.1 Various formulations

In terms of their mathematical description, 2D localization can be modeled as two problems:

Problem 1. A set of points $P_i : \ (x_i, y_i)$, $i \in [n]$, $n \in \mathbb{N}^*$, is considered, whose positions are known. The "noisy" distances (from times of arrival)

$d_i + X_i = \sqrt{(x-x_i)^2 + (y-y_i)^2} + X_i$ from an unknown point $P: (x,y)$ are also known, where X_i, $i \in [n]$ are random variables representing noise. The goal is to determine the statistically optimal estimate $\hat{P}: (\hat{x},\hat{y})$ of P.

Assuming no noise is present, (x,y) is determined, of course, as the solution of the system

$$(x-x_i)^2 + (y-y_i)^2 = d_i^2,\ i \in [n], \tag{1}$$

which, in general, has a unique solution if $n \geq 3$, two solutions for $n = 2$, and infinitely many solutions for $n = 1$.

Problem 2. A set of points $P_i: (x_i, y_i)$, $i \in [n]$, $n \in \mathbb{N}^*$, is considered, whose positions are known. The "noisy" distance differences (from time delays of arrival) $d_{ij} + X_j - X_i = \sqrt{(x-x_j)^2 + (y-y_j)^2} - \sqrt{(x-x_i)^2 + (y-y_i)^2} + X_j - X_i$ from an unknown point $P: (x,y)$ are also known, where X_i, $i \in [n]$ are random variables representing noise. The goal is to determine the statistically optimal estimate $\hat{P}: (\hat{x},\hat{y})$ of P.

Assuming no noise is present, (x,y) is determined, of course, as the solution of the system

$$\sqrt{(x-x_i)^2 + (y-y_i)^2} - \sqrt{(x-x_j)^2 + (y-y_j)^2} = \\ = d_i - d_j,\ i > j,\ i,j \in [n], \tag{2}$$

which, in general, has a unique solution if $n \geq 4$, two solutions for $n = 3$, and infinitely many solutions for $n = 2$.

From now on, we will assume that n is large enough for the systems to be over-determined ($n > 2$ or 3, respectively). In terms of applications, these problems lead to two pairs of dual models, depending on the assignment of the roles of the receivers and the transmitters:

Model 1. A 2D wireless system consisting of n transmitters T_i, $i \in [n]$ and one receiver R is considered, satisfying the following conditions:

1. The location of the transmitters is fixed on the plane and known to the receiver.
2. The location of the receiver is fixed but unknown.
3. Each transmission indicates the transmission time and contains enough information for the receiver to know which transmitter it originates from (e.g. each transmitter uses a different frequency). The transmitters and the receiver have synchronized clocks.

The receiver's aim is to discover its own position.

Model 2. A 2D wireless system consisting of n receivers R_i, $i \in [n]$ and one transmitter T is considered, satisfying the following conditions:

1. The location of the receivers is fixed on the plane and known to the transmitter.

2. The location of the transmitter is fixed but unknown.

3. Each transmission indicates the transmission time. The transmitter and the receivers have synchronized clocks.

The receivers' aim is to determine the position of the transmitter.

Model 3. A 2D wireless system consisting of n transmitters T_i, $i \in [n]$ and one receiver R is considered, satisfying the following conditions:

1. The location of the transmitters is fixed on the plane and known.

2. The location of the receiver is fixed but unknown.

3. Each transmission contains enough information for the receiver to know which transmitter it originates from (e.g. each transmitter uses a different frequency).

The receiver's aim is to discover its own position.

Model 4. A 2D wireless system consisting of n receivers R_i, $i \in [n]$ and one transmitter T is considered, satisfying the following conditions:

1. The location of the receivers is fixed on the plane and known.

2. The location of the transmitter is fixed but unknown.

The receivers' aim is to determine the position of the transmitter.

Note that in Models 3 and 4 transmitter(s) and receiver(s) do not have synchronized clocks.

2.2 Noise effects in more detail

Assume now that, instead of knowing the distances (from times of arrival) d_i, we only know noisy versions thereof, namely $d_i + X_i$, where X_i, $i \in [n]$ is a collection of independent random variables (and possibly identically distributed, too). The two over-determined systems of the previous section now become

$$(x - x_i)^2 + (y - y_i)^2 = (d_i + X_i)^2,\ i \in [n] \tag{3}$$

and

$$\sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{(x - x_j)^2 + (y - y_j)^2} = \\ = d_i - d_j + X_i - X_j,\ i > j,\ i, j \in [n]. \tag{4}$$

It is clear that almost certainly neither system will admit a solution any longer, and we are faced with the task of choosing an approximate solution. We propose the following method to achieve this.

As mentioned earlier, for sufficiently large n, both systems (1) and (2) are over-determined (often severely so), and, in fact, in both cases two equations are enough to determine two possible solutions (x, y). In the context of system (3), this implies that every pair of known points will produce a pair of solutions for the unknown point; similarly, in the context of system (4), every triplet of known points will produce a pair of solutions for the unknown point. Given the fact that n points are available, we get the opportunity to determine $\binom{n}{2}$ pairs of solutions for system (1) and $\binom{n}{3}$ pairs of solutions for system (2), amounting to $n(n-1)$ or $n(n-1)(n-2)/3$ solutions (x, y), respectively. In each case, and in the absence of noise, half of the points coincide on the plane, thus determining the correct solution (x, y): we only need to look for the most frequent point in the set.

When noise is present, however, the previous methodology will no longer lead to the coincidence of half of the solutions on the plane. Instead, we can only hope that, for moderately noisy conditions, some of the solutions will form a cluster, whose mean value (in some sense), or some other representative, will lead to a good approximation of the correct (x, y).

2.3 Some previously published work

It is very difficult to solve (1), because it is non-linear, so (5) is usually solved instead:

$$(x - x_1)^2 + (y - y_1)^2 \quad = \quad d_1^2 \tag{5}$$

$$(x-x_i)^2+(y-y_i)^2 \;=\; (d_{i1}+d_1)^2, i>1,\ i\in[n],$$

where $d_{i1}=d_i-d_1$. A few closed form solutions have been proposed over the years including [35] which involves intersecting spheres from (5), and [38] which involves minimizing the error between the two sides of (5). Furthermore, [36] uses the fact that three sensors result to an equation of a plane, and $\binom{n}{3}$ such planes are intersected, while [29] uses the result from [38] and adds a linear correction step using Lagrange multipliers; [12] also uses the result from [38] and adds a correction step, which, however, requires knowledge of the covariance matrix of errors in the range differences.

A good overview of some of the techniques mentioned here is provided in [41], where a new correction step is introduced. Other algorithms for solving the source localization and direction of arrival problem given time delay estimates using closed form methods are described in [5–7, 10, 13, 20, 24, 27, 34, 40, 44, 46–48]. The localization problem can be extended to sensor networks where a few nodes or sensors know their position within the network's coordinate system and all other nodes calculate their position in the local coordinate system based on information received from neighboring nodes. Some of the methods used to solve this problem are described in [1–3, 8, 9, 21, 25, 30].

2.4 Solving the systems

Let us begin with two equations of the system (1), corresponding to indices i and j; this also leads to the solution of system (3), substituting the noisy Times of Arrival in the formula instead. The resulting 2×2 nonlinear system is:

$$\begin{aligned}(x-x_i)^2+(y-y_i)^2 &= d_i^2\\ (x-x_j)^2+(y-y_j)^2 &= d_j^2.\end{aligned} \tag{6}$$

Similarly, let us consider three equations of the systems (2) and (4), corresponding to indices i, j, and k. The resulting 2×2 nonlinear system this time is:

$$\begin{aligned}\sqrt{(x-x_i)^2+(y-y_i)^2} &= \sqrt{(x-x_j)^2+(y-y_j)^2}+d_i-d_j\\ \sqrt{(x-x_i)^2+(y-y_i)^2} &= \sqrt{(x-x_k)^2+(y-y_k)^2}+d_i-d_k.\end{aligned} \tag{7}$$

Further straightforward, yet tedious, manipulations yield the solutions to these systems. For system (6), and assuming $y_i\neq y_j$, the solution is expressed as

$$(x, u+vx), \tag{8}$$

where x is a root of the polynomial

$$(1+v^2)x^2 + 2(uv - vy_i - x_i)x + (u^2 - 2uy_i + a_i^2 - d_i^2) = 0, \tag{9}$$

and where

$$u = \frac{a_j^2 - a_i^2 - (d_j^2 - d_i^2)}{2(y_j - y_i)}, \;\; v = -\frac{x_j - x_i}{y_j - y_i}, \;\; a_i^2 = x_i^2 + y_i^2; \tag{10}$$

whenever $y_i = y_j$, it must necessarily hold that $x_i \neq x_j$, and it follows that

$$x = \frac{x_i + x_j}{2} - \frac{d_j^2 - d_i^2}{2(x_j - x_i)}, y = y_1 \pm \sqrt{d_1^2 - (x - x_1)^2}. \tag{11}$$

In the absence of noise, this quadratic equation has two real roots; furthermore, the resulting equations for all pairs (i, j) all have a root in common. When noise is present, it is possible for this equation to have no real roots, while the event of the equation having a double root clearly has probability zero and may be ignored.

For system (7), the solution can be written in the form

$$x = u_x + v_x z, \;\; y = u_y + v_y z \tag{12}$$

where z is a root of the polynomial

$$(v_x^2 + v_y^2 - 1)z^2 + 2((u_x - x_i)v_x + (u_y - y_i)v_y)z + [(u_x - x_i)^2 + (u_y - y_i)^2] = 0, \tag{13}$$

and where

$$d = (x_i - x_j)(y_i - y_k) - (x_i - x_k)(y_i - y_j) \tag{14}$$
$$2du_x = (y_i - y_k)(d_{ij}^2 + a_i^2 - a_j^2) + (y_j - y_i)(d_{ik}^2 + a_i^2 - a_k^2) \tag{15}$$
$$2du_y = (x_k - x_i)(d_{ij}^2 + a_i^2 - a_j^2) + (x_i - x_j)(d_{ik}^2 + a_i^2 - a_k^2) \tag{16}$$
$$v_x = \frac{(y_i - y_k)d_{ij} + (y_j - y_i)d_{ik}}{d} \tag{17}$$
$$v_y = \frac{(x_k - x_i)d_{ij} + (x_i - x_j)d_{ik}}{d} \tag{18}$$
$$d_{ij} = d_j - d_i \tag{19}$$
$$a_i^2 = x_i^2 + y_i^2. \tag{20}$$

In the absence of noise, this quadratic equation has again two real roots; furthermore, the resulting equations for all triplets (i, j, k) all have a root in common. When noise is present, it is similarly possible for this equation to have no real roots, while double roots have probability zero as before and may be ignored.

2.5 Post-processing under noise

There remains the question of how to use the information available through the $n(n-1)$ or $n(n-1)(n-2)/3$ solution points obtained in order to "denoise" and produce the best possible approximation to the actual solution. There are various alternative strategies to achieve this, for example:

1. Use the mean or median value (coordinate-wise) of all solution points found.

2. Set $e_i(x,y) = |(x-x_i)^2 + (y-y_i)^2 - d_i^2|$ or $e_{ij}(x) =$

$$= |\sqrt{(x-x_i)^2+(y-y_i)^2} - \sqrt{(x-x_j)^2+(y-y_j)^2} - (d_i - d_j)| \quad (21)$$

   for system (1) or (2), respectively, where d_i are taken to be the observed noisy values. Set now

$$w(x,y) = \sum_{i=1}^{n} e_i(x,y) \text{ or } w(x,y) = \sum_{\substack{i,j=1 \\ i>j}}^{n} e_{ij}(x,y), \quad (22)$$

   respectively, and choose that (x,y) amongst the solution points that minimizes w, or, alternatively, choose

$$(x,y) = \frac{\sum_{i=1}^{N} \exp(-w(x_i,y_i))(x_i,y_i)}{\sum_{i=1}^{N} \exp(-w(x_i,y_i))}. \quad (23)$$

2.6 Multiple transmissions

Further noise reduction can be achieved if it is possible to observe multiple transmissions. Assuming m transmissions are observed in total, and that noise exhibits no dependence across transmissions, one of the suggested strategies of Section 2.5 can be applied for each transmission, resulting in m candidates which can then be further combined using one of the strategies (possibly a different one than before) to yield a better estimate of P.

This combination of temporal information from multiple transmissions in order to produce a better localization estimate is one of the strengths of the algorithm we propose, though it can obviously be added as a further processing stage to any localization algorithm.

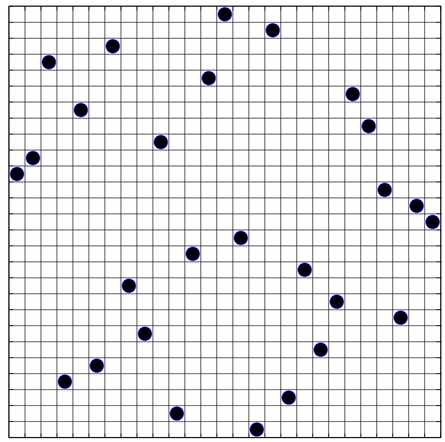

Figure 1: An example of a 27×27 Costas array

3 Costas arrays

As in Section 4 below we will be considering Costas arrays topologies, we believe it is in order to provide here the definition and a few facts about them. A Costas array [15, 16, 18] is a permutation array, namely an array whose elements are dots/1s and blanks/0s so that there is exactly one dot per row and column, with the extra property that a) no four dots form a parallelogram and b) no three dots lying on the same straight line are equidistant. More precisely, assuming $A = [a_{ij}]$ is an $n \times n$ array, there exists a permutation $f : [n] \to [n]$ such that $a_{ij} = 1$ if $i = f(j)$ and 0 otherwise, and such that the vectors $\{(f(i) - f(j), i - j) : 1 \leq j < i \leq n\}$ are all distinct. This, in turn, is equivalent to saying that for all $i, j, k \in [n]$ such that $i + k, j + k \in [n]$ as well, the equality $f(i + k) - f(i) = f(j + k) - f(j)$ necessarily implies $i = j$. It is precisely this maximal diversity of distance vectors that motivated us to consider using Costas arrays in localization. An example of a 27×27 Costas array is given in Figure 1.

Costas arrays were originally used in SONAR (and later in RADAR) systems, but many fascinating mathematical problems still remain open about them, as, for example, their existence for all n. Currently all Costas arrays for $n \leq 27$ are known [19], whereas available algebraic constructions can be used to obtain Costas arrays for infinitely many (but not every) n [18, 22, 23].

4 Results

Our goal in this section will be to provide evidence that the arrangement of receivers/transmitters on a circular pattern leads to excellent localization results, but also further details on and insight into the operation of the algorithm. In order to keep the discussion as brief and to the point as possible, we will only present a few representative results.

4.1 Setup of the experiments

We consider various topologies of receiver/transmitter placement. In order to perform a fair comparison, we keep the value of n constant throughout all experiments: we chose $n = 15$, but our conclusions hold for other values of n. We consider both discrete placement and in the continuum: in the latter, points can be placed anywhere on the plane; in the former, points can be placed only on the integer lattice of the plane. Although it could be argued that in the continuum case it might make more sense to normalize by placing all 15 points within the unit circle (or the unit square), doing so would make a direct comparison between continuum and discrete results rather difficult. For this reason, we used approximately equivalent placements in the two cases.

The topologies we consider are:

- Linear: the 15 points are equispaced on the y-axis of the plane between -7 and 7, in both the continuum and the discrete case.
- Cyclic: the 15 points lie equispaced on the circumference of the circle of radius 7.5 centered at the origin; for the discrete case, the locations are rounded to the nearest lattice point.
- Cyclic with center: the 14 points lie equispaced on the circumference of the circle of radius 7.5 centered at the origin, while the 15th lies at the origin; for the discrete case, the locations are rounded to the nearest lattice point.
- Random: the 15 points are positioned randomly within the square of side 15 with sides parallel to the axes and centered at the origin; for the discrete case, a random permutation array of order 15 is used, centered at the origin.
- Costas array: the 15 points are positioned on the dots of a Costas array of order 15, centered at the origin.

A comment about the random placement is in order. In order for the comparison of the different placements to be fair, we had to ensure that the points for all placements lie in approximately the same area. More precisely, we made sure that the minimal square enclosing all the points is (approximately) the one of side 15 with sides parallel to the axes and centered at the origin. Randomly choosing 15 points within this square is, however, not enough to guarantee this condition, and this necessitated "sieving" our random choices appropriately.

Once the topology is set, we attempt to localize 100 points equispaced on the circumference of the circle of radius $r = 10$, 100, and 1000, centered at the origin. We note that, when $r = 10$, rounding does not lead to 100 distinct points on the integer lattice, as some points occur multiple times. The random variables $X_i,\ i \in [15]$ are taken to be i.i.d., following a uniform distribution within $(-\epsilon, \epsilon)$, where $\epsilon = 0.1$, 1, and 10. Overall, then, we perform nine experiments on each topology. As the number of experiments grows too large, though, we refrained from performing all experiments for all topologies under all scenarios, as long as we were confident that enough evidence had already been given for the behavior of a certain topology to be understood.

But how is the initial best estimate selected, to begin with? In Section 2.5, we proposed two alternatives. Our tests suggested, however, that, in all cases, selecting the point with the minimal weight (22) yields the best result, as opposed to selecting the weighted sum of all points given by (23). Furthermore, in accordance with Section 2.6, we consider $m = 10$ multiple transmissions, and reapply (22) to the set of best estimates from each transmission in order to determine the overall best estimate. In order to directly compare BEST to other algorithms in the literature, we carried out an additional experiment for a single transmission.

The main results of our experiments are presented in Tables 1, 2, 3, and 4. Each table entry corresponds to the double mean value of the localization error over all positions on the circle of the point to be localized and over five runs of the same experiment.

4.2 Known Times of Arrival in the continuum

Our first set of experiments assumes that times of arrival are known and that points are allowed to be placed in the continuum. We compared four topologies: Costas, random, circular, and linear. The results are shown in Table 1, where the following conclusions can immediately be drawn from:

- Costas and random topologies behave almost identically. Perhaps a case can be made that in the very far field ($r = 1000$) the Costas topology has a slight edge.

- The circular topology is the overall best performer, outperforming Costas and random topologies by 15%-25%.

	$r = 10$	$r = 100$	$r = 1000$
$\epsilon = 0.1$	0.0060	0.0510	0.4838
$\epsilon = 1.$	0.0592	0.5049	4.8611
$\epsilon = 10.$	0.5402	5.1345	48.6262

	$r = 10$	$r = 100$	$r = 1000$
$\epsilon = 0.1$	0.0061	0.0562	0.4993
$\epsilon = 1.$	0.0587	0.5180	5.3161
$\epsilon = 10.$	0.5513	5.0893	53.5340

	$r = 10$	$r = 100$	$r = 1000$
$\epsilon = 0.1$	0.0049	0.0398	0.4196
$\epsilon = 1.$	0.0487	0.4090	4.1781
$\epsilon = 10.$	0.4131	4.3762	41.5098

	$r = 10$	$r = 100$	$r = 1000$
$\epsilon = 0.1$	6.6105	66.2533	628.8932
$\epsilon = 1.$	6.8625	65.5471	640.7802
$\epsilon = 10.$	6.6483	69.7801	682.6302

Table 1: The localization error for various topologies, assuming continuum placement and known times of arrival: Costas (top left), random (top right), circular (bottom left), and linear (bottom right).

- The linear topology performs extremely poorly. In fact, it appears that the error is insensitive to the noise level, and is comparable to the radius of the circle the point to be localized lies on. Intuitively, we expect this to be the case because the linear topology is completely symmetric and unable to distinguish between "front" and "back": in other words, assuming P lies at (x, y), half of the time the algorithm will return an estimate close to $(-x, y)$ instead. Further inspection of the results indeed confirms this. This naturally implies further that the poor performance of this topology will be a recurring observation in all subsequent experiments.

4.3 Known Times of Arrival on the integer lattice

This set of experiments assumes that times of arrival are known and that points are allowed to be placed on the integer lattice of the plane only. We compared essentially three topologies: Costas (in two versions, with and without grid search), circular, and linear. The results are shown in Table 2, where the following conclusions can immediately be drawn from:

- The linear topology behaves very poorly once more, due to the same reasons as before.

- The effect of grid search on the Costas topology is significant: the method now performs flawlessly for $r = 10$ and $r = 100$, and yields substantially improved results for $r = 1000$, for all noise levels.

- The circular topology yields once more the best results (notwithstanding the grid search improvement, of course).

$\epsilon \backslash r$	10	100	1000
0.1	0	0	0.2778
1.	0	0.1643	9.2789
10.	0.3814	5.3727	50.9016

$\epsilon \backslash r$	10	100	1000
0.1	0	0	0.0473
1.	0	0	7.4227
10.	0	0	37.5299

$\epsilon \backslash r$	10	100	1000
0.1	0	0	0.2043
1.	0	0.1308	7.7718
10.	0.2353	4.0161	45.2438

$\epsilon \backslash r$	10	100	1000
0.1	6.2640	58.2800	623.0136
1.	6.6340	61.9405	648.6946
10.	7.1240	69.1000	743.1117

Table 2: The localization error for various topologies, assuming lattice placement and known times of arrival: Costas (top left), Costas with additional grid search (top right), circular (bottom left), and linear (bottom right).

We do not show results for random topologies, as they are virtually identical to the ones for Costas topologies. Furthermore, we do not show the results of grid search applied on the circular topology, as they show an improvement analogous to the Costas topology. It is important, however, to compare Tables 1 and 2: it is clear that the errors on the discrete lattice are smaller than in the continuum, at least for small noise levels. This we may attribute to the "snap to grid" effect: as the output of the algorithm gets rounded to the closest integer coordinates, small noise levels are not enough to shake it away from its true value.

4.4 Known Time Differences of Arrival in the continuum

This set of experiments assumes continuum placement and that time differences of arrival are known, instead of times of arrival. We compared four topologies: Costas, random, circular, and circular with center. The results are shown in Table 3, where the following conclusions can immediately be drawn from:

- Noise levels corresponding to $\epsilon = 1$ and 10 for $r = 1000$ clearly lie beyond the capabilities of the algorithm, as the resulting localization error exceeds 100% of the true value.

- Costas and random topologies are essentially equivalent, with one notable exception: the Costas topology seems to lead to significantly smaller error for $r = 100$ and $\epsilon = 10$. This difference appears consistently and is too large to be attributed to random fluctuations.

$\epsilon\backslash r$	10	100	1000
0.1	0.0092	0.9857	96.9312
1.	0.0927	9.9992	2468.2627
10.	0.3169	132.2623	1021.5333

$\epsilon\backslash r$	10	100	1000
0.1	0.0083	0.9986	92.7296
1.	0.0839	10.1565	2389.6868
10.	0.3153	195.5992	1856.0710

$\epsilon\backslash r$	10	100	1000
0.1	0.0065	0.7451	72.5603
1.	0.0667	7.9183	1515.2364
10.	0.2320	148.2784	1532.3317

$\epsilon\backslash r$	10	100	$r1000$
0.1	0.0070	0.7264	76.2603
1.	0.0613	7.8091	2027.2827
10.	0.2478	254.9344	1033.606

Table 3: The localization error for various topologies, assuming continuum placement and known time delays of arrival: Costas (top left), random (top right), circular (bottom left), and circular with middle dot (bottom right).

- The circular topology leads once more to the smallest errors (though, in this particular sample, the Costas topology outperforms it for $r = 100$ and $\epsilon = 10$): in general, it outperforms Costas and random topologies by 20%-30%.

- The performance of the circular topology with center is slightly below that of the circular. Further analysis of the results, however, indicates that it is prone to producing outliers, namely estimates that lie extremely far from the true points. A more careful implementation of the method helps alleviate this effect, but not eliminate it: it is still clearly visible in the value corresponding to $r = 100$ and $\epsilon = 10$.

Comparing Tables 1 and 3, we see that the error levels now rise much faster as the noise levels rise. Intuitively, this is to be expected, as the difference of two uniformly random variables within $[-\epsilon, \epsilon]$ is a variable with triangular distribution in $[-2\epsilon, 2\epsilon]$, which carries higher variance (namely energy) than the former.

4.5 Known time delays of arrival on the integer lattice

This set of experiments assumes continuum placement and that time differences of arrival are known, instead of times of arrival. We compared essentially three topologies: Costas, random (with and without grid search), and circular. The results are shown in Table 4, where the following conclusions can immediately be drawn from:

- Noise levels corresponding to $\epsilon = 1$ and 10 for $r = 1000$ clearly lie beyond the capabilities of the algorithm, as the resulting localization

$\epsilon\backslash r$	10	100	1000
0.1	0	0.8417	97.9958
1.	0	9.6985	1507.2193
10.	0.0630	145.1440	1180.5988

$\epsilon\backslash r$	10	100	1000
0.1	0	1.0018	89.3668
1.	0	10.6216	1488.902
10.	0.0653	149.9236	1166.8957

$\epsilon\backslash r$	10	100	1000
0.1	0	0.6224	87.5055
1.	0	7.6713	1648.2829
10.	0.0020	150.1818	984.2374

$\epsilon\backslash r$	10	100	1000
0.1	0	0.2649	92.6167
1.	0	8.1402	1844.8227
10.	0	131.2996	972.7405

Table 4: The localization error for various topologies, assuming lattice placement and known times of arrival: Costas (top left), random (top right), circular (bottom left), and random with additional grid search (bottom right).

error exceeds 100% of the true value.

- Costas and random topologies are essentially equivalent.
- The circular topology leads once more to the smallest errors (excluding grid search, of course), though it does not outperform the other topologies by the same high percentages as in Section 4.4.

Comparing with Section 4.3, we see that the effect of the grid search diminishes significantly as r and ϵ increase. On the other hand, comparing with Section 4.4, we see that the "snap to grid" effect, described in Section 4.3, persists for small r ($r = 10$).

4.6 Comparison with other algorithms

As might be expected, multiple transmissions give a significant edge to BEST over existing algorithms published in the literature. But how does it fare under single transmission? We carried out such an experiment in the same context as Section 4.2, namely assuming known times of arrival and continuum placement, and using circular formation, comparing BEST to the OSLS algorithm, as described in [28] and a Maximum Likelihood grid search algorithm [33].

Regarding OSLS, the results, shown in Table 5, demonstrate that BEST is indeed better, even using single transmission. We also compared the algorithms using random formations: the use of random versus circular formation shows the same improvement. Regarding ML, the errors at the lower levels of noise returned by the grid search algorithm is quantisation noise. Clearly, BEST cannot compete with an exhaustive search grid algorithm; though, however,

ϵ	r	Errors
0.1	10	0.0176/0.0435/0.0665/0.4887/0.3986
0.1	100	0.1572/0.3341/0.4936/0.7948/0.7097
0.1	1000	1.6338/3.4441/4.7834/1.4067/1.1071
1.	10	0.1924/0.4536/0.6195/0.9872/0.7232
1.	100	1.5236/3.4655/4.9969/1.6409/1.22
1.	1000	16.0842/37.0392/49.0416/1.5502/1.2505
10.	10	1.6086/4.5860/6.8418/1.9682/2.084
10.	100	15.2902/35.4447/48.1418/1.9286/1.4675
10.	100	154.0628/311.8042/528.7115/1.9097/1.5047

Table 5: A single transmission localization error comparison between BEST using circular formation (left) and the algorithm described in [28] using circular formation (second from left) and random formation (middle) and the corresponding errors for the ML grid search algorithm [33] using circular formation (second from right) and random formation (right): known times of arrival and continuum placement was assumed, just as in Section 4.2.

grid algorithms are more robust in higher noise levels, they incur much higher computational complexity.

4.7 Argument estimates

A striking and surprising feature of BEST is that the localization error, even under high noise, is mostly radial: the argument of the point on the circle to be localized is captured with perhaps unexpectedly high accuracy, almost without error, even for the highest noise levels we tested. This could potentially make BEST particularly suitable for applications where angle determination is the main purpose of localization. An example of such an application, consider a round table with a teleconferencing link in the middle, including a rotating camera equipped with a microphone: the camera only needs to identify correctly the angle at which the person currently speaking is seated, as the distance can be reasonably assumed to be known and fixed (just beside the table). In any case, standard camera focusing technology is based on image processing and does not involve sound. Similar problems are, in fact, considered in [28, 47], but the solutions proposed are not as robust as the method described here.

Figure 2 shows the argument estimate of the localization algorithm as the point to be localized moves around a circle of radius $r = 1000$, assuming circular topology and continuum placement, and setting $\epsilon = 10$, when either times of arrival or time delays of arrival are known. In both cases, argument recovery is almost exact, despite the otherwise high localization errors.

How do other algorithms compare to best? For comparison purposes, the same figure also shows the corresponding estimate by the OSLS algorithm described in [28], which is clearly inferior, as well as the ML grid algorithm described in [33], which is practically exact! Recall, though, that this performance comes at a much higher computational complexity.

4.8 Argument-dependent error

The specific experiment setup we have been following, according to which the point P to be localized moves on a circle around the topology, allows us to trace the "uncertainty profile" of the topology used: in other words, we can graph the localization error as a function of the argument θ of P. We show these profiles for the circular and the Costas topology in the continuum in Figure 3, for both cases when times of arrival or time delays of arrival are known; we used $r = 100$ and $\epsilon = 1$. Regarding the linear topology, in particular, about which we noted in Section 4.2 that it performs extremely poorly, we can now additionally confirm that the localization error is strongly directional: it is small along the line on which the points are arranged, and large perpendicularly to it. This fact is also suggested by the observation we made in Section 4.2 that the linear topology gets confused between $P:\ (x, y)$ and $P':\ (-x, y)$.

5 Conclusion and future work

We have presented a localization algorithm, dubbed Best Estimate out of a Sequence of Transmissions (BEST), which we tested with many different sensor topologies (including Costas topologies), and compared to pre-existing algorithms (such as OSLS and ML). Our conclusions were the following:

- BEST performs better than OSLS, one of the best performing (non exhaustive-search) published algorithms, even assuming single transmission, while multiple transmissions give it a further boost. While the ML grid search algorithm shows a much better performance, it only does so at a much higher cost in computational complexity.

- The best topology in all cases to be used with BEST is the circular. The performance of the circular topology with center is close to the circular but prone to outliers, while Costas and random topologies behave is usually indistinguishable, though perhaps the Costas topology has a slight edge. The linear topology is best avoided.

- Assuming lattice placement, an additional grid search step leads to substantial improvement of the localization. Comparing lattice to continuum placement, the “snap to grid” effect in the former leads, as a rule, to smaller errors.
- Assuming time delays of arrival are known instead of the times of arrival themselves, errors grow much faster and much higher as the noise level increases. For $r = 1000$, in particular, localization is reliable only for $\epsilon = 0.1$; the algorithm clearly fails for $\epsilon = 1$ and 10.

The corresponding study in 3D is perhaps more relevant for applications, and will be considered in future work.

Acknowledgements

The authors would like to thank Dr. Paul O’Grady (UCD CASL) for his comments on the manuscript. This material is based upon works supported by the Science Foundation Ireland under Grant No. 05/YI2/I677, 06/MI/006 (Claude Shannon Institute), and 08/RFP/MTH1164.

References

[1] H. Akcan, V. Kriakov, H. Bronnimann and A. Delis. “GPS-Free Node Localization in Mobile Wireless Sensor Networks.” *Proceedings of the 5th ACM MobiDE,* Chicago, 2006.

[2] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. “A theory of network localization.” *IEEE Transactions on Mobile Computing,* 5(12): pp. 1663–1678, 2006.

[3] J. Aspnes, D. K. Goldenberg, and Y. R. Yang. “On the com- putational complexity of sensor network localization.” *First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS04),* pp. 32–44, Turku, Finland, July 2004.

[4] A. Beck, P. Stoica and J. Li. “Exact and approximate solutions of source localization problems.” *IEEE Transactions on Signal Processing,* Volume 56, Issue 5, pp./ 1770–1778, May 2008.

[5] J. Benesty. “Adaptive eigenvalue decomposition algorithm for passive acoustic source localization.” *Journal of the Acoustic Society of America,* Volume 107, Issue 1, pp. 384–391, Mar 2000.

[6] M.S. Brandstein, J.E. Adcock and H.F. Silverman. "A closed-form method for finding source locations from microphone-array time-decay estimates." *1995 International Conference on Acoustics, Speech, and Signal Processing,* Volume 5, pp. 3019–3022, May 1995.

[7] M.S. Brandstein, J.E. Adcock and H.F. Silverman. "A closed-form location estimator for use with room environment microphone arrays." *IEEE Transactions on Speech and Audio Processing,* Volume 5, Issue 1, pp. 45–50, Jan 1997 .

[8] N. Bulusu, J. Heidemann, and D. Estrin. "Gps-less low cost outdoor localization for very small devices." *IEEE Personal Communications Magazine,* 7(5): pp. 28–34, Oct 2000.

[9] S. Capkun, M. Hamdi, and J.P. Hubaux. "GPS-free Positioning in Mobile Ad Hoc Networks." *Cluster Computing,* 5(2): pp. 157–167, 2002.

[10] G.C. Carter. "Coherence and Time Delay Estimation." *Proceedings of the IEEE,* Volume 75, Issue 2, pp. 236–255, Feb 1987.

[11] J. Chaffee and J. Abel. "GDOP and Cramer-Rao Bound." *IEEE Position Location and Navigation Symposium,* pp. 663–668, Apr 1994.

[12] Y.T. Chan and K.C. Ho. "A Simple and Efficient Estimator for Hyperbolic Location." *IEEE Transactions on signal processing,* Volume 42, Issue 8, pp. 1905–1915, Aug 1994.

[13] J.C. Chen, Y. Kung and R.E. Hudson. "Source localization and beamforming." *IEEE Signal Processing Magazine,* Volume 19, Issue 2, pp. 30–39, Mar 2002.

[14] K.W. Cheung, H.C. So, W.K. Ma and Y.T Chat. "Least squares algorithms for time-of-arrival-based mobile location ." *IEEE Transactions on Signal Processing,* Volume 52, Issue 4, pp. 1121–1128, Apr 2004.

[15] J.P. Costas. "Medium constraints on sonar design and performance." Technical Report Class 1 Rep. R65EMH33, GE Co., 1965.

[16] J.P. Costas. "A study of detection waveforms having nearly ideal range-doppler ambiguity properties." *Proceedings of the IEEE,* Volume 72, Issue 8, pp. 996–1009, Aug 1984.

[17] P. Duff and H. Muller. "Autocalibration Algorithm for Ultrasonic Location Systems." *IEEE International Symposium on Wearable Computers,* pp. 62–68, Nov 2003.

[18] K. Drakakis. “A review of Costas arrays.” Journal of Applied Mathematics, Volume 2006.

[19] K. Drakakis, S. Rickard, J. Beard, R. Caballero, F. Iorio, G. O’Brien, J. Walsh. “Results of the enumeration of Costas arrays of order 27.” *IEEE Transactions on Information Theory,* Volume 54, Issue 10, pp. 4684–4687, Oct 2008.

[20] B. Friedlander. “A Passive Localization Algorithm and Its Accurancy Analysis.” *IEEE Journal of Oceanic Engineering,* Volume 12, Issue 1, pp. 234–245, Jan 1987.

[21] T. Eren, O. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Ander- son, and P. Belhumeur. “Rigidity, computation, and randomization in network localization.” *INFOCOM,* Volume 4, pp. 2673–2684, Hong Kong, China, March 2004.

[22] S. Golomb. “Algebraic Constructions For Costas Arrays.” *Journal Of Combinatorial Theory Series A,* Volume 37, Issue 1, pp. 13–21, 1984.

[23] S. Golomb and H. Taylor. “Constructions and properties of Costas arrays.” *Proceedings of the IEEE,* Volume 72, pp. 1143–1163, Sep 1984.

[24] S.M. Griebel and M.S. Brandstein. “Microphone array source localization using realizable delay vectors.” *2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics,* pp. 71–74.

[25] F. Gustafsson and F. Gunnarsson. “Mobile positioning using wireless networks.” *IEEE Signal Processing Magazine,* pp. 41–53, 2005.

[26] Ho and Vicente. “Sensor allocation for Source Localization With Decoupled Range and Bearing Estimation.” *IEEE Transactions on Signal Processing,* Volume 56, pp. 5773–5789, Dec 2008.

[27] J.Y.L.I. Hietanen, K. Koppinen and J. Astola. “Time-delay Selection For Robust Angle of Arrival Estimation.” *SIP99: Proceedings of the IASTED International Conference Signal and Image Processing,* pp. 81–83, Oct 1999.

[28] Y. Huang, J. Benesty, and G.W. Elko. “Passive acoustic source localization for video camera steering.” *ICASSP ’00: Proceedings of the IEEE International Conference on the Acoustics, Speech, and Signal Processing,* pp. II909–II912, 2000.

[29] Y. Huang, J. Benesty, G.W. Elko and R.M. Mersereau. “Real-Time Passive Source Localization: A Practical Linear-Correction Least-Squares Approach.” *IEEE Transactions on Speech and Audio Processing,* Volume 9, Issue 8, pp. 943–956, Nov 2001.

[30] R. Iyengar and B. Sikdar. “Scalable and distributed GPS free posi- tioning for sensor networks.” *Proceedings of the IEEE International Conference on Communications (ICC03),* pp. 338–342, Anchorage, AK, USA, May 2003.

[31] D. Li and Y.H. Hu. “Least Square Solutions of Energy Based Acoustic Source Localization Problems.” *IEEE Proceedings of the 2004 International Conference on Parallel Processing Workshops,* Aug 2004.

[32] K.W.K. Lui and H.C. So. “A study of two-dimensional sensor placement using time-difference-of-arrival measurements.” *Digital Signal Processing,* Volume 19, pp. 650–659, 2009.

[33] A. Moni and S. Rickard. “Comparison of Localization Algorithms Using Attenuation Estimates.” *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop,* pp. 42–47, Jan 2009.

[34] Y. Rui and D. Florencio. “New Direct Approaches to Robust Sound Source Localisation.” *Proceedings of the 2003 International Conference on Multimedia and Expo,* Volume 2, pp. 737–740.

[35] H.C. Schau and A.Z. Robinson. “Passive Source Localization Employing Intersecting Spherical Surfaces from Time-of-Arrival Differences.” *IEEE Transactions on Acoustical Speech and Signal Processing,* Volume 35, Issue 8, pp. 1223–1225, Aug 1987.

[36] R.O. Schmidt. “A New Approach to Geometry of Range Difference Location.” *IEEE Transactions on Aerospace and Electronic Systems,* Volume 8, Issue 6, pp. 821–835, Nov 1972.

[37] R.O. Schmidt. “Multiple emitter location and signal parameter estimation.” *IEEE Transactions on Antennas and Propagation,* Volume 34, Issue 3, pp. 276–280, Mar 1986.

[38] J.O. Smith and J.S. Abel. “The Spherical Interpolation Method of Source Localization.” *IEEE Journal of Oceanic Engineering,* Volume 12, Issue 1, pp. 246–252, Jan 1987.

[39] J.O. Smith and J.S. Abel. "Closed-Form Least-Squares Source Location Estimation from Range-Difference Measurements." *IEEE Transactions on Acoustics, Speech and Signal Processing,* Volume 35, Number 12, pp. 1661–1669, Dec 1987.

[40] P. Stoica and K.C. Sharman. "Maximum likelihood methods for direction-of-arrival estimation." *IEEE Transactions on Acoustics, Speech and Signal Processing,* Volume 38, Issue 7, pp. 1132–1143, Jul 1990.

[41] P. Stoica and J. Li. "Source Localization from Range-Difference Measurements." *IEEE Signal Processing Magazine,* Volume 23, Issue 6, pp. 63–69, Nov 2006.

[42] J.B-Y. Tsui. "Fundamentals of Global Positioning System Receivers - A Software Approach." John Wiley and Sons Inc, 2000.

[43] I. Sharp, Y. Kegen and Y.J. Guo. "GDOP Analysis for Positioning System Design." *IEEE Transactions on Vehicular Technology*, Volume 58, Issue 7, pp. 3371–3382, Sep 2009.

[44] T-L. Tung, K. Yao, C.W. Reed, R.E. Hudson, D-C. Chen and J. Chen. "Source localization and time delay estimation using constrained least-squares and best-path smoothing." *In "Advanced Signal Processing Algorithms, Architectures, and Implementations IX (Editor: Franklin T. Luk)", Proceedings of The International Society for Optical Engineering,* Volume 3807, pp. 220–233, Jul 1999.

[45] A. Urruela and J. Riba'. "Novel closed-form ML position estimator for hyperbolic location." *ICASSP'04: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Volume 2, pp. 149–152.

[46] M.C. Vanderveen, C.B. Papadias, and A. Paulraj. "Joint Angle and Delay Estimation (JADE) for Multipath Signals Arriving at an Antenna Array." *IEEE Communication Letters,* Volume 1, Issue 1, pp. 12–14, Jan 1997.

[47] H. Wang and P. Chu. "Voice Source Localization for Automatic Camera Pointing System in Videoconferencing." *ICASSP'97: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Volume 1, pp. 187–190.

[48] M. Wax and A. Leshem. "Joint estimation of time delays and directions of arrival of multiple reflections of a known signal." *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Volume 45, Issue 10, pp. 2477–2484, Oct 1997.

[49] B. Yang and J. Scheuing. "Cramer-Rao bound and optimum sensor array for source localization from time differences of arrival." *ICASSP'05: 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Volume 4, pp. 961–964

[50] B. Yang'and J. Scheuing. "A theoretical analysis of 2D sensor arrays for TDOA based localization." *ICASSP'06: 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Volume 4, pp. 901–904

[51] B. Yang. "Different sensor placement strategies for TDOA based localization." *ICASSP'07: 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Volume 4, pp. 1093–1096

[52] K. Yang, G. Wang and Z.Q. Luo. "Efficient convex relaxation methods for robust target localization by a sensor network using time difference of arrivals." *IEEE Transactions on Signal Processing,* Volume 57, pp. 2775–2784, Jul 2009.

Received: June, 2010

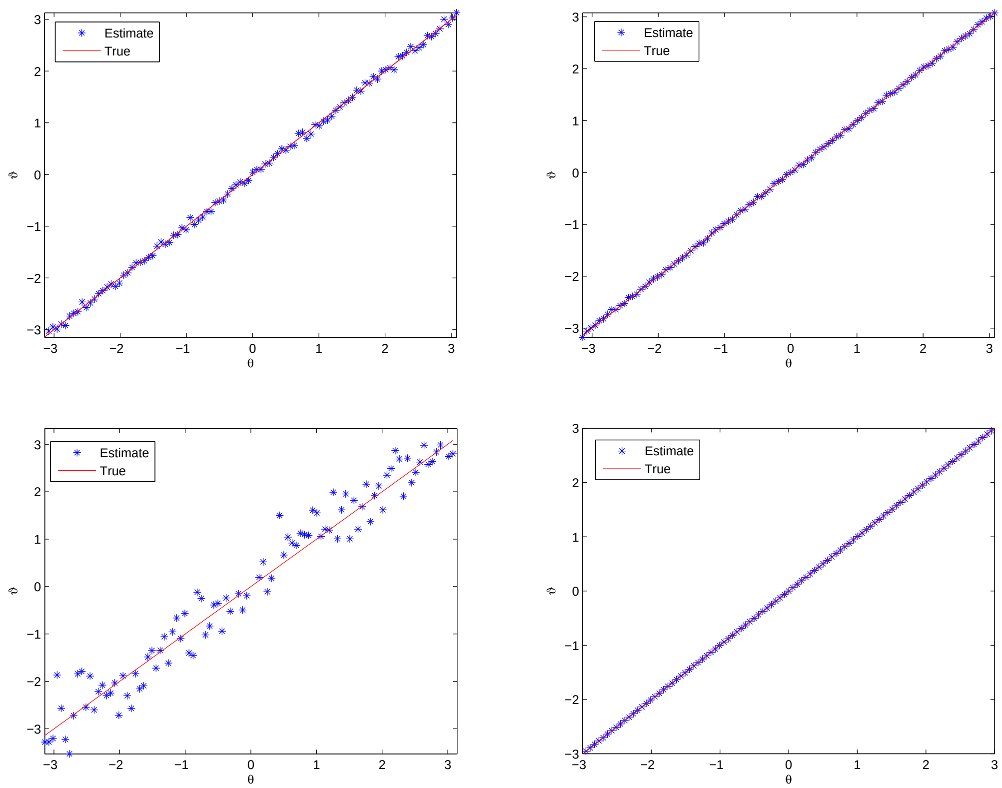

Figure 2: The estimated (by our localization algorithm) argument ϑ as a function of the actual argument θ of a point moving around the circle of radius $r = 1000$, when times of arrival (top left) and time delays of arrival (top right) are known (in both cases, continuum placement is assumed and $\epsilon = 10$); the bottom left figure shows the corresponding estimate by the algorithm described in [28] assuming known times of arrival, while the bottom right figure shows the corresponding estimate by the ML grid search [33] algorithm.

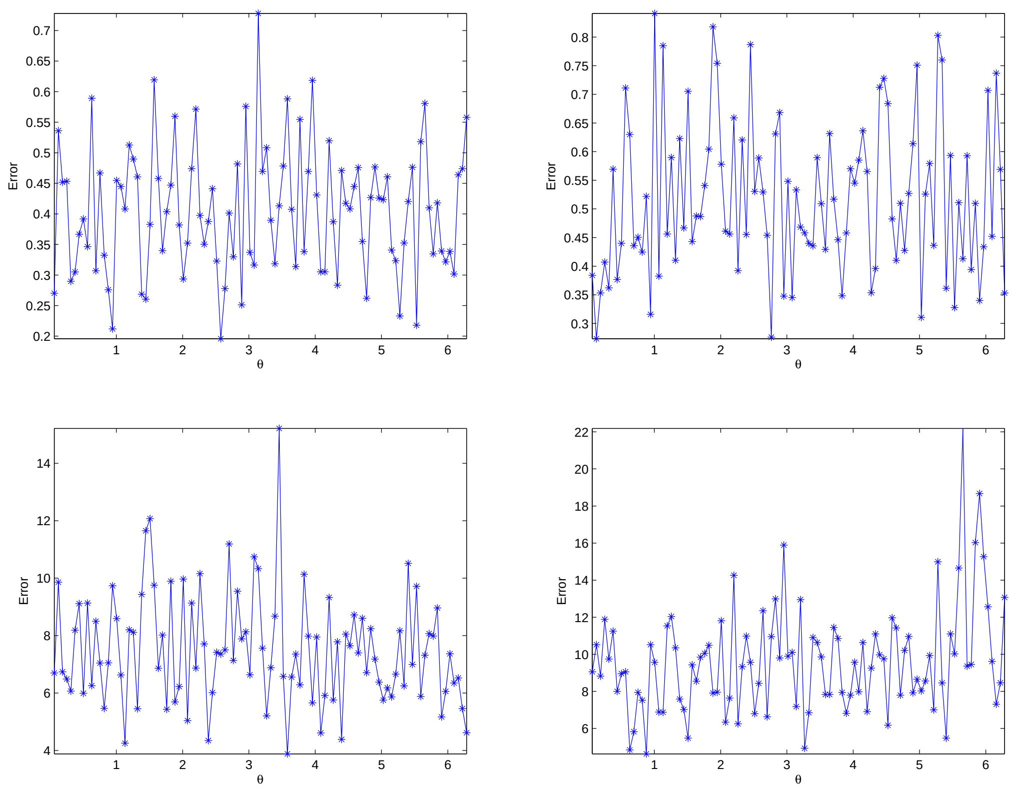

Figure 3: Localization error as a function of the argument θ for a point moving on a circle of radius $r = 100$, when $\epsilon = 1$ and under continuum placement. The topologies studied are circular (left) and Costas (right), assuming times of arrival (top) or time delays of arrival (bottom) are known.