國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering & Computer Science

National Taiwan University

master thesis

利用雙曲線的幾何特性對於適性無線電波干涉定位系統的定位精準度進行模型化與最佳化

Modeling and Optimizing Positional Accuracy based on Hyperbolic Geometry for the Adaptive Radio Interferometric Positioning System

吳昊極

Wu, Hao-ji

指導教授：朱浩華 博士

Advisor: Chu, Hao-hua, Ph.D.

中華民國 96 年 6 月

June, 2007

謝辭

我要謝謝創文和鶴齡，我們一起完成了這個 project，你們真的幫了我很多很多忙，你們對這個 project 的貢獻和我一樣多。

我要謝謝我的指導教授朱浩華老師，感謝您指導我如何作研究。我也要謝謝我的口試委員：黃寶儀老師、王傑智老師、許永真老師、廖世偉老師，你們在我的口試中提出了很多寶貴的意見。

我要謝謝實驗室的所有成員：鎮鴻、東昀、培堯、人豪、麗珊、佩妤、崔文，謝謝你們對我的實驗的幫忙，我在實驗室的這兩年和各位相處愉快。我也要謝謝曾經給這個 project 意見的人：力韋、Seng-Yong Lau、子桓，還有這兩年間陪伴我的朋友們。

我還要謝謝 Vanderbilt University 的 Brano Kusy 和 Miklós Maróti，謝謝你們熱心地解答我們的問題，你們寶貴的意見讓我們的 project 得以開始進行。

最後，謝謝我的爸爸，媽媽和姊姊。

摘要

「定位精準度」是評量一個定位系統效能最重要的指標，在現存的各種定位系統中，「無線電波干涉定位法（Radio Interferometric Positioning, RIP）」是少數可以達到公分尺度精準度的定位系統。在這篇畢業論文中，我們動態選擇環境中已知位置的感應點當作無線電波的發射點，藉此改善對於多個目標同時定位時的定位精準度。我們建立了一個誤差估計模型，這個模型可以預測使用任何發射點組合時的定位誤差。利用這個誤差估計模型，我們發展出「適性的無線電波干涉定位法（Adaptive RIP）」，可以根據定位目標的位置選擇最佳的發射點組合。我們實做 Adaptive RIP，並且在實際環境中實驗。根據我們的實驗結果，不管在單一目標或是多個目標，Adaptive RIP 的定位精準度都比原本的 RIP 好，平均定位精準度進步了 47%~60%。

Abstract

One of the most important performance objectives for a localization system is positional accuracy. It is fundamental and essential to general location-aware services. The radio interferometric positioning (RIP) method [1] is an exciting approach which promises sub-meter positional accuracy. In this work, we would like to enhance the RIP method by dynamically selecting the best anchor nodes as beacon senders, and further optimizing the positional accuracy when tracking multiple targets. We have developed an estimation error model to predict positional error of the RIP algorithm given different combinations of beacon senders. Building upon this estimation error model, we further devise an adaptive RIP method that selects the optimal sender-pair combination (SPC) according to the locations of targets relative to anchor nodes. We have implemented the adaptive RIP method and conducted experiments in a real sensor network testbed. Experimental results have shown that our adaptive RIP method outperforms the static RIP method in both single-target and multi-target tracking, and improves the average positional accuracy by 47%~60% and re-duces the 90% percentile error by 55%~61%.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many ubiquitous computing applications require deployment of a sensor network infrastructure to collect a variety of data sensed from the physical world. These sensor data are then processed to implement different digital services that can exhibit intelligent context-aware behaviors by automatically adapting their services to changing environments. In order to make correct inference on these sensor data, these systems require reliable, accurate location information on the observed sensor data. This brings up the need for accurate location tracking in sensor networks.

To address this need, there have been many sensor network localization systems utilizing different sensing techniques, e.g., MoteTrack [9], Cricket [10], Spotlight [13], APIT [14], ENSBox [16], etc. Among them, the radio interferometric positioning (RIP) method from the NEST project [1] has shown a promising, exciting location sensing technique for sensor network applications. Its main advantages are (1) sub-meter positional accuracy (e.g., in the range of tens of centimeters), (2) a long sensing range (e.g., 50~100 meters between two anchor nodes), and (3) no additional hardware requirement (i.e., reusing the same radio module for both communication and localization).

In this work, our innovations come in two parts. First, we have developed an estimation error model for the RIP method, which can accurately predict the amount of estimation error given the relative positions of anchor sensor nodes and (moving) target nodes. We have also validated the correctness of this estimation error model empirically from an experimental sensor network testbed. Building upon this estimation error model, our second innovation is the design and implementation of an adaptive RIP method that dynamically chooses the best anchor nodes in locating targets and minimizes their positional error. Our experimental results have demonstrated that our adaptive RIP method outperforms the static RIP method, improving the average positional accuracy by 47%~60% and reducing the 90% percentile error by 55%~61%.

The remainder of this paper is organized as follows. Section 2 provides background on the basic RIP system, and formulates our multi-target tracking problem. Section 3 derives our estimation error model that can accurately predict the estimation error in the RIP method. Section 4 presents the design of our adaptive RIP method. Section 5 describes its implementation. Section 6 explains the experimental setup and results. Section 7 discusses related work. Section 8 draws our conclusion and future work.

Chapter 2

Background on the Radio Interferometric Positioning (RIP)

We first provide a brief background on the original, single-target RIP method, followed by our multi-target tracking RIP extension. For a more detailed description of the original RIP method, we refer interested readers to [1].

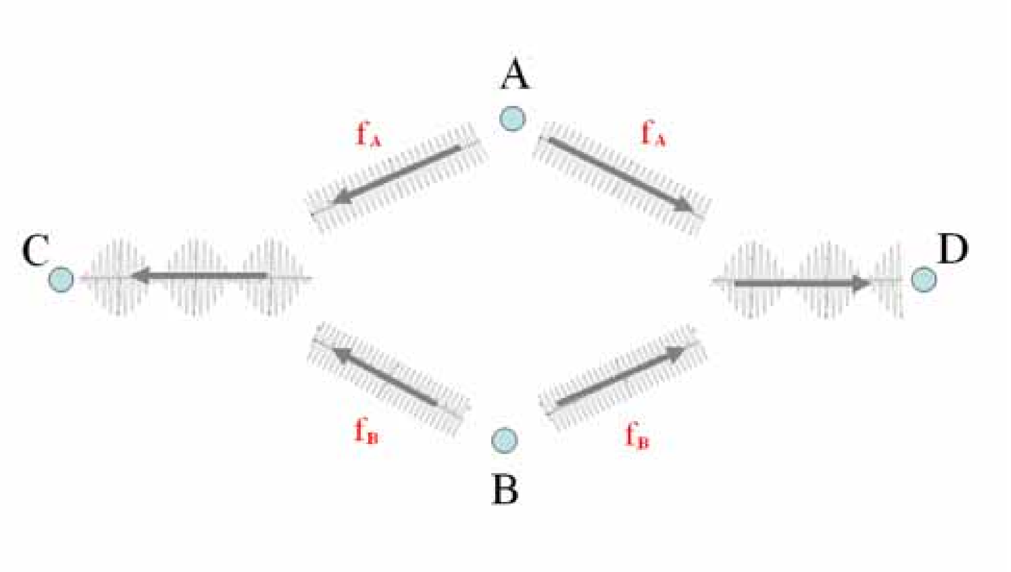


Figure 1. The RIP method

The RIP method is a novel way of localizing targets by measuring relative phase offset. It is typically realized in a sensor network setting [1], involving at least three anchor nodes and one target node, all within a common radio communication range as shown in Figure 1. Among the anchor nodes, two of them, *A* and *B*, act as senders and transmit pure sine wave simultaneously with two close frequency f_A and f_B. At nearby frequency, these two signals interfere with each other, therefore producing a resulting signal with a low beat frequency $|\, f_B - f_A \,|$. For example, our experiments using two MICA2 Motes with 900 MHz radio showed that interference produced a signal with a low beat frequency around 350 Hz. Two sensor nodes *C* and *D* act as receivers and can use simple hardware afforded on inexpensive sensor nodes to detect the phase of this low-beat interference wave.

Based on the relative phase difference detected on the receivers *C* and *D*, a geometric constraint among the locations of *A, B, C*, and *D* can be derived in the following equation (the details of this derivation is described in [1]):

$$\Delta\phi = \frac{2\pi}{\lambda}\,(d_{AD} - d_{BD} + d_{BC} - d_{AC})\ (mod\ 2\pi) \tag{1}$$

where $\lambda = \frac{2c}{f_A + f_B}$, $\Delta\phi$ is the phase difference detected by receivers *C* and *D*, λ is the wavelength of the mean carrier frequency of the interference signal, and d_{XY} is the distance between nodes *X* and *Y*. Furthermore, Equation (1) can be reformulated as follows:

$$\frac{\Delta\phi}{2\pi}\,\lambda = d_{ABCD}\ (mod\ \lambda)\ ,\ where\quad d_{ABCD} = d_{AD} - d_{BD} + d_{BC} - d_{AC} \tag{2}$$

In Equation (2), d_{ABCD} is also denoted as q_{range}. Due to (mod λ)-related ambiguity of d_{ABCD}, there can be more than one values of d_{ABCD} satisfying Equation (2). In order to resolve this d_{ABCD} ambiguity, the system must take multiple measurements (e.g., N times) at slightly different frequency channels (or different wavelength $\lambda_{i=1..N}$) and obtain corresponding phase differences $\Delta\phi_i$. Since each (λ_i, $\Delta\phi_i$) pair provides an instance of Equation (2), measuring N channels brings N such equations. To see how well a d_{ABCD} value fits this set of equations, an error function is defined below. By trying different values of d_{ABCD}, it is possible to find the best-fit one that minimizes this error function:

$$error\ (d_{ABCD}) = \sqrt{\sum_i \left(\frac{\Delta\phi_i}{2\pi}\lambda_i - mod(d_{ABCD}, \lambda_i)\right)} \tag{3}$$

After q_{range} is obtained, there are two approaches to track targets as proposed in [4]: *target-as-sender* or *target-as-receiver*. In the target-as-sender approach, a target is also a sender, and only one target can perform ranging operation in each measurement round. On the other hand, in the target-as-receiver approach, multiple targets can perform ranging operations in each measurement round independently and simultaneously. The tradeoff between these two approaches is concurrency (target-as-receiver is better) versus accuracy (target-as-sender is better). We adopt the target-as-receiver approach for our multi-target tracking extension. We first describe the single-target tracking mechanism, followed by the multi-target extension.

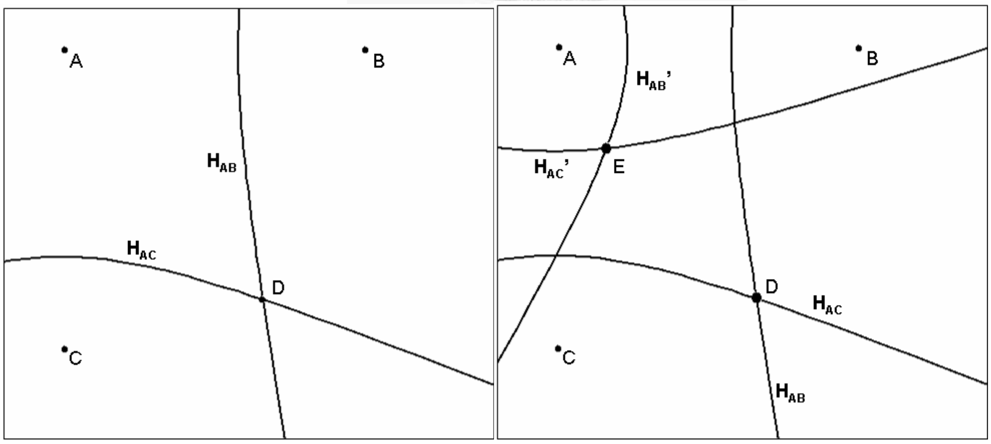
Consider the example in Figure 2(a), node D is the tracked target, and nodes A, B, and C are anchor nodes with known locations. Then, d_{ABCD} can be transformed into the following equation:

$$d_{ABCD} + d_{AC} - d_{BC} = d_{AD} - d_{BD} \tag{4}$$

Since the left hand side of Equation (4) contains only variables with known values, its value can be calculated, and is referred to as t_{range}. Equation (4) can be further rewritten as follows:

$$t_{range} = d_{AD} - d_{BD} \tag{5}$$

Equation (5) can be drawn as a hyperbolic curve H_{AB} shown in Figure 2(a). It is one arm of a hyperbola with two foci A and B passing through D with the semi-major axis of the length $t_{range}/2$. In other words, D can lie anywhere on this hyperbolic curve H_{AB}. To precisely locate D, each positioning operation must take a second measurement round using a different pair of senders. In Figure 2(a), the second measurement round selects nodes A and C as senders, and nodes B and D as receivers. This gives another hyperbolic curve H_{AC}. The intersection of these two hyperbolic curves (H_{AB} and H_{AC}) fixes the location of D. In this example, *(A, B)* and *(A, C)* are called *sender-pair combination (SPC)*. We can think of each measurement round selects one *pair of senders* to jointly localize a moving target.



(a) (b)

Figure 2. Tracking (a) single and (b) multiple targets with intersections of hyperbolic curves.

Multi-target Extension

Tracking multiple targets simultaneously is similar to single-target tracking, except that each target measures and calculates its own t_{range} independently and concurrently from Equations (1~5). Figure 2(b) shows a two-target case of nodes *D* and *E*. Assume that {(*A, B*), (*A, C*)} are selected as SPC. This gives two pairs of hyperbolic curves: (H_{AB}, H_{AC}) intersecting at the point *D,* and (H_{AB}', H_{AC}') intersecting at the point *E*. In the multi-target extension, we can think of each measurement round selects one *sender pair* to jointly localize *all* moving targets within a common radio range.

Since the targets' locations are estimated from the intersection of hyperbolic curves, geometric properties of the curves at the intersection points can significantly impact the amount of estimation error in RIP. The reason is that these hyperbolic curves inheritably have error due to imperfect measurements of q_{range} at the receivers. This error may be amplified to different amount depending on the curves' geometric properties. These geometric properties are in turn dependent on the SPC (sender pair combination) selection. Consider the examples in Figure 3(a) and 3(b). They share the same spatial layout of sensor nodes but different SPC selection: Figure 3(a) selects *{(S1, S2)*, *(S1, S4)}* and Figure 3(b) selects *{(S1, S3)*, *(S2, S4)}*. The black lines represent hyperbolic curves that perfectly intersect at the target *T*, whereas the gray lines represent hyperbolic curves with an error added to the q_{range} calculation. As shown in these examples, the same amount of q_{range} error is amplified differently, causing more estimation error in locating the target *T* in Figure 3(a) than in Figure 3(b). There are two geometric factors contributing to this error amplification (positional error is $|T' - T|$): (1) the intersectional angle formed between H_{12} and H_{14} is more acute than the intersectional angle between H_{13} and H_{24}, and (2) the amount of displacement between the black and gray lines is larger in Figure 3(a) than in Figure 3(b) at the intersection points.

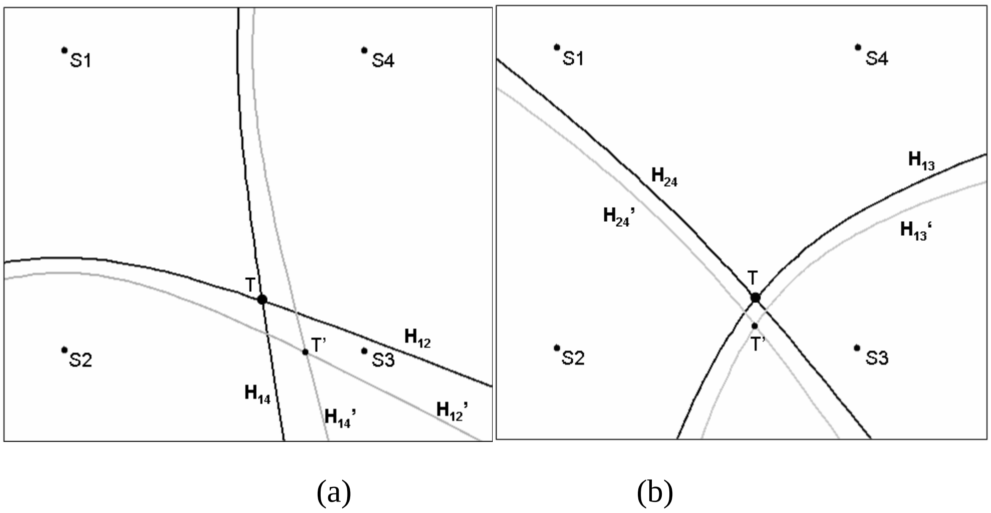


Figure 3. Show the how different SPC selections and their produced geometric factors (the intersectional angle and the displacement of hyperbolic curve) affect the estimation error in locating the target *T*. Figure 3(a) selects *{(S1, S2), (S1, S4)}* as SPC, and Figure 3(b) selects *{(S1, S3), (S2, S4)}* as SPC. The black lines represent perfect hyperbolic curves that intersect on the target *T*. The gray ones represent hyperbolic curves with the same amount of error added to the q_{range} calculation, and intersect on *T'*.

Although Figure 3 shows the effect of these two geometric factors in the single-target tracking, such effect is also applicable to multi-target tracking; except that for a given SPC selection, each target node at different location can experience different effect and varying amount of estimation error. Nevertheless, a good SPC selection is still the key in reducing estimation error in multi-target tracking. Multi-target tracking must consider estimation errors of all moving targets and optimize them as a whole, rather than simply consider the estimation error of a single target in single-target tracking. More detailed analysis of the estimation error is described in Section 3.

Multi-target Problem Formulation

Since different sender-pair-combination (SPC) gives different amount of estimation error, we can turn this multi-target tracking into an optimization problem as follows. Given a set of infrastructure anchor nodes with fixed known locations ($P_{1..m}$), and a set of moving target nodes ($T_{1..n}$) sharing the same radio range as these anchor nodes. Each anchor node can be assigned either a sender or a receiver dynamically. Define the estimation error as the difference between the actual (ground-truth) position and the position estimated by the radio interferometric positioning engine. Design an optimization scheme in which by dynamically selecting a set of SPC from ($P_{1..m}$) to localize targets, minimizes the average estimation error of all targets ($T_{1..n}$).

The SPC selection algorithm mentioned above is described in more details in Section 4, which must utilize an *estimation error model* that can accurately approximate the amount of error given a specific SPC selection. The following section explains this estimation error model.

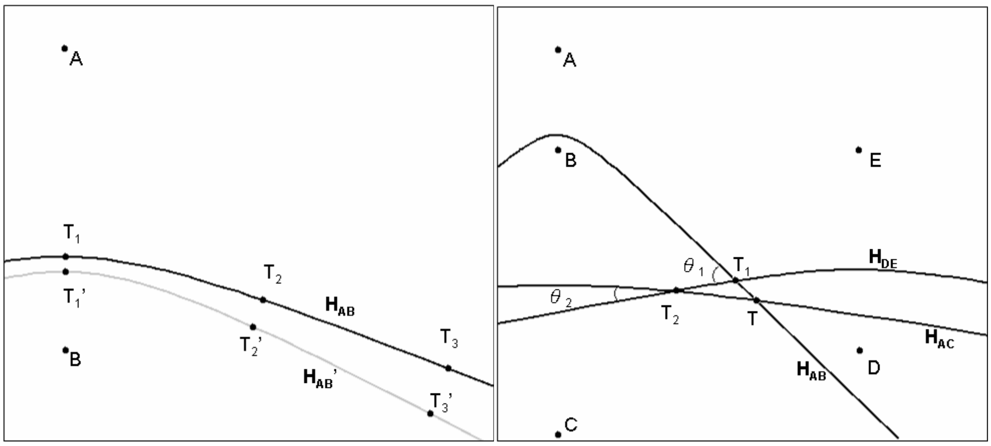
Chapter 3

Estimation Error Model

Given a specific sender-pair combination (SPC) selection and a target node, the estimation error model can accurately approximate the amount of estimation error from a RIP engine. To derive this estimation error model analytically, we first identify factors that contribute to the positional error: (1) q_{range} *estimation error* *(q_{error})*: it comes from imperfect phase difference measurements at the receivers, leading to the error in finding the best-fit d_{ABCD} from Equation (3); (2) *displacement of a hyperbolic curve*: the minimum distance from the deviated hyperbolic curve to the target; and (3) *intersectional angle of hyperbolic curves*.

To explain how these factors contribute to the estimation error, consider the examples in Figure 4. First, we describe how depending the target *T's* position on the curve, a displacement *of a hyperbolic curve* can cause different amount of estimation error. In Figure 4(a), the pair of senders (A, B) can produce the perfect hyperbolic curve H_{AB} under no q_{error}, and a slightly displaced hyperbolic curve H_{AB}' under q_{error}. If the target is at T_1, the closest point between the two hyperbolic curves is T_1'. Regardless of how the other hyperbolic curve intersects with the H_{AB}', $|T_1' – T_1|$ becomes the minimum estimation error in locating T_1 under q_{error}. Next, we can observe how the estimation error grows when the target position moves to T_2 and grows even larger when it moves to T_3.

Second, we want to discuss how the intersectional angle of two hyperbolic curves can cause different amount of estimation error in locating a target *T*. We use the example in Figure 4(b). Suppose that the first measurement round produces a hyperbolic curve H_{DE} with q_{error}, and the second measurement round produces a perfect hyperbolic curve without any q_{error}. Consider two such perfect curves H_{AB} from the pair of senders *(A, B)*, and H_{AC} from *(A, C)*. From the Figure 4(b), H_{AB}/H_{AC} has a different intersection point of T_1/T_2. In addition, we can observe that because the intersectional angle θ_1 at T_1 is wider than the intersection angle θ_2 at T_2, the positional error of T_1 is smaller than the positional error of T_2.



(a) (b)

Figure 4. (a) The displacement of a hyperbolic curve changing with target location. (b) Positional error changing with intersectional angle.

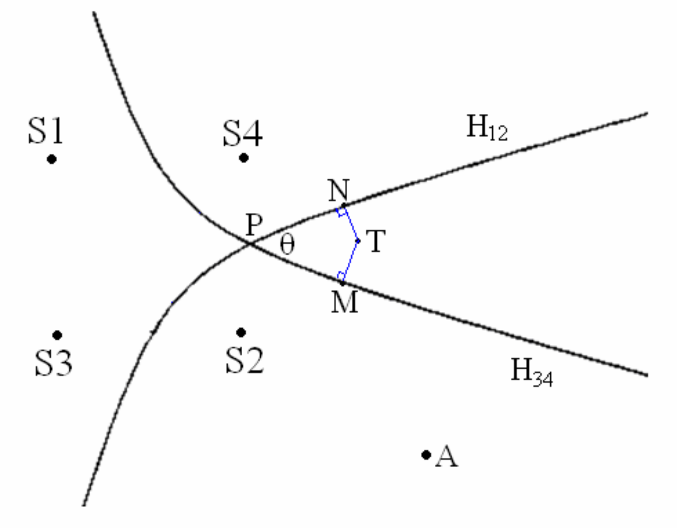


Figure 5. Intersection of two hyperbolic curves by using SPC *{(S1, S2), (S3, S4)}*.

Analytic Expression. We derive the estimation error model mathematically. Consider the single-target case in Figure 5, *S1~S4* are anchor nodes, and *T* is the target. The *SPC:{(S1, S2), (S3, S4)}* gives two hyperbolic curves of H_{12} and H_{34}. Given q_{error}, these two hyperbolic curves intersect on *P*. $\overline{TP}$ is the distance between the target's ground-truth position *T* and the estimated position *P*, or the estimation error.

If the target *T* is not so close to the focus of the hyperbola, i.e., the curvature of the hyperbola around the target *T* is relatively flat, $\overline{PN}$ and $\overline{PM}$ could be approximated as straight lines[1]. Under such an assumption, we calculate $\overline{TP}$ by solving the geometric problem shown in Figure 6. Values unknown are $\overline{TN}$, $\overline{TM}$ (namely, the displacement), and intersectional angle θ. Described in the following is the method we use to obtain the unknown values.

[1] According to the mathematical simulation, as long as the distance between the target and one of the foci is greater than 0.2m, the approximation will be good in case of 20m major-axis length.

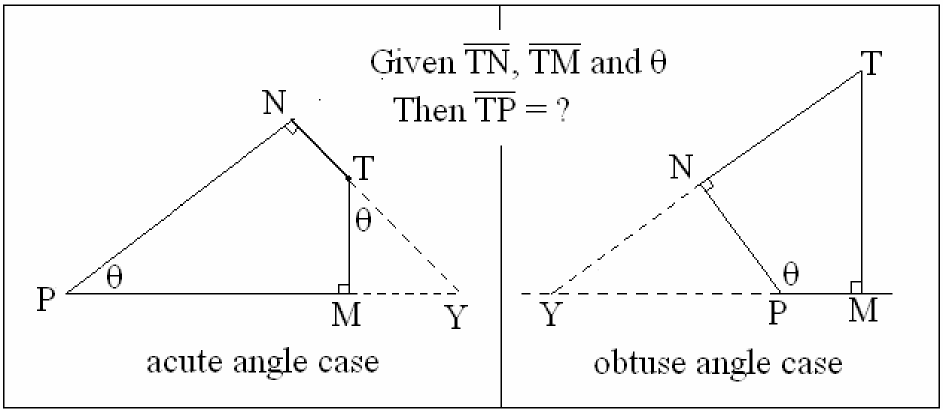


Figure 6. Approximate the real positional error by $\overline{TP}$.

Model the displacement of a hyperbolic curve. In the first measurement round, (*S1, S2)* are selected as senders, and *(A,T)* as receivers. This gives the first hyperbolic curve *H12*. Since (*S1, S2)* and *A* have known locations and *T*'s location is unknown, their geometric relation, by substituting into Equation (2), is as follows:

$$d_{S1,S2,A,T} = d_{S1,T} - d_{S2,T} + d_{S2,A} - d_{S1,A} = q_{range}, \quad where \begin{cases} S1 = (S1_x, S1_y) \\ S2 = (S2_x, S2_y) \\ A = (A_x, A_y) \\ T = (T_x, T_y) \end{cases} . \tag{6}$$

Rewrite Equation (6) by substituting these coordinates:

$$q_{range} - (d_{S2,A} - d_{S1,A}) = d_{S1,T} - d_{S2,T} = \sqrt{(T_x - S1_x)^2 + (T_y - S1_y)^2} - \sqrt{(T_x - S2_x)^2 + (T_y - S2_y)^2}. \tag{7}$$

If q_{range} has no error, Equation (7) is a hyperbolic curve which passes through *T*(T_x,T_y). However, when q_{error} is added to q_{range}, its hyperbolic curve is displaced from *T*. We describe a method to approximate the amount of displacement. Since the only non-constant terms in Equation (7) are q_{range} and *T*(T_x,T_y), q_{range} can be written as a function of *T*(T_x,T_y), the gradient of $q_{range}(T_x,T_y)$ is derived as follows:

$$
\begin{aligned}
&q_{range}(T_x,T_y)=\sqrt{(T_x-S1_x)^2+(T_y-S1_y)^2}-\sqrt{(T_x-S2_x)^2+(T_y-S2_y)^2}+(d_{S2,A}-d_{S1,A})\\
&\nabla q_{range}(T_x,T_y)=\left(\frac{T_x-S1_x}{d_{S1,T}}-\frac{T_x-S2_x}{d_{S2,T}}\ ,\ \frac{T_y-S1_y}{d_{S1,T}}-\frac{T_y-S2_y}{d_{S2,T}}\right).
\end{aligned}
\tag{8}
$$

By the definition of gradient, $\left|\nabla q_{range}(T_x,T_y)\right|$ is the maximum changing rate of $q_{range}(T_x,T_y)$. That is, if target T is shifted by a small ε movement, the maximum q_{range} incremental change is $\varepsilon\left|\nabla q_{range}(T_x,T_y)\right|$. Equivalently, to produce this q_{error}, $\frac{q_{error}}{\left|\nabla q_{range}(T_x,T_y)\right|}$ is the minimum movement of the target T on the displaced hyperbola H_{12}. The minimum movement is a good approximation of $\overline{TN}$ when q_{error} is small.

Denote q_{error1} and $q_{range1}(T_x,T_y)$ as q_{error} and $q_{range}(T_x,T_y)$ measured in the 1st round, and q_{error2} and $q_{range2}(T_x,T_y)$ in the 2nd round. By applying the above approximation to these two measurement rounds, $\overline{TN}$ and $\overline{TM}$ can be obtained as follows:

$$
\overline{TN}=\frac{q_{error1}}{\left|\nabla q_{range1}(T_x,T_y)\right|},\quad \overline{TM}=\frac{q_{error2}}{\left|\nabla q_{range2}(T_x,T_y)\right|}. \tag{9}
$$

Model the Intersectional angle of two hyperbolic curves. The intersectional angle θ can be approximated by the tangent slopes at N (denotes m_N and m_M respectively) :

$$
\theta=tan^{-1}\left(\frac{m_N-m_M}{1+m_N m_M}\right). \tag{10}
$$

If the coordinates of N and M are known, the tangent slopes at N and M can be obtained. Since we can approximate $\overline{TN}$ and $\overline{TM}$ and the unit vector from T to N

(which runs parallel to $\nabla q_{range1}(T_x, T_y)$), we can obtain the coordinate of $N(N_x, N_y)$ by the following equation:

$$N(N_x, N_y) = T(T_x, T_y) + \frac{q_{error1}}{\left|\nabla q_{range1}(T_x, T_y)\right|} \frac{\nabla q_{range1}(T_x, T_y)}{\left|\nabla q_{range1}(T_x, T_y)\right|} \cdot \tag{11}$$

Note that the last term is the product of $\overline{XN}$ and the unit vector from T to N. In addition, we can obtain M in a similar way:

$$M(M_x, M_y) = T(T_x, T_y) + \frac{q_{error2}}{\left|\nabla q_{range2}(T_x, T_y)\right|} \frac{\nabla q_{range2}(T_x, T_y)}{\left|\nabla q_{range2}(T_x, T_y)\right|} \cdot \tag{12}$$

Return to our original problem – solving the length of $\overline{TP}$. First, extend $\overline{TN}$ and $\overline{PY}$ to the intersection point Y to form a triangle as shown in Figure 6. After obtaining the intersectional angle θ, $\overline{TN}$, and $\overline{TM}$ from the above approximation, $\overline{TP}$ can be solved geometrically.

$$For\ \Delta MTY : \begin{cases} \overline{TY} = \overline{TM}\ sec\theta \\ \overline{MY} = \overline{TM}\ tan\theta \end{cases} \tag{13}$$

$$For\ \Delta NPY : \begin{cases} \overline{NY} = \overline{PY}\ sin\theta \\ \overline{NY} = \overline{TN} + \overline{TY} \\ \overline{PY} = \overline{PM} + \overline{MY} \end{cases} \tag{14}$$

According to Equation (14), we obtain:

$$\overline{PM} = \overline{PY} - \overline{MY} = \frac{\overline{NY}}{sin\theta} - \overline{MY} = \frac{\overline{TN} + \overline{TY}}{sin\theta} - \overline{MY} \cdot \tag{15}$$

Substitute Equation (13) into Equation (15):

$$\overline{PM} = \frac{\overline{TN} + \overline{TY}}{sin\theta} - \overline{MY} = \frac{\overline{TN} + \overline{TM}\ sec\theta}{sin\theta} - \overline{TM}\ tan\theta \cdot \tag{16}$$

Apply Pythagorean Theory to ΔTMP and combine with Equation (16):

$$\begin{aligned}\overline{PT} &= \sqrt{\overline{PM}^2 + \overline{TM}^2} \\ &= \sqrt{\left(\frac{\overline{TN} + \overline{TM}\ sec\,\theta}{sin\,\theta} - \overline{TM}\ tan\,\theta\right)^2 + \overline{TM}^2}\,.\end{aligned} \tag{17}$$

Finally, we substitute Equation (9) into Equation (17) to obtain the final estimation error:

$$\overline{PT} = \sqrt{\left(\frac{\frac{q_{error1}}{\left|\nabla q_{range1}(T_x,T_y)\right|} + \frac{q_{error2}}{\left|\nabla q_{range2}(T_x,T_y)\right|}sec\theta}{sin\theta} - \frac{q_{error2}}{\left|\nabla q_{range2}(T_x,T_y)\right|}tan\theta\right)^2 + \left(\frac{q_{error2}}{\left|\nabla q_{range\,2}(T_x,T_y)\right|}\right)^2} \cdot \tag{18}$$

If the intersectional angle θ is obtuse, there will be some minor differences in the deduction from Equations (13) ~ (17). However, despite these differences at the intermediate steps, the solution to $\overline{PX}$ is still the same as in Equation (18).

Experimental Validation of Estimation Error model

We have designed and conducted two experiments to validate the correctness and accuracy of the estimation error model derived above. These two experiments differ on what parameters, in the estimation error model, are considered as known (observable) values or not. For example, at localization runtime, the values of q_{error1}

and q_{error2} are not observable, since they would require the knowledge of the target's ground-truth position.

To approximate these q_{error} values, we introduce a *calibration phase* prior to runtime when samples at known locations are collected. Then the average $|q_{error}|$ is calculated, and it can be used as the magnitude of q_{error1} and q_{error2} in our estimated error model. Note that the sign (+/-) for q_{error1} and q_{error2} is unknown. Therefore, for each position estimation, there are four possible +/- combinations of q_{error1} and q_{error2}. Consider the example in Figure 7. For the target T, H_{14} and H_{12} are the hyperbolic curves created from the sender pairs *(S_1,S_4)* and *(S_1,S_2)* with no q_{error}. When different combinations of plus/minus q_{error1} and q_{error2} are substituted into our estimated error model, we obtain four estimated positions (T_1 ~ T_4) and estimation errors ($|T – T_1|$, $|T – T_2|$, $|T – T_3|$, and $|T – T_4|$). We then use the average of these four estimation errors as the target's estimation error.

In our experimental setup, six anchor nodes were placed uniformly on the ring with ten meters radius. Five targets were placed insides this circle. We measure these target nodes with 7 different SPCs. For each SPC, about 50 samples were collected. Since we had the ground-truth location of each target, the magnitude and sign of q_{error} and the estimation error were determined. Using these data, we validated the correctnesss of our error prediction model by comparing the real error and the estimation error.

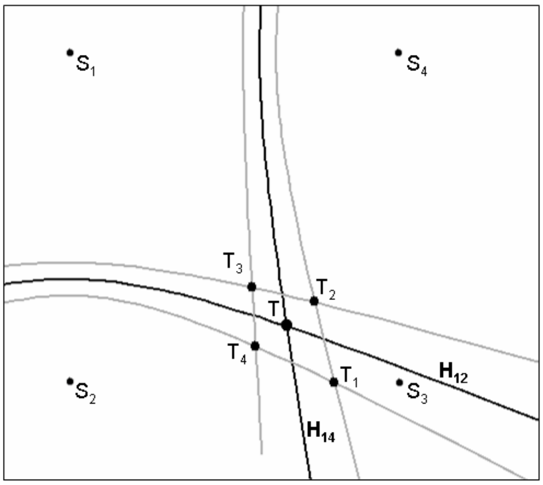


Figure 7. Four estimation positional errors utilized $|q_{error}|$ from the calibration phase

All Parameters Known Case. This case considers all the parameters in the estimation error model are known. Although this is unrealistic, we conducted this experiment for the purpose of verifying the correctness of our estimation error model. Figure 8 plots the real ground-truth error vs. the estimation error (p_{error}) calculated from our model in Equation (18). The red line plots a perfect diagonal line representing perfect error prediction, and the blue dots are our measurements. The results show that our estimation error model is accurate as it falls within 6 centimeters from the real error 90 % of the time.

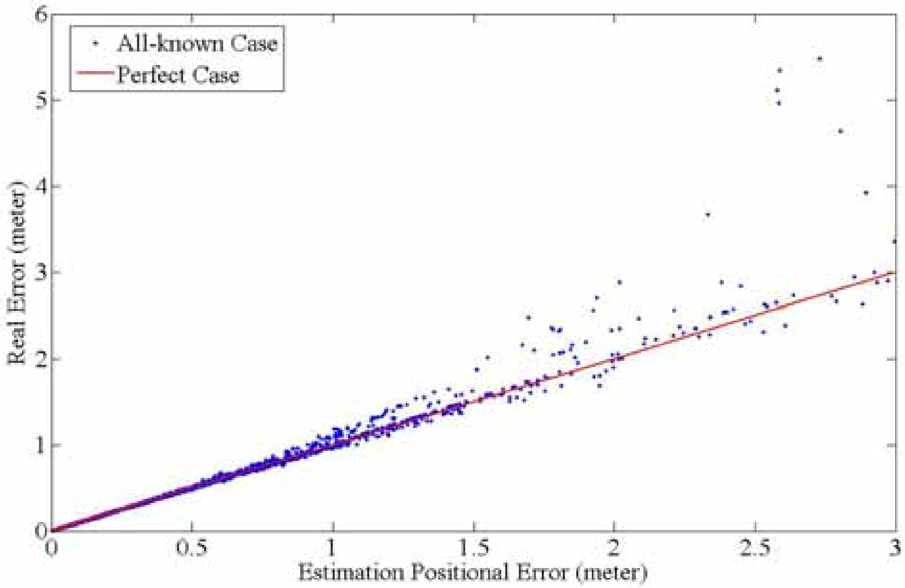


Figure 8. Validation of the estimation error model in the all-parameters-known case.

Runtime Case. At runtime, the system has no knowledge of the actual q_{error}, therefore, we use the average q_{error} obtained from a calibration phase, which is 26 centimeters, in estimation error model. Figure 9 plots the cumulative density function (CDF) of the difference between the real error and predicted error from our estimation error model. The average difference is 35 centimeters, which is not as good as in the All-Parameters-Known case, but sufficient for our error estimation purpose.

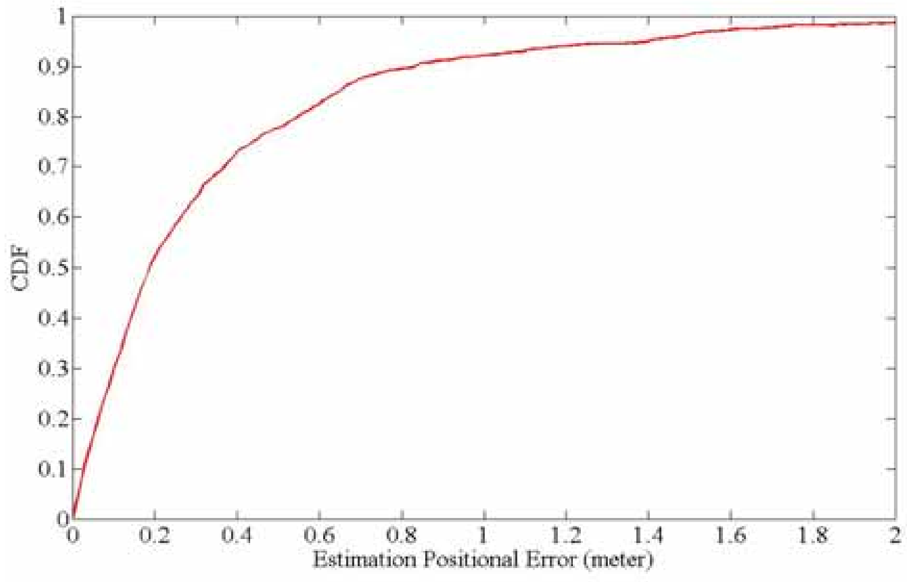


Figure 9. CDF of the estimation error for the runtime case.

Chapter 4

Design of the Adaptive RIP Method

The design of our adaptive RIP method is shown in Figure 10. It consists of following components: (1) *adaptive SPC selection algorithm*, (2) *the estimation error model*, and (3) *the radio interferometric positioning engine*. In the first step, the adaptive SPC selection algorithm is invoked to find the optimal anchor nodes as sender-pair-combination (SPC) that can locate mobile targets most accurately. To find the optimal SPC, the adaptive SPC selection algorithm currently performs an *exhaustive search* through all possible SPCs, and selects the SPC that gives the minimal estimation error.

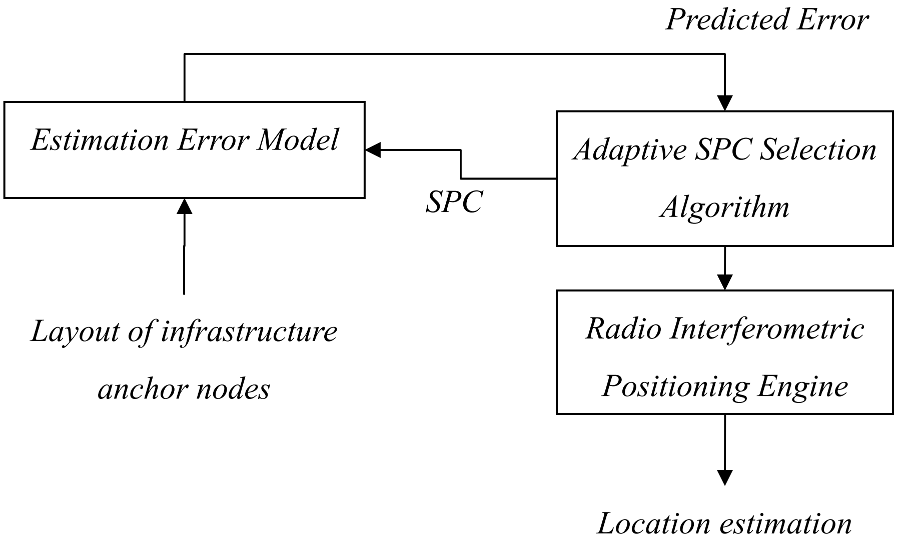

Predicted Error
Estimation Error Model
Adaptive SPC Selection Algorithm
SPC
Radio Interferometric Positioning Engine
Layout of infrastructure anchor nodes
Location estimation

Figure 10. System architecture of our adaptive RIP method.

Specifically, for each unique SPC combination, the adaptive SPC selection algorithm invokes the Estimation Error Model in Equation (18), and calculates its corresponding estimation error. There are three notable details here. First, the exhaustive search strategy is still computationally manageable, because the estimation error computation is relatively straightforward and the number of different combinations (proportional to the number of anchor nodes within the same radio range) is relatively small. Second, when tracking multiple targets, the Estimation Error Model computes an error for each target. If an application considers equal importance to all targets, an optimal SPC minimizes the aggregate error from all targets. Third, the Estimation Error Model requires the knowledge of the approximate locations of mobile targets, which be obtained by using the most recently estimated locations of the mobile targets. When the optimal SPC is selected, the system invokes the RIP engine to obtain the locations of targets.

Some applications may consider some high priority targets, demanding stricter accuracy requirement, as well as some lower priority targets, having looser accuracy requirement. Our system can provide an *optimization policy*, which allows location-based applications to specify preference policies for different classes of target nodes. Based on this policy specification, our adaptive SPC selection algorithm can intelligently choose SPC that favors accuracy of certain targets over the others.

Chapter 5

Implementation of the Adaptive RIP System

Our adaptive RIP system has been implemented on MICA2 Motes with 900 MHz radios made by Crossbow Inc. One MICA2 Mote connects to a laptop with MIB520 programming board and relays phase measurement packets to a positioning engine developed in Java. The MICA2 Motes are running TinyOS. We modified the Radio Interferometric Positioning (RIP) engine [20] released by Vanderbilt University and ported it to 900MHZ MICA2 Motes. In addition, we extended the RIP engine to implement multi-target tracking. In each measurement round, the base station (PC) sends a command with selected SPC information to all sensor nodes. After time synchronization is performed, the selected sender nodes transmit sine wave in predefined carrier frequency. At our test site location in Taiwan, the frequency band of GSM-900[11] also happens to be around 900MHz, overlapping with a part of MICA2 radio channels. To avoid interference from the GSM-900 up/down link channels, we selected 18 carrier frequencies between 821.277MHZ to 921.337MHZ whose ranges are away from GSM-900 channels. At the end of each measurement round, receiver nodes send back their phase measurement data to a base station. After the RIP engine collects phase measurement data from receivers, it estimates targets' locations.

Chapter 6

Experimental Results

We conducted two experiments to evaluate the accuracy performance of our adaptive RIP system in a real sensor network environment. The first experiment tested the single-target tracking, whereas the second experiment tested multi-target tracking.

Both experiments were performed on a square near the sport stadium of Nation Taiwan University, which is shown in Figure 11. The tracking area is a circle with 10-meter radius. Six infrastructure anchor nodes were deployed uniformly on the ring, and their locations (*A~F*) are marked in Figure 12(a).

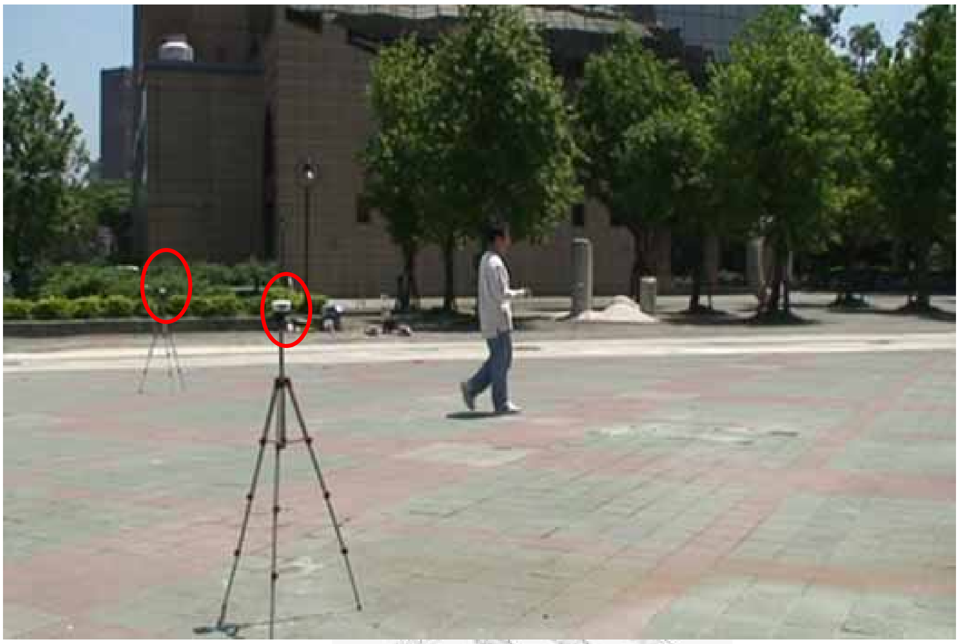
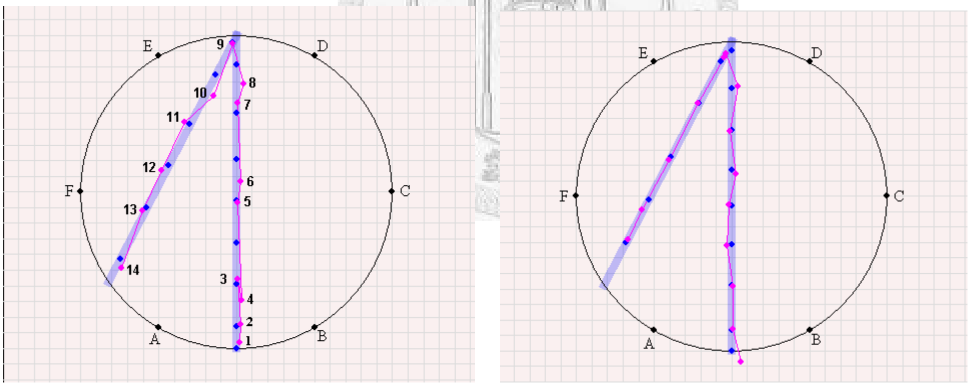


Figure 11. Experimental setup.



(a) the static RIP method (b) the adaptive RIP method

Figure 12. Results of static RIP method versus our adaptive RIP method. The blue line is the target's movement path. Blue dots indicate the target's ground-truth positions at location sampling time. Red dots show the estimated positions from static/adaptive RIP method.

Single-target tracking experiment. The first experiment tracked a single target, which was a person carrying a MICA2 Mote and walking under normal speed. His movement path is plotted as the blue line in Figure 12. This path was walked repeatedly 5 times for a total distance of 37 meters.

To show that our adaptive RIP can improve the positional accuracy of the original static RIP method, we repeated this experiment twice, once using the static RIP method and once using our adaptive RIP method, and then compared their positional accuracy results. Figure 12(a)/(b) shows the result from the static/dynamic RIP methods. For the static RIP method, two pairs of senders are selected a-priori and fixed to *{(B,C), (C,F)}* regardless of the changing position of the target. Blue dots indicate the target's ground-truth positions at the time of location samples, and red dots show the estimated positions from each of the RIP methods. Figure 12 shows our adaptive RIP method tracks the moving target more accurately than in the static RIP method. Table 1 shows the average positional error and the amount of improvement of our adaptive RIP method over the static RIP method: 47% reduction in average error and 55% reduction at 90% percentile error.

Table 1. Result comparison between the static and adaptive RIP methods in single-target tracking.

	Average error (meter)	90%-th percentile (meter)
Static RIP	0.93	1.66
Adaptive RIP	0.49	0.75
Improvement	47%	55%

Multi-target tracking experiment. The second experiment tracked six targets. The first five targets are stationary with their locations marked in Figure 13. The 6th target is mobile and follows the same movement path as in the first experiment.

Similar to the first experiments, we want to show that our adaptive RIP method can improve the positional accuracy of the static RIP method. Therefore, we repeated the experiment twice, once using the static RIP method and once using our adaptive RIP method, and then compared their positional accuracy results. Figure 14 shows the average positional error for each target using the static/adaptive RIP methods. Our adaptive RIP method reduces the average positional error in all the 6 targets. Table 2 summarizes the results, showing our adaptive RIP method with 60% reduction in average positional error and 61% reduction at the 90% percentile error.

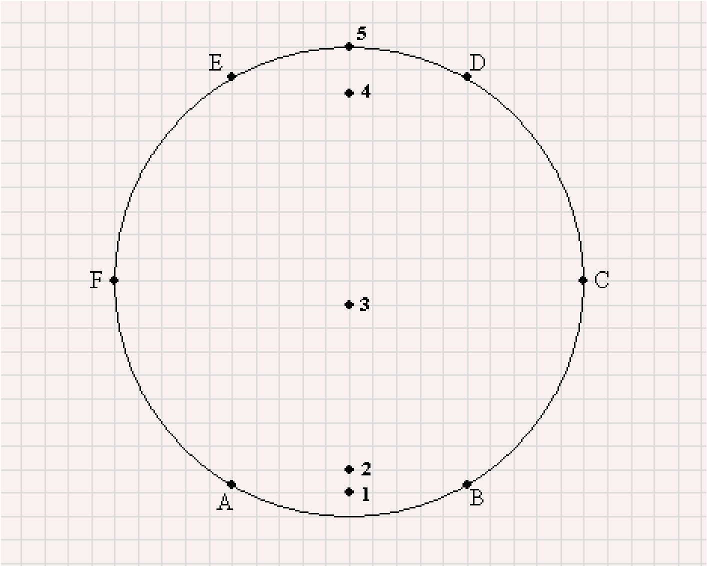


Figure 13. Stationary target distribution of multi-target tracking experiment.

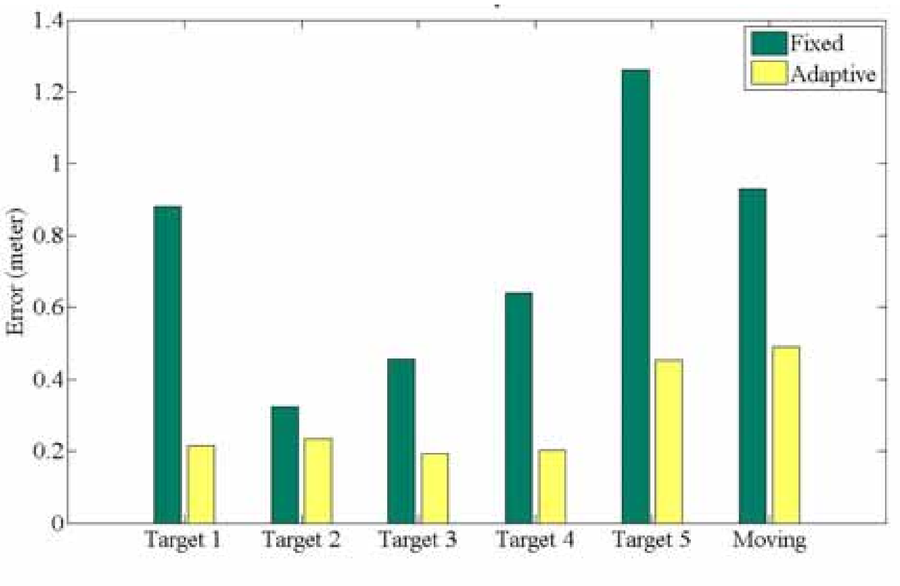


Figure 14. Average positional error of each target using the static/adaptive RIP methods.

Table 2. Performance result comparison between the static and adaptive RIP method in multi-target tracking.

	Average error (meter)	90%-th percentile (meter)
Static RIP	0.75	1.41
Adaptive RIP	0.30	0.54
Improvement	60%	61%

Table 3 provides insight into why our adaptive RIP method works better than the static RIP method. At a specific time point, estimation and real errors were recorded for 6 different SPCs as if each of them were used for positioning targets. They showed a wide range. The largest estimation/real error is 1.00/0.89 meter(s), which is 4.2/4.0 times of the smallest estimation/real error 0.24/0.22 meter. Our adaptive RIP method would select the optimal SPC#6, whereas the static RIP method could pick any SPC.

Table 3. Average estimation/error for locating each of five stationary targets using different SPCs

SPC #	1	2	3	4	5	6
First Sender Pair	(D,E)	(B,C)	(B,C)	(A,C)	(B,E)	(A,D)
Second Sender Pair	(D,F)	(C,D)	(C,F)	(B,D)	(C,F)	(B,E)
Average estimation error (meter)	1.00	0.84	0.60	0.37	0.33	0.24
Average real error (meter)	0.89	0.89	0.74	0.63	0.37	0.22

Chapter 7

Related Work

The most relevant related work is the Radio Interferometric Positioning (RIP) method from Vanderbilt University [1][2][3][4]. It is a novel way of localizing targets by measuring relative phase offset with inexpensive hardware on sensor nodes. They proposed a tracking system called inTrack [3] based on the RIP method. Their result shows that, in 80 meters x 90 meters area,it can track a moving target with sub meter accuracy. Furthermore, they have developed mTrack [4] to track multiple targets simultaneously. However, as the number of moving targets increases, mTrack may not be able to provide good positional accuracy to all moving targets with fixed sender pairs. This motivates our adaptive RIP method that dynamically selects sender pairs given the approximate positions of targets.

There are many other sensor network positioning systems. In general, they can be classified into two broad categories of ranging-based and ranging-free methods.

Ranging-based methods. They commonly require signal communications between an anchor observer and a locating target. The major differences among them are the calibration methods and the usage of signal sources, such as sonic, ultrasonic, infrared, camera, RF, etc. For example, Acoustic ENSBox [16] develops a distribution acoustic sensing platform, which an acoustic embedded networked sensing box can be rapidly deployed and perform self-calibration. It claims to achieve 5 centimeters positional accuracy in a partially obstructed 80m╳50m outdoor. Given that signal propagates with constant velocity, TOA (time of arrival) methods [18] estimate distance by measuring the signal propagation time. AOA (angle of arrival) [19] is a network-based solution that exploits the geometric property of the arriving signal. By measuring the angle of the signal's arrival at multiple receivers, it is able to provide an accurate location estimation. TDOA (time difference of arrival) [17] is also network-based. It infers distance by measuring the time differences. Some hybrid approaches of TOA, AOA, and TDOA have been proposed [5].

Another class of techniques measures the received signal strength indication (RSSI). These techniques exploit the decaying model of electronic-magnetic field to translate RSSI to the corresponding distance [6] [7] [9]. Also, the frequency bands used for transmission vary. For example, the well-known RADAR system [8] uses the radio frequency (RF). LADAR and SONAR use the visible light and the audible sound bands respectively. LADAR, SONAR, for instances, analyze the signal reflected from the object to estimate location. A recent innovation, Cricket [10], takes a hybrid approach, using both the RF and ultrasonic bands. But, the propagation characteristic is irregular under real environment [15]. Localization systems using RSSI information suffer from these problems and usually give meter-level accuracy.

Range-free methods. They are not based the range estimation between anchor nodes to localize targets. For example, APIT [14] estimates location of targets based on the connectivity information to anchor node with known location. The more anchor nodes are deployed, the narrow area that this technique could locate. In other words, the accuracy highly depends on the deployment density of anchor nodes. There is a class of techniques that detect the sequence of some artificially generated events from an event scheduler. For example, Spotlight [13] and Lighthouse [12] correlate the event detection time of a sensor node with the known spatiotemporal relationship. Then, the detection events can be mapped into a possible position. However, it is relatively difficult to generate and disseminate these events to a large-scale area, especially considering calibration efforts.

Chapter 8

Conclusion and Future Work

In this paper, we have designed, implemented, and evaluated an adaptive RIP method that can enhance the positional accuracy of the static RIP method [1]. The adaptive RIP algorithm is based on the Estimation Error Model that can accurately predict the positional error of RIP method, given a specific SPC selection and the approximate location of a target. We have analytically derived the Estimation Error Model, which was then verified with its correctness through real experimental results. Furthermore, we have built upon this Estimation Error Model to devise an adaptive SPC (sender-pair combination) selection algorithm that dynamically finds the most optimal SPC according to the changing locations of targets relative to anchor nodes. Our experimental results show that the adaptive RIP method outperforms the static RIP method in both single-target and multi-target tracking, improving the average positional error by 47%~60% and at the 90% percentile error by 55%~61%.

For our future work, we would like to pursue several directions. One direction is to better estimate q_{error}, because using an average value as q_{error}. in the Estimation Error Model is not so accurate. One possible improvement is to find a way to model the distribution of q_{error} at a finer-grained level, as well as further analyzing factors causing q_{error}, such as multi-path fading (environmental factor), receiving power of interference signal (deployment factors), etc. Considering these factors can help improving the prediction accuracy of our Estimation Error Model.

Bibliography

[1] M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, A. Nadas, K. Molnar, S. Dora, and A. Ledeczi, "Radio-interferometric geolocation," Proc. 3rd Int'l Conference Conf. Embedded Networked Sensor Systems (SenSys 05), November 2005, pp. 1-12 .

[2] B. Kusy, M. Maroti, G. Balogh, P. Volgyesi, J. Sallai, A. Nadas, A. Ledeczi, L. Meertens, "Node Density Independent Localization," Proc. 5th Int'l Conf. Information Processing in Sensor Networks (IPSN/SPOTS 06), April 2006, pp. 441-448.

[3] B. Kusy, Gy. Balogh, A. Ledeczi, J. Sallai, M. Maroti, "inTrack: High Precision Tracking of Mobile Sensor Nodes," Proc. 4th European Workshop on Wireless Sensor Networks (EWSN 2007), January 2007, pp. 51-66.

[4] B. Kusy, J. Sallai, G. Balogh, A. Ledeczi, V. Protopopescu, J. Tolliver, F. DeNap, M. Parang, "Radio Interferometric Tracking of Mobile Wireless Nodes," Proc. 5th Int'l Conf. Mobile systems, applications and services (MobiSys 07), June 2007.

[5] L. Cong and W. Zhuang, "Hybrid TDOA/AOA mobile user location for ideband CDMA cellular systems," IEEE Tran. Wireless Communications, vol. 1, no. 3, July 2002, pp. 439-447.

[6] N. Patwari, "Relative location estimation in wireless sensor networks," IEEE Tran. Signal processing, vol. 51, no. 8, Aug. 2003, pp. 2137-2148.

[7] D. Niculescu, "Positioning in ad hoc sensor networks," IEEE Networks, vol. 18, no. 4, July 2004, pp. 24-29.

[8] P. Bahl and V. Padmanabhan, "An in building RF-based user location and tracking system," Proc. Conf. Computer Communications (IEEE Infocom 00), March 2000, pp. 775-784

[9] K. Lorincz and M. Welsh, “Motetrack: A robust, decentralized approach to RF-based location tracking,” Proc. Int’l Workshop on Location- and Con-text-Awareness (LoCA 05) May 2005, pp. 63-82

[10] N. Priyantha, A. Charkraborty, and H.Balakrishnan, “The cricket location support system,” Proc. 6th Int’l Conf. Mobile Computing and Networking (MOBICOM 00), August 2000, pp. 32-43.

[11] 3rd Generation Partnership Project, *3GPP TS 05.05*, http://www.3gpp.org/.

[12] K. R¨omer, “The lighthouse location system for smart dust,” Proc. 1st Int’l Conf. Mobile systems, applications and services (MobiSys 03), May 2003, pp. 15-30.

[13] R. Stoleru, T. He, J. a. Stankovic, and D. Luebke, “A high-accuracy, lowcost localization system for wireless sensor networks,” Proc. 3rd Int’l Conference Conf. Embedded Networked Sensor Systems (SenSys 05), November 2005, pp. 13-26 .

[14] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-Free Localization Schemes in Large-Scale Sensor Networks,”. Proc. 9th annual Int’l Conf. Mobile computing and networking (MOBICOM 03), September 2003, pp. 81-95.

[15] G. Zhou, T. He, and J. A. Stankovic, “Impact of Radio Irregularity on Wireless Sensor Networks,” Proc. 2rd Int’l Conf. Mobile systems, applications and services (MobiSys 04), June 2004, pp. 125-138.

[16] L. Girod, M. Lukac, V. Trifa, and D. Estrin, "The design and implementation of a self-calibrating acoustic sensing platform," Proc. 4th Int'l Conference Conf. Embedded Net-worked Sensor Systems (SenSys 06), October 2006, pp.71-84.

[17] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors," Proc. 7th annual Int'l Conf. Mobile computing and net-working (MOBICOM 01), July 2001, pp. 166-179.

[18] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, R. J. O'Dea, “Relative location estimation in wireless sensor networks,” IEEE Tran. Signal Process, Special Issue on Signal Processing in Networking, vol. 51, no. 9, Aug. 2003, pp. 2137-2148.

[19] N. Dragos, and B. Nath, “Ad hoc positioning system (APS) using AoA,” Proc. Conf. Computer Communications (IEEE Infocom 03), April 2003, pp.1734-1743.

[20] RIPS, http://tinyos.cvs.sourceforge.net/tinyos/tinyos-1.x/contrib/vu/apps/RipsOneHop/