

Article

# Tag Localization with Asynchronous Inertial-Based Shifting and Trilateration

Abdallah Y. Alma'aitah <sup>1</sup>, Lobna M. Eslim <sup>2</sup> and Hossam S. Hassanein <sup>2,\*</sup>

<sup>1</sup> Department of Network Engineering and Security, Jordan University of Science and Technology, Irbid 22110, Jordan; ayalmaaitah@just.edu.jo

<sup>2</sup> School of Computing, Queen's University, Kingston, ON K7L 3N6, Canada; lobna.eslim@cs.queensu.ca

\* Correspondence: hossam@cs.queensu.ca; Tel.: +1-613-533-6052

Received: 15 October 2019; Accepted: 25 November 2019; Published: 27 November 2019



**Abstract:** Personal Area Networks (PAN) are key topologies in pervasive Internet of Things (IoT) localization applications. In the numerous object localization techniques, centralization and synchronization between the elements are assumed. In this paper, we leverage crowdsourcing from multiple fixed and mobile elements to enhance object localization. A cooperative crowdsourcing scheme is proposed to localize mobile low power tags using distributed and mobile/fixed readers for GPS assisted environments (i.e., outdoor) and fixed readers for indoors. We propose Inertial-Based Shifting and Trilateration (IBST) technique to provide an accurate reckoning of the absolute location of mobile tags. The novelty in our technique is its capability to estimate tag locations even when the tag is not covered by three readers to perform trilateration. In addition, IBST provides scalability since no processing is required by the low power tags. IBST technique is validated through extensive simulations using MATLAB. Simulation results show that IBST consistently estimates location, while other indoor localization solutions fail to provide such estimates as the state-of-the-art techniques require localization data to be available simultaneously to provide location estimation.

**Keywords:** inertial sensor; localization; RSSI; crowdsourcing

## 1. Introduction

Radio Frequency Identification (RFID) and IEEE 802.15-based tags are widely accepted to be the de facto of low power identification and computation in wireless communication. These tags, battery assisted ones in particular, can be equipped with sensing elements and memories to provide rich information for services requiring location estimation. Therefore, Personal Area Network (PAN) technologies are considered to be a key enabler of many Internet of Things (IoT) smart applications. Smart applications rely on contextual information, predominantly the location of the tags or “things” at a specific time [1]. Location awareness of the tags will aid contextual decision-making, hence enhancing the quality of experience for users. Nevertheless, localizing the tags, while considering the IoT characteristics in terms of scalability, heterogeneity, and mobility, remains a challenging problem.

During the last decade, PAN technology has developed rapidly in conjunction with a steep cost reduction of the supporting system-on-chip integrated circuits [2]. Due to this development, several low-power, small footprint, and sensor-assisted devices have sprouted. Consequently, myriad of pervasive and context-aware applications have been proposed [3–11]. In particular, in RFID systems, tag- and reader-based localization schemes were proposed. In tag-based localization, the tag location is mainly estimated based on its proximity to a given reader(s) with a known location [12]. In reader-based localization, reader location can be estimated similarly by its proximity to a pre-deployed set of tags [5]. Unsurprisingly, as readers' prices are orders of magnitude more than tags, RFID tag-localization is the dominant in localization solutions.

In this paper, we propose a cooperative tag-localization scheme called Inertial-Based Shifting and Trilateration (IBST) based on crowdsourcing in both indoor and outdoor environments. The system is comprised of battery-assisted tags that are attached to mobile objects, while mobile or fixed readers (with known absolute locations) will read and write corresponding information on these tags. In our scheme, the *Readers* (heterogeneous, independent, and dynamic PAN-based standardized readers) detect surrounding tags, read their current memory, and update the tags' memory with the estimated distance and current reader location information. The readers report the detected tags' information to a backend server, the server processes the data from all readers, and then it responds to users' queries about objects of interest. We remark that our approach is fundamentally different from existing tag localization techniques.

Existing localization techniques assume that the tag's location can be estimated based on simultaneous information from all surrounding readers. This assumption, however, may not always be applicable in mobile or dynamic environments. For instance, at a given time, if the current information for localization is obtained from less than three readers, location cannot be reckoned [13]. This challenge is evident as the non-uniformity of readers' locations, heterogeneity in readers' detection ranges, and mobility of both tags and readers prevent having sufficient and synchronous detection information for each tag.

The proposed asynchronous crowdsourcing localization technique, Inertial-Based Shifting and Trilateration (IBST), is the focus of this paper. Whenever sufficient detection information is not available, IBST utilizes asynchronous reader detections to localize tags that are equipped with inertial sensors. The tags are localized based on Received Signal Strength Indicator (RSSI) trilateration/proximity and inertial-based shifting. To the best of our knowledge, our approach is the first to develop a PAN object localization system utilizing reader crowdsourcing,

- utilize tags' memory to store reader detection information and location information that can be read by other passing readers, and
- use asynchronous detection information and internal inertial sensor information to enhance localization when the concurrent spatial information is not sufficient to localize a tag.

We validate the proposed system through extensive simulations using MATLAB. Results show that our approach can achieve accurate location estimation in typical IoT settings.

The remainder of this paper is organized as follows: Section 2 reviews some of the related work and motivates our proposed approach. In Section 3, we provide system models, such as system architecture, channels model, and inertial sensors displacement model. The proposed IBST crowdsourcing technique is described in Section 4. Section 5 presents the performance evaluation of the proposed system. Finally, our conclusion is given in Section 6.

## 2. Related Work and Motivation

Various PAN localization schemes have been proposed in the literature, in which, an infrastructure of RF transceivers (e.g., RFID readers, Zigbee sink nodes, etc.) are deployed to detect and collect information from surrounding tags [6–10]. In [6], the Coupled RSSI and Inertial Navigation System (INS) Localization scheme (CRIL) has been proposed, having two main advantages. CRIL can adapt effectively and quickly to dynamic communication environments, and it can account for the uncertainties in RSSI measurements, among which are varying covariance and outliers. The indoor localization approach, named iBILL [7], uses iBeacon and inertial sensors simultaneously in large open areas. Users' real-time locations are estimated by inertial sensors through an improved particle filter. The latter is used to alleviate the effect of sensing fluctuations of localization errors. In HILS [8], a more efficient Heron-bilateration-based position determination technique (HBPD) is proposed. This novel technique requires Wi-Fi signals from only two access points to localize a mobile device. HILS utilizes HBPD technique to reset the bias drift errors associated with inertial sensors and to indemnify the unavailability of strong Wi-Fi signals from Access Points (APs).

ANTspin [9] has been proposed as an efficient absolute localization method for RFID tags using Spinning Antenna. It introduces a rotary table in the experiment where the reader antennas are fixed to continuously collect dynamics data. Based on the characteristics of collected RSSI data, the relative incident angle and distance between tags and antenna are analyzed for localization. An approach for RSSI-based, calibration-free, and real-time indoor localization is proposed in [10]. Although the author presents switch-beam array-based hardware, compliant with IEEE 802.15.4 router functionality, the focus is on the creation of an algorithm layer to be used with the pre-existing hardware that is capable to enable full localization and data contextualization over a standard 802.15.4 Wireless Sensor Network using RSSI information with no need for a prolonged offline calibration phase.

In each of these five solutions, having sufficient information to localize at a given time is required. In addition, the centralized and fixed infrastructure-based systems provide limited scalability, especially in solutions that require bulky readers that cannot be mobile and may not be applicable for IoT settings, especially in outdoor environments. In the next section, we show the advantage of crowdsourcing on the adaptability of the system given a dynamic environment.

### 3. System Model

#### 3.1. System Architecture

We consider an (indoor or outdoor) environment where we track tagged mobile users (a person, animal, or object) in two-dimensional space in a specific place. A number of dynamic, heterogeneous, and uncoordinated readers in known positions are distributed and authorized to discover, access, and update the memories of tags within their vicinity. The tags are assumed to be battery-assisted and can be read through a PAN standardized protocol (e.g., IEEE 802.15.4 [14], RFID [15], WiFi [16], etc.). Tags are equipped with an ultra-low power inertial measurement unit (IMU) [17]. Thus, the object within the range of a given reader can communicate data and measure the received signal strength.

#### 3.2. Ranging Model

In our proposal, we adopt RSSI-based ranging in which the distance is estimated based on the received signal strength due to its simplicity and availability in most wireless transceivers. However, other ranging techniques (e.g., Angle of Arrival and Time difference of Arrival) may be incorporated (especially in fixed dedicated readers) if higher hardware complexity can be tolerated.

Based on the lognormal path loss model [18], the RSSI of the transmitted signal from the tag, denoted by  $P_{RX}$ , is written as

$$P_{RX}(dBm) = A - 10\eta \log_{10}\left(\frac{d}{d_0}\right) + N_0, \quad (1)$$

where  $A$  is the received signal strength at a reference distance  $d_0$ ,  $d$  is the distance between the reader and the tag,  $\eta$  is the path loss exponent, and  $N_0$  is the noise in the environment. The value of  $A$  depends on the transmitted signal power  $P_{TX}$  and the antenna gains of the transmitter and the receiver. The noise  $N_0$  is usually defined as a zero-mean Gaussian random variable  $N(0, \sigma)$ . It is worth mentioning that the antenna in the proposed system is assumed to be omnidirectional.

Rewriting Equation (1), the distance  $d$  between the reader and the tag can be written as:

$$d = d_0 \cdot 10^{\frac{A - P_{RX} + N_0}{10\eta}}. \quad (2)$$

Shadowing and multipath effects are severe on RSSI ranging at long distances due to noise factor in the power part of Equation (2). In other words, it is more accurate to estimate the distance if the power of the received signal is higher than estimating the distance at lower RSSI.

### 3.3. Inertial Sensor Model

In our scheme, we utilize inertial sensor modules in the tags to improve the localization accuracy. In the Inertial Navigation System (INS) [17], IMU is used to estimate a tag's location from a given reference point. The sensor is used to estimate the orientation (heading) from that reference point. However, the location error of the IMU usually accumulates, leaving the sole use of inertial sensors impractical.

To model the INS sensor estimated location, accumulated error per displacement within a time period  $\Delta t$  is presented based on errors in the sensor which output the degrees, the gyroscope which outputs the angular velocity of the object, and the accelerometer which outputs the linear acceleration. Accordingly, if the current two-dimensional location is  $P_i$  at time  $t$ , the next estimated location,  $P_{i+1}$ , at  $t + \Delta t$  can be given as

$$P(t + \Delta t) = P(t) + dis + dis_e, \quad (3)$$

where  $dis$  is the mean displacement vector calculated for  $\Delta t$ , and  $dis_e$  is the displacement error, including sensor biases, which is a function of angular velocity and linear acceleration errors. Figure 1 illustrates the inertial tag displacement model.

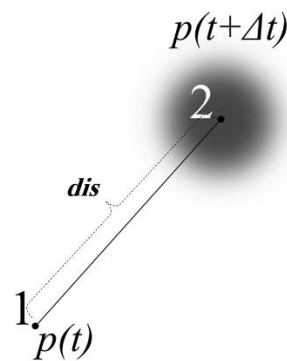


Figure 1. Illustration of the inertial tag displacement model in Equation (3).

As the inertial error reduction is beyond the scope of this paper, the error is numerically modeled based on the results reported in [19].

## 4. Proposed Solution

### 4.1. Crowdsourcing Scheme

Our approach suggests that in dynamic environments, the available crowdsourcing in terms of mobile and fixed readers, along with tags' memories, can be leveraged to provide localization service.

Given a set of  $N$  tags and a set of  $M$  readers, when a reader  $R_m$  ( $m \in \{1, 2, \dots, M\}$ ) detects a tag  $T_n$  ( $n \in \{1, 2, \dots, N\}$ ) successfully, it generates a detection event containing time and range information about the tag with respect to the reader. Each reader adds the event to a Detections table in the tag's memory, and these events are subsequently used to localize the tag.

At any tag detection, the reader creates and then updates two types of information:

- Detections table*, shown in Table 1, contains temporal and spatial information about a tag  $T_n$  with respect to reader  $R_m$ . Such information is considered the raw information about a tag's vicinity to a specific reader at a given time. The number of entries in Detections table is denoted by  $k$ .
- Absolute Tag Location (ATL) table*, shown in Table 2, contains the estimated locations of a tag  $T_n$ . Each location is identified by its estimation time.
- Tag Displacement Vector (TDV) table* contains the distance vectors that are measured based on inertial sensors (IS) records. TDV contains  $k - 1$  tag displacement vectors between every two subsequent reader detections.

**Table 1.** Schema of a Detections table.

Field	Description
time	The time at which a Reader $R_m$ detects tag $T_n$ and creates the detection record.
position	The 2D position of the Reader $R_m$ at time of detection is represented by relative $x, y$ coordinates.
distance	The tag to Reader distance, measured using RSSI.

**Table 2.** Schema of an Absolute Tag Location table.

Field	Description
time	The time at which a Reader $R_m$ estimates the location of tag $T_n$ based on the tag's detection information.
location	The estimated location of $T_n$ , is represented by $x, y$ coordinates.

*Readers* scan the surrounding vicinity to detect tags in their proximity, fetch *Detections*, *ATL*, and *TDV* tables, in addition to *IS* records. If enough detection entries are available to perform trilateration, the tags' location is estimated, and tables in the tag's memory are updated.

By this schema, tags' memories are always updated once detected by *Readers*. For each successfully identified tag  $T_n$  by a given reader  $R_m$ , the reader creates a detection record in the memory of  $T_n$ .

The tables in the tag's memory allow the reader to estimate the tag's current location. Note that the location estimation algorithm is performed by readers. Tags function is limited to inertial sensors information recording. Location information processed by the reader is then reported to a central database (online or offline), which is accessible by a user that is interested in the location of a specific tag(s).

#### 4.2. Inertial-Based Shifting Trilateration (IBST) Technique

Typically, most distance-based localization techniques assume that the measured spatial information, even those from mobile anchors, is synchronous and sufficient to localize objects. Thus, they estimate the object position based on the intersection of the given spatial information (i.e., trilateration, bounding box, etc.). In dynamic reader deployment environments, this assumption may not be applicable, which results in blind spots where no location estimation is possible. In fixed and pre-deployed reader environments, high-density reader deployment will be needed to ensure full coverage of the targeted area.

Inertial-Based Shifting Trilateration (IBST) is proposed to overcome insufficient spatial information (i.e., non-intersecting ranges from three or more readers is available). In IBST, once new ranging measurements by RSSI are available (i.e., once a tag is detected by a reader), we expand the range around the previously calculated location(s) based on the tag's *IS* records. Next, we introduce IBST shifting process and algorithm.

#### 4.3. IBST Process

In IBST, the range from the reader to a detected tag is considered a circle that is centered at the *Reader* position. The radius of such a circle is the mapped distance from the RSSI channel model and is bounded by the maximum reading range of the reader.

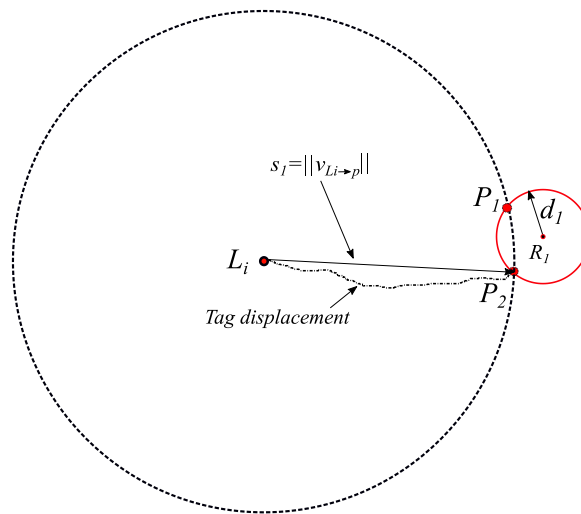
IBST makes the previously estimated location(s) centers for circles with radii based on the recorded inertial sensor readings upon current detection. Ultimately, to apply trilateration, at least two previous estimated locations in addition to the current detection range by the reader are needed.

**Definition 1.** (*detection set*): Given a set of  $K$  Readers, the detection set of a tag  $i$  is the spatial information measured simultaneously or consecutively, is denoted by  $l_k^i$ , where  $k \in \{K\}$ , and is ordered chronologically in the tag's memory.

Each element  $l_k^i$  in the *Detections* table of tag  $i$  is represented by  $\{t_k, R_k(x,y), d_k\}$ , which are defined in Table 1 as detection time, reader position, and distance (range). The tag can be localized once it has three entries in its *Detections* table.

**Theorem 1.** *If a given tag  $i$  with a known initial location  $L_i$  with a relative two-dimensional coordinates  $(x_{L_i}, y_{L_i})$  has been detected at a range of  $d_1$  from a reader  $R_1$  of a known two-dimensional coordinates  $(x_1, y_1)$ , then at most, two points ( $P_1$  and  $P_2$ ) of the tag’s absolute location can be determined.*

**Proof.** As long as the inertial sensor displacement vector  $v_{L_i \rightarrow p}$ , from the point  $L_i$  to the moment of being detected by  $R_1$ , is available, the circle centered at  $(x_{L_i}, y_{L_i})$  with a radius of  $s_1 = \|v_{L_i \rightarrow p}\|$  and the circle centered at  $(x_1, y_1)$  with a radius of  $d_1 = \|v_{R_1 \rightarrow p}\|$  will intersect. If  $v_{L_i \rightarrow p}$  is perpendicular on the circle around  $R_1$ , one intersecting point will result; otherwise, two intersecting points will result as  $\|v_{L_i \rightarrow p}\| + \|v_{R_1 \rightarrow p}\| > \|v_{L_i \rightarrow R_1}\|$ . An illustration is shown in Figure 2. □

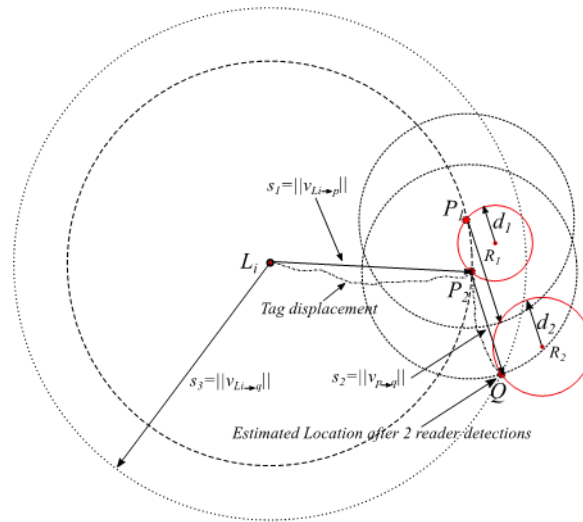


**Figure 2.** Illustration of the proof of Theorem 1 for three different displacement vectors from reader 1 to reader 2.

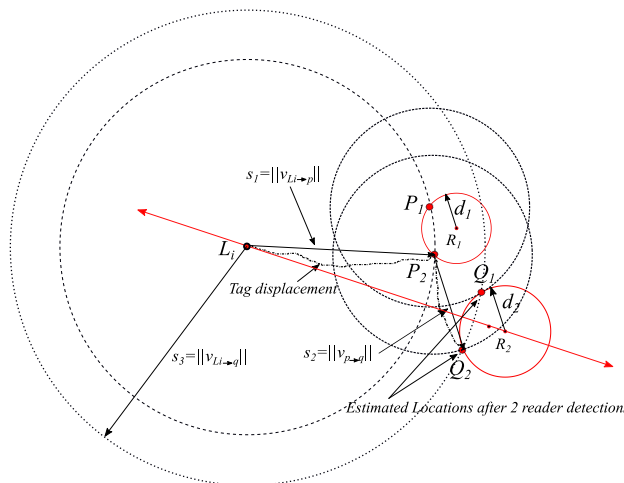
**Theorem 2.** *If a given tag  $i$  of a known initial location  $L_i$  with a relative two-dimensional coordinates  $(x_{L_i}, y_{L_i})$  has been sequentially or simultaneously detected at ranges of  $d_1$  and  $d_2$  from readers  $R_1$  and  $R_2$ , respectively (the relative two-dimensional coordinates of the readers are  $(x_1, y_1)$  and  $(x_2, y_2)$ ), then, one solution of the tag’s absolute location can be determined.*

**Proof.** As long as the inertial sensor displacement vectors  $v_{L_i \rightarrow p}$  and  $v_{p \rightarrow q}$  are available, trilateration of the circle  $C_1$  centered at  $(x_{L_i}, y_{L_i})$  with a radius of  $s_3 = \|v_{L_i \rightarrow p} + v_{p \rightarrow q}\|$ , the circle  $C_2$  centered at  $(x_1, y_1)$  with a radius of  $s_2 = \|v_{p \rightarrow q}\|$ , and the circle  $C_3$  centered at  $(x_2, y_2)$  with a radius of  $d_2 = \|v_{R_2 \rightarrow q}\|$  will intersect in a single point  $Q$ , which is a point on all three circles if the intersection points between  $C_1$  and  $C_2$  are not symmetrical around the vector passing through the initial location  $(x_{L_i}, y_{L_i})$  and current reader  $(x_2, y_2)$ . An illustration is shown in Figure 3. □

It is worth mentioning that in a case of collinearity in the shifted readers’ centers, a unique solution of the tag’s location may not be calculated and further detection(s) would be required. Figure 4 illustrates the collinearity effect on localization after two detections from the initial location.



**Figure 3.** Illustration of the proof of Theorem 2 for a tag being detected by two readers. The shifted readers 1 and 2 in dotted gray circles are non-collinear, which results in a single location solution.



**Figure 4.** Illustration of the case of collinearity between the shifted readers 1 and 2 in dotted gray circles and the third reader. This case results in two location solutions.

4.4. IBST Algorithm with Ideal Inertial-Magnetic and Range Readings

IBST is executed, as in Algorithm I, to maintain location estimation in fixed and dynamic reader settings. The input is a number of asynchronous detections and the output is the updated Absolute Tag Location (ATL) tables. In Algorithm I, inertial sensors readings are stored in the tags’ memories between subsequent detections from readers (lines 2–4). Once the tag is detected by a reader, the reader reads the Inertial Sensors (IS) records and updates the Detections table with the time of detection, reader’s location, and range between the tag and the reader (lines 5–6). If the tag has been detected previously by another reader, the current reader calculates the displacement vector from the previous reader to the current one (lines 7–9).

Once the tag with a known initial location has three entries in the Detections table, trilateration equations are applied to calculate the tag’s current location. The current location is the intersection of three circles: the two circles around latest two Absolute Tag Location (ATL) entries (with radii that are based on the shifts from inertial sensor vectors) and the circle around the current reader (with estimated range as the radius). Once the location is calculated, the oldest detection entry has no bearing since next detection will be the new third detection (line 15). If the trilateration results in one

location (unique solution), the readings are not collinear, so the location is reported to central database, and the ATL table in the tag is updated (lines 16–18). Otherwise, the three circles intersect in two points (i.e., no unique solution), and further reading(s) would be needed to calculate a unique solution (lines 19–20). If two entries are in the Detections table (including the current detection), the reader calculates the intersection(s) between the circles around latest ATL entry (with radius that is based on the shifts from inertial sensor vectors) and the circle around the current reader (lines 22–25). If less than three entries are in the Detections table, including the current reader, the vector between the current detection and the previous one is stored in the Tag Displacement Vector (TDV) table, and further detection(s) are required to perform trilateration (lines 26–27).

---

**Algorithm I.** Ideal Inertial-Based Shifting Trilateration.

---

Input: Asynchronous readers' detections, raw inertial sensor data

Output: Updated Absolute Tag Location (ATL)

```

1: initialize tag memory: empty Detections, IS, TDV, and ATL tables
2: while (tag is not detected)
3:   Do record data from inertial sensors in IS table
4: End While
5: Read Detections and IS tables // a tag is detected
6: Update Detections table with  $l_k^m$ 
7: If (Detections table has  $k > 0$  entries)
8:   Calculate displacement vector(s)  $v_{(k-1) \rightarrow k}$  based on IS table
9: End If
10: If (Detections table has  $k = 2$ ) // check for enough detections to perform trilateration
11:    $C_0 =$  Circle around  $ATL_0$  coordinates in  $l_0^m$  by a radius of  $\|v_{0 \rightarrow 1} + v_{1 \rightarrow 2}\|$ 
12:    $C_1 =$  Circle around  $ATL_1$  coordinates in  $l_1^m$  by a radius of  $\|v_{1 \rightarrow 2}\|$ 
13:    $C_2 =$  the circle around the current reader by a radius of RSSI mapped detection range
14:   Calculate the current absolute tag location by trilateration of  $C_0$ ,  $C_1$ , and  $C_2$ 
15:   Delete  $l_0^m$  and  $v_{0 \rightarrow 1}$ ; // delete first entry in Detections and TDV tables
16:   If (solution is unique)
17:     Report current ATL ( $ATL_i$ ) to a central database server
18:     Update ATL table; go to 2 // add the new ATL to previous ATL entries
19:   Else
20:     go to 2
21:   End If
22: Else (Detections table has  $k = 1$ )
23:    $C_1 =$  Circle around  $L_2$  coordinates in  $l_2^m$  by a radius of  $\|v_{1 \rightarrow 2}\|$ 
24:    $C_2 =$  the circle around the current reader by a radius of RSSI mapped detection range
25:   Calculate the current absolute tag location(s) by intersecting  $C_1$  and  $C_2$ 
26: Else
27:   Update Detections table with  $l_k^m$  and TDV table with  $v_{(k-1) \rightarrow k}$ , go to 2
28: End If

```

---

Note that Algorithm I shows how the IBST works under the assumptions of perfect estimation of range and inertial readings. Algorithm I (ideal IBST) is modified to Algorithm II (practical IBST) to account for errors in inertial sensors, in addition to RSSI ranging errors.

#### 4.5. IBST Algorithm with Incorporated Errors in Inertial and Range Readings

As mentioned in Section 3, the drift in inertial sensor readings will result in a margin of error which renders the estimated location useless. RSSI ranging, including its error margin, will provide a continuous reckoning of the estimated absolute location and will harness inertial sensor drifting error. Algorithm II below is a modified version of Algorithm I to accommodate such drifts in IS readings and RSSI ranging margins.



Similar to Algorithm I, once a tag is detected, its Detections table is updated with time, reader's location, and two ranges of information, one based on mean RSSI and the other on RSSI and one standard deviation in noise ( $\sigma_r$ ) (see line 6).

In Algorithm I, the tag location calculation was based on ideal IS readings (i.e., exactly as the actual tag trajectory); hence, trilateration will result in one unique solution as shown in Figure 3 (recall that collinearity will result in two solutions.). With drifting errors in IS, trilateration may result in two solutions (even when readers are not collinear) or no solution at all. Therefore, once a tag is detected by the first reader, the first reader calculates the intersection between the two circles with centers of  $L_i$  and  $R_1$  and radii of  $s_1$  and  $d_1$ , respectively. The resulting intersection points are at most two and both are considered possible ATLs and denoted as  $ATL_{1,1}$  and  $ATL_{1,2}$ , as illustrated in Figure 5. After the first reader detection and updating the ATL table with all possible ATLs, the tag records the inertial sensor information until the next reader detection.

Once the tag is detected by the second reader at an estimated distance of  $d_2$  from its center coordinates, the reader fetches previous entries in the Detections table, ATL table, and IS table. Based on the entries in the three tables, the reader defines three sets of circles:

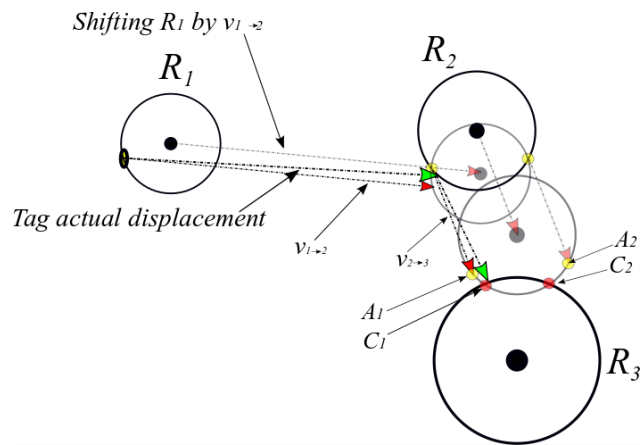
1. Circle  $C_0$  with its center at the second latest ATL(s) and radius of  $s_0 = \|v_{1 \rightarrow 2} + v_{2 \rightarrow 3}\|$ ,
2. Circle  $C_1$  with its center at latest (i.e., most recent) ATL(s) and radius of  $s_1 = \|v_{2 \rightarrow 3}\|$ , and
3. Circle  $C_2$  with its center at the reader  $R_2$  and radius of  $d_2$ .

Then  $R_2$  calculates the intersection points between circles  $C_1$  and  $C_0$ . As shown in Figure 5, the latest ATLs are  $ATL_{1,1}$  and  $ATL_{1,2}$ , and the second, latest ATL is  $ATL_0 = L_i$ . Each circle in  $C_1$  may intersect with  $C_0$  at most in two points. For instance, the circle  $C_1$  around  $ATL_{1,1}$  intersects with the circle  $C_0$  around  $ATL_0$  in two points denoted by the points  $A_1$  and  $A_2$ . Similarly,  $C_1$  around  $ATL_{1,2}$  and will intersect with  $C_0$  around  $ATL_0$  at the points  $A_3$  and  $A_4$ . As will be described shortly, these intersection points (i.e.,  $A_1 \dots A_4$ ) will be the means for selecting the next ATL(s).

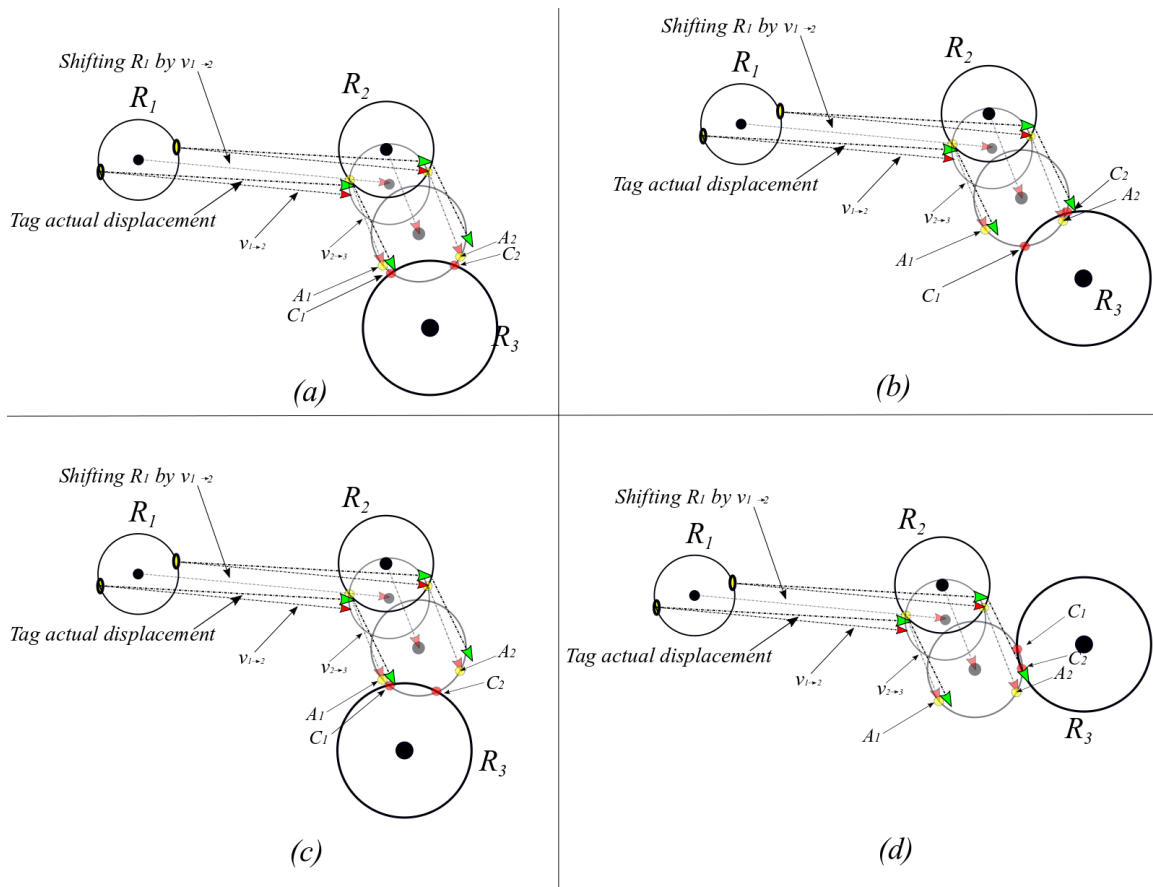
$R_2$  also calculates the intersection points between  $C_1$  and  $C_2$ . The intersection points (noted as  $B$  points) are the candidates for the current ATL (i.e.,  $ATL_2$ ). In case of two points in the latest ATL (as in the example in Figure 5 where latest ATL has  $ATL_{1,1}$  and  $L_{1,2}$ ), three possible cases may occur:

1.  $C_2$  intersects with one of the  $C_1$  circles (e.g.,  $C_{1,2}$  as shown in Figure 5). In this case,  $\|A_i - B_j\|$  will be calculated for all  $A$ 's on  $C_{1,2}$  and  $B$ 's on  $C_2$ . Then,  $B$  with  $\min(\|A_i - B_j\|)$  will be considered  $ATL_2$ .
2.  $C_2$  intersects with two  $C_1$  circles (e.g.,  $C_{1,1}$  and  $C_{1,2}$  circles shown in Figure 6). In this case,  $\|A_i - B_j\|$  will be calculated for all  $A$ 's on  $C_{1,1}$  and  $B$ 's on  $C_2$ . Then  $B$  with  $\min(\|A_i - B_j\|)$  will be considered  $ATL_{2,1}$ . Similarly,  $\|A_i - B_j\|$  will be calculated for all  $A$ 's on  $C_{1,2}$  and  $B$ 's on  $C_2$ . Then,  $B$  with  $\min(\|A_i - B_j\|)$  will be considered  $ATL_{2,2}$ . Since two points result from the intersection of such a case,  $ATL_{2,1}$  and  $ATL_{2,2}$  replace  $ATL_{1,1}$  and  $ATL_{1,2}$ .
3.  $C_2$  does not intersect with any of  $C_1$  circles as shown in Figure 7.  $ATL_2$  is considered the intersection point between  $C_2$  and the line which connects  $A$  with the minimum distance to the reader  $R_2$ .

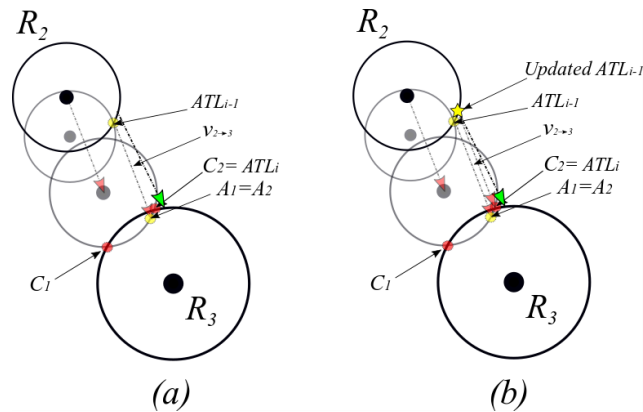
In the case of no intersection between  $C_2$  and  $C_1$  (third case shown in Figure 7), the resultant ATL will be at a longer displacement from previous ATL than  $v_{ATL_{i-1} \rightarrow A}$ . Therefore, the displacement vector in the tag is updated by  $v_{ATL_{i-1} \rightarrow ATL_i}$ . An example of the displacement vector update is depicted in Figure 8. The update will ensure that the displacement vector is on the borders of the current detecting reader  $R_2$  and the previous reader  $R_1$ .



**Figure 5.** Example of the intersection points set (C) between shifted  $R_2$  ( $R_2^{\rightarrow}$ ) and  $R_3$  in red circles. Example of shifted previous intersection points set (A) in yellow circles.



**Figure 6.** The four possible scenarios for the selection between points  $C_1$  and  $C_2$  addressed in lines 15–23 of Algorithm II.



**Figure 7.** Example of the updating process of  $ATL_{i-1}$ . (a) Selection of feasible point in C (b) based on the selected C ( $C_2$  in this example =  $ATL_i$ );  $ATL_{i-1}$  is updated by  $ATL_i - v_{2 \rightarrow 3}$  denoted by the yellow star.

---

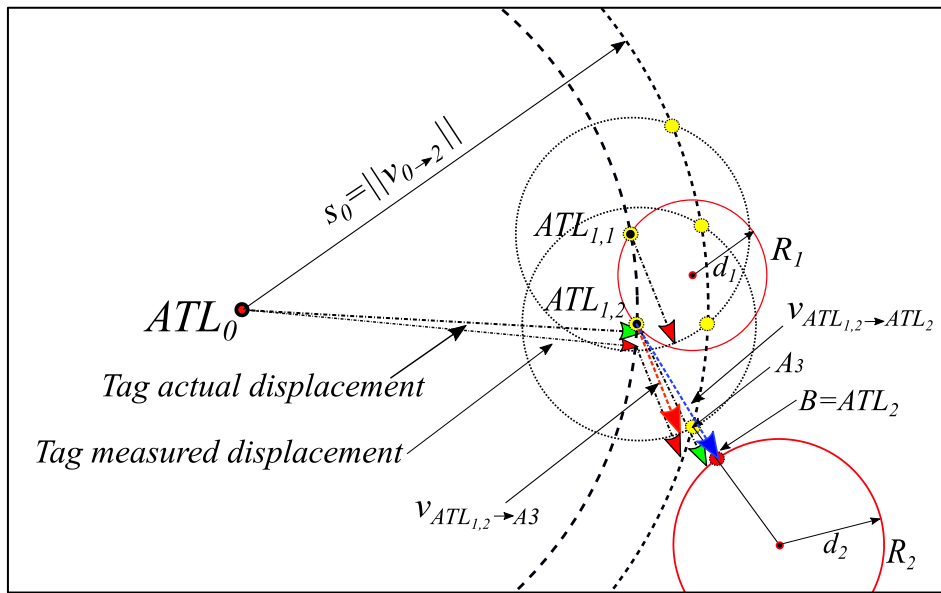
**Algorithm II.** Inertial-Based Shifting Trilateration.

---

Input: Asynchronous readers' detections, raw inertial sensor data

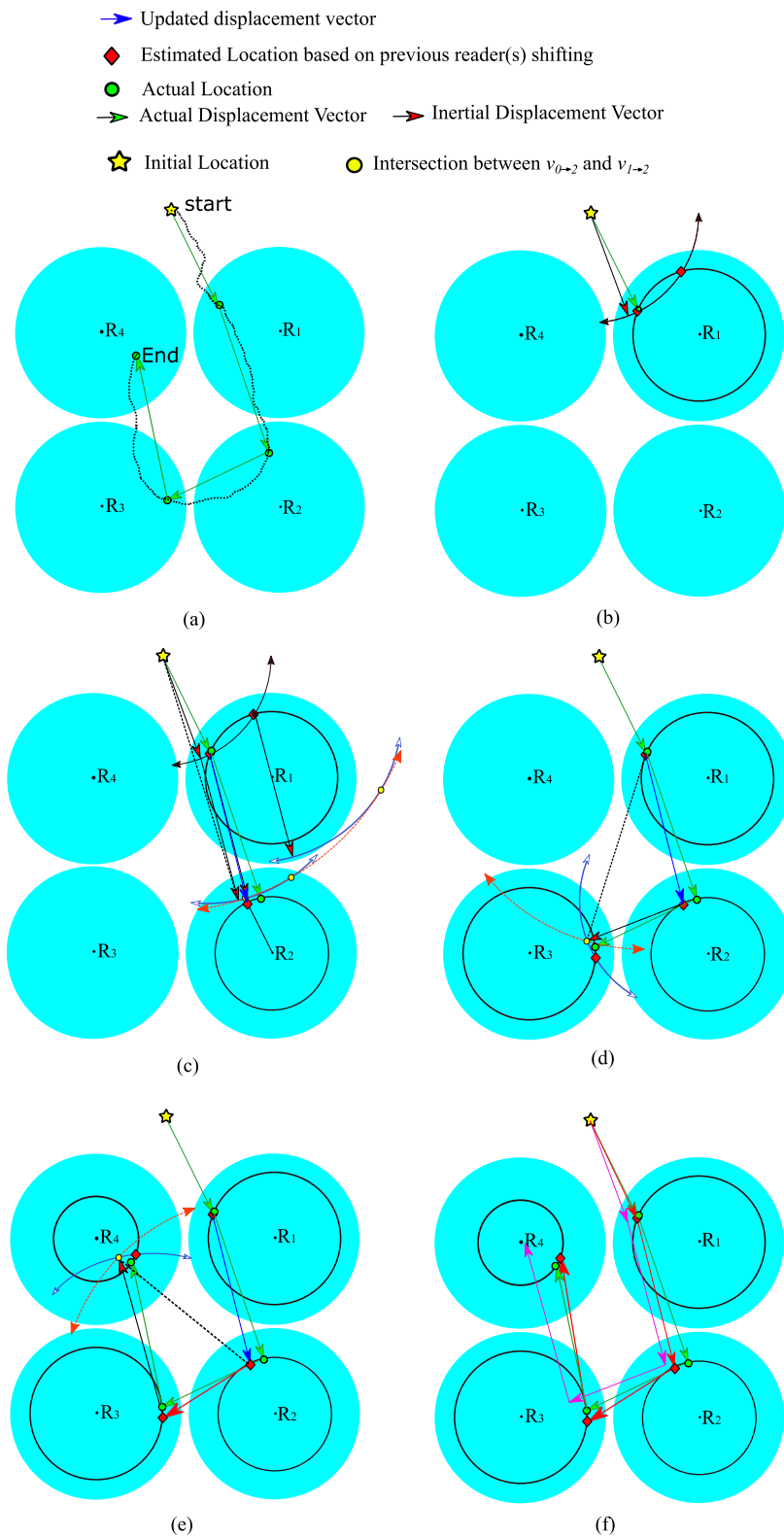
Output: Updated Absolute Tag Location (ATL)

- 1: initialize tag memory: empty Detections, IS, TDV
  - 2: while (tag is not detected)
  - 3:     Do     record data from inertial sensors in IS table
  - 4:     End While
  - 5: Read Detections, IS, and TDV tables // a tag is detected
  - 6: Update Detections table with current detection
  - 7: If (Detections table has 2 entries)
  - 8:     Calculate displacement vector(s)  $v_{0 \rightarrow 1}$  based on IS table
  - 9:      $C_0 =$  Circle around  $ATL_0$  coordinates in  $l_0^m$  by a radius of  $\|v_{0 \rightarrow 1}\|$
  - 10:      $C_1 =$  the circle around the current reader by a radius of RSSI mapped detection range
  - 11:     Calculate the set of intersection points between  $C_0$  and  $C_1$
  - 12:     All intersection points  $I$  are reported to a central server as  $ATL_{1,i}$ ,  $i \in I$
  - 13: Else if (Detections table has 3 entries) // check for enough detections to perform IBST
  - 14:     Calculate displacement vector(s)  $v_{1 \rightarrow 2}$  based on IS table
  - 15:      $C_0 =$  Circle around  $ATL_0$  coordinates in  $l_0^m$  by a radius of  $\|v_{0 \rightarrow 2}\|$
  - 16:      $C_1 =$  Circle around  $ATL_1$  coordinates in  $l_1^m$  by a radius of  $\|v_{1 \rightarrow 2}\|$
  - 17:      $C_2 =$  the circle around the current reader by a radius of RSSI mapped detection range
  - 18:     Calculate the set of intersection points  $A$ 's between  $C_0$  and  $C_1$
  - 19:     Calculate the set of intersection points  $B$ 's between  $C_1$  and  $C_2$
  - 20:     If (there are 2 intersections between  $C_1$  and  $C_2$ )
  - 21:          $ATL_{2,1} = B$ . of  $\min (\|A_1 - B_1\|, \|A_1 - B_2\|, \|A_2 - B_1\|, \|A_2 - B_2\|)$ ,
  - 22:     Else if (there are 4 intersections between  $C_1$  and  $C_2$ )
  - 23:          $ATL_{2,1} = B$ . of  $\min (\|A_1 - B_1\|, \|A_1 - B_2\|, \|A_2 - B_1\|, \|A_2 - B_2\|)$ ,
  - 24:          $ATL_{2,2} = B$ . of  $\min (\|A_3 - B_3\|, \|A_3 - B_4\|, \|A_4 - B_3\|, \|A_4 - B_4\|)$ ,
  - 25:         Update Detections table with  $l_2^m$  and TDV table with  $v_{1 \rightarrow 2}$ , go to 2
  - 26:     Else (there is no intersection between  $C_1$  and  $C_2$ )
  - 27:         Find vector from  $\vec{D}_i = A_i \rightarrow R_2$ , find  $D_i = \|A_i - R_2\|$  for  $i=1$  to 4
  - 28:          $ATL_2 = R_2 + d_2 \cdot \vec{D}_i$  of  $\min D_i$ ,
  - 29:     End if
  - 30:     Report  $ATL_i$  to a central database server
  - 31:     Update Detections table with  $l_3^m$  and TDV table with  $v_{1 \rightarrow 2}$ , go to 2
  - 32: Else
  - 33:     go to 2
  - 34: End If
-



**Figure 8.** Example of the updating process of displacement vector when  $C_2$  and  $C_1$  do not intersect. The updated vector is  $v_{ATL_{1,2} \rightarrow ATL_2}$  (in blue), and inertial sensor reading vector is  $v_{ATL_{1,2} \rightarrow A_3}$  (in red).

In Figure 9, Algorithm II is applied to a scenario in which traditional trilateration is not feasible since there is no intersection between any three readers. The tag track is set to pass through the four readers  $R_1$  to  $R_4$ , as in Figure 9a. Note that depending solely on ranging or inertial sensors will yield no answer (in RSSI ranging) or high error margins from actual tag track (in inertial sensors readings). In Figure 9b, it is assumed that the tag has a known initial location (i.e.,  $ATL_0$ ) and moves from the initial location to  $R_1$ , where the intersection points  $ATL_{1,1}$  and  $ATL_{1,2}$  (the red diamond symbol) are selected based on lines 7–12 in Algorithm II. When the tag is detected by  $R_2$  as shown in Figure 9c, three entries will be in the tag Detections table and no intersection between circles  $C_2$  and  $C_1$  (lines 26–28 in Algorithm II), resulting in  $ATL_2$  and  $ATL_{1,1}$  being replaced by  $ATL_1$ . In Figure 9d,e, there is an intersection between circles  $C_2$  and  $C_1$ , and lines 13–21 of Algorithm II are executed. The actual track and the resultant track by IBST are marked, respectively, by green and red in Figure 9f; and for comparison, Figure 9f shows the drift in inertial sensor readings (in purple) after four detections.



**Figure 9.** Inertial-Based Shifting and Trilateration (IBST) example of a tag that moves in a path which intersects the four readers  $R_1$  to  $R_4$ .

### 5. Performance Evaluation

In this section, we analyze the performance of our proposed IBST system through extensive simulations and validate the proposed scheme by comparing it with the following two previous

methods: (1) RSSI localization system, which estimates the object's position based on the trilateration of RSSI values only [20]; and (2) INS system, which the results from the inertial and magnetic sensors are used to determine tag location [19]. As will be shown, our proposed system gives precise and stable localization results, even in sparse and random reader deployment scenarios.

### 5.1. Simulation Environment and Parameter Setting

In our numerical evaluation, we consider  $N$  mobile readers in an area of 100m by 100m. The tagged object moves in two different paths within this track area. Readers are assumed to have a maximum detection range of 20m. The simulation is executed under sparse and dense deployments of random readers, with 5 and 20 readers, respectively. In addition, two paths are tested as tracks for the tag a rectangular back-and-forth track and a circular track.

The rectangular back-and-forth track starts at the point (10,20) meters and the circle track starts at (50,20) with respect to a relative Cartesian origin at the bottom left corner of the track area as in Figure 10a,b and Figure 11a,b. The starting point is assumed to be stored in the tag memory; hence, it will be known to the readers once the tag is detected. The values of  $\sigma_a$  and  $\sigma_v$  in the covariance matrix  $Q_k$  of the inertial sensors readings process noise are set to 0.1 m/s<sup>2</sup> and 0.1 m/s [19]. The log-normal shadowing path loss model is used as the signal propagation model and the value of the noise variance in RSSI ( $\sigma_{RSSI}$ ) is set to two meters with path loss exponent of 3. The selection of the above values is based on the typical settings of low cost IMUs under indoor/outdoor walking speed conditions [19]. As the main focus of this paper is to provide an estimation of the absolute tag location based on inertial sensors readings and overlapped and non-overlapped readers, providing more accurate RSSI measurements or precise inertial sensors readings is beyond the scope of this work.

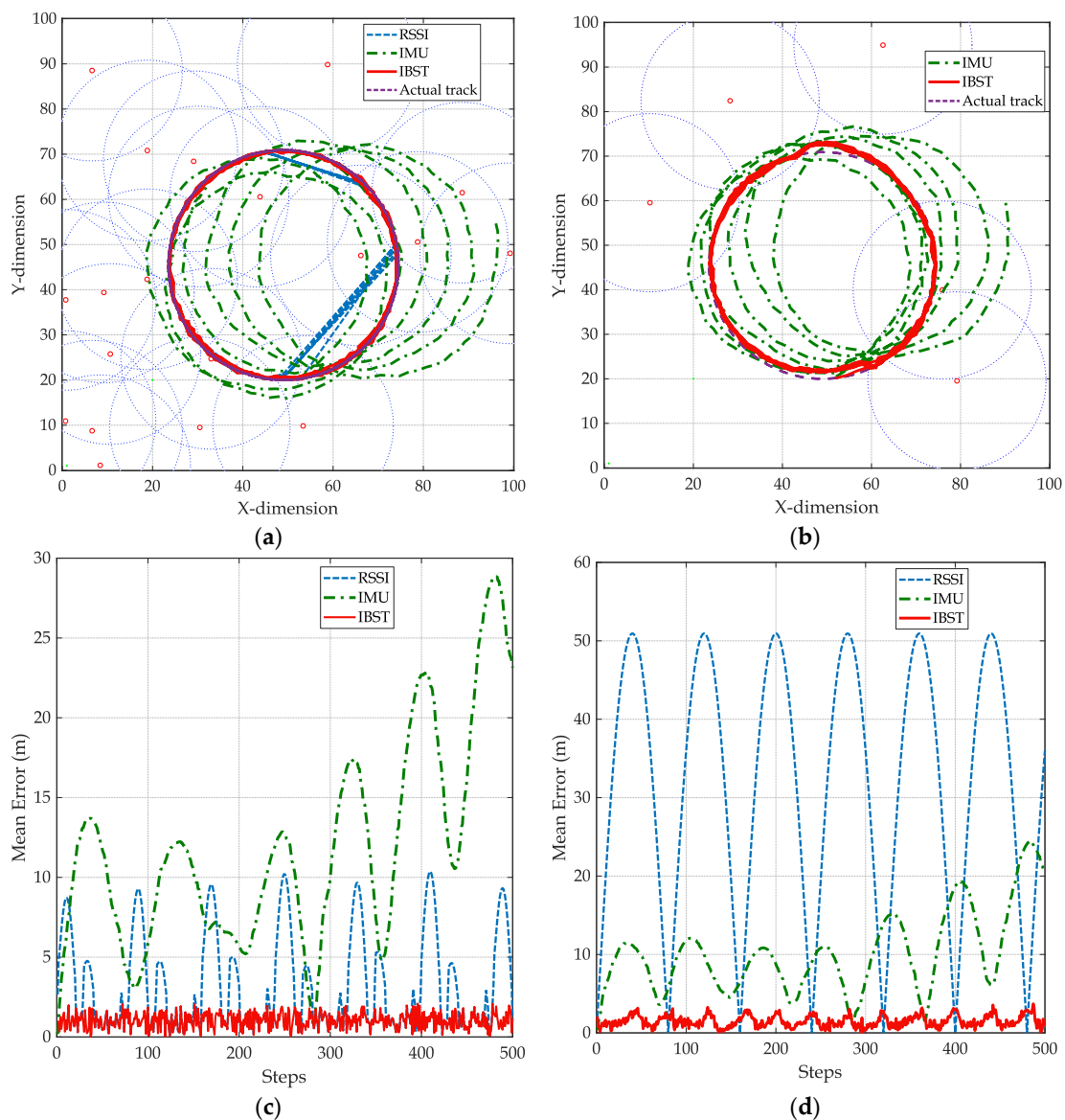
### 5.2. Simulation Results

The first scenario is the circular track of the tag with 500 steps to return to the starting x-y coordinate of (50,20). We consider a dense reader deployment of 20 readers in the track area. The readers are deployed randomly, and the tag advertises itself continuously to surrounding readers; if the tag is detected by three or more readers, RSSI location can be estimated; otherwise, RSSI trilateration outage is considered (i.e., location cannot be estimated). The tag in IBST will also advertise itself continuously to the readers, however, as in Algorithm II, the detection by one or more readers will be enough to provide a location estimation.

Figure 10a shows a scenario of dense and random deployment of 20 readers within a 100 m<sup>2</sup> localization area. The actual circular track (in purple) and the tracks based on IMU, RSSI, and IBST after 500 steps. Note that the drifting in the IMU readings is significant after two turns, and with no referencing mechanism, the drift from the actual location will continue to accumulate. RSSI is following the actual track whenever the track passes through three readers simultaneously. If an RSSI localization outage occurs, once the tag passes by three or more readers again, the line connecting between the current estimated location and the one before the outage is considered the track during the outage. IBST track, on the other hand, is continuously referenced to the estimated intersection points due to the dense coverage of the readers.

In Figure 10c, the mean error between the actual track and the track by RSSI, IMU, and IBST is plotted. IMU error is accumulating due to the drift in inertial sensor readings, which causes an incremental, yet cyclic, error as the number of steps increases. RSSI track error is non-cumulative; however, the actual track is not always passing by points where at least three readers' ranges exist. Note that when RSSI trilateration is not available, a straight line is used to connect the last available and next available locations, causing a deviation from the actual track, which in turn is represented in periodic error areas. IBST error is none-cumulative and not constrained by three readers simultaneous detection; hence, it is the lowest with continuous update to the displacement vector to follow the actual track. The average mean errors for IMU, RSSI, and IBST from the simulation scenario in Figure 10c are 11.47 m, 3.66 m, and 1.43 m, respectively.

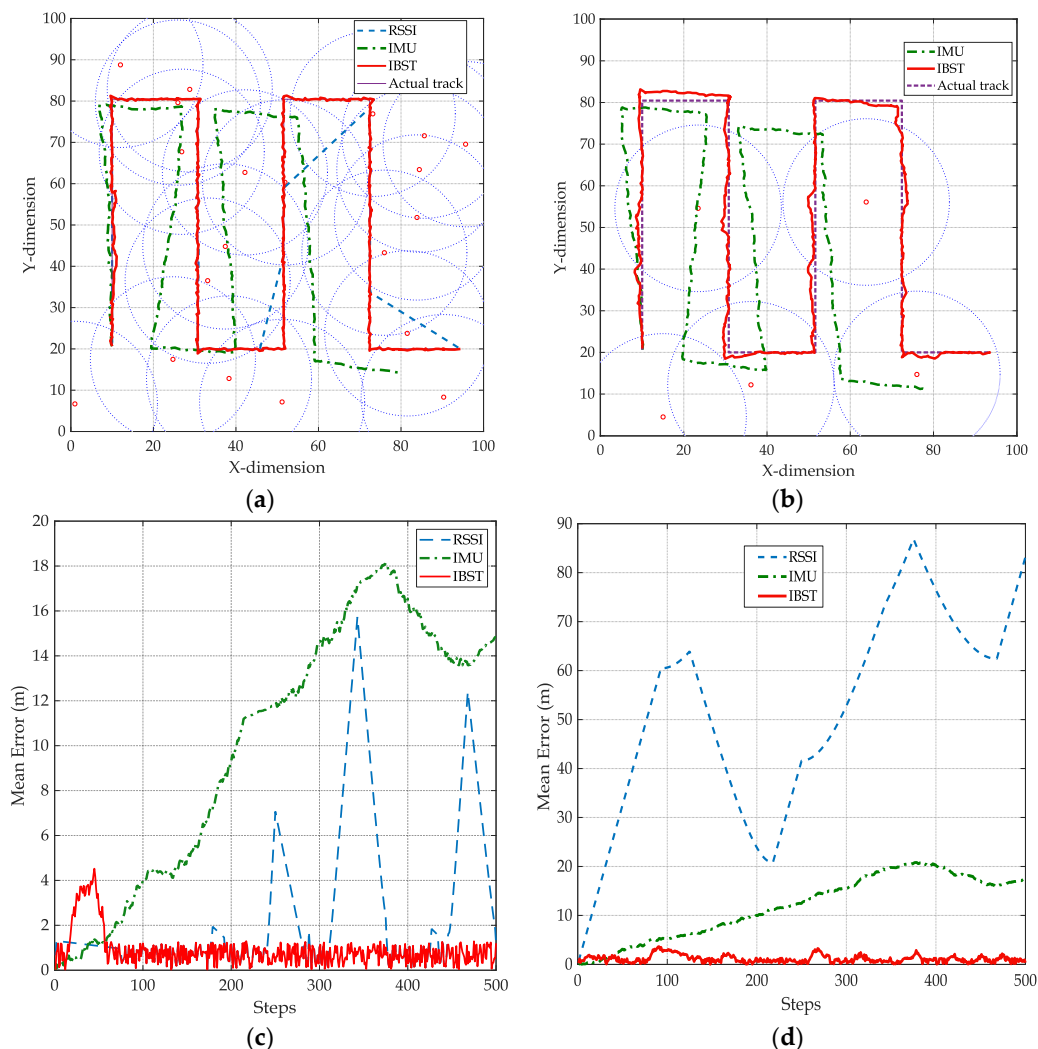
Similarly, Figure 10b shows a sparse and random deployment of 5 readers, the tracks from inertial sensors only, an actual track, and IBST. In this deployment scenario, the drifting in the IMU readings is comparable to the one in Figure 10a as IMU readings are not influenced by the number of readers. RSSI, on the other hand, had no estimated locations as the actual track does not pass by three or more readers simultaneously. As a result, no track can be estimated based on RSSI trilateration. Conversely, more deviation is observed in the IBST track from the actual track. This is due to the lower availability of readers, which results in longer inertial sensors readings before finding an intersection point to represent the estimated actual tag location. The effect of lack of any point within the test area that is covered by three readers or more on the performance of IBST is less severe than the RSSI-based track. In fact, the performance of IBST is significantly superior as shown in Figure 10d, with much lower error than both RSSI and IMU. The average mean errors for IMU, RSSI, and IBST from the simulation scenario in Figure 10d are 10.91 m, 32.84 m, and 2.34 m, respectively.



**Figure 10.** Circular track scenario with dense and sparse and random reader deployment (a) The resulting inertial measurement unit (IMU), RSSI, and IBST tracks in dense deployment. (b) The resulting IMU, RSSI, and IBST tracks in sparse deployment. (c) Location error of IMU, RSSI, and IBST from actual track in Figure 10a. (d) Location error of IMU, RSSI, and IBST from actual track in Figure 10b.

Another example of a localization scenario is depicted in Figure 11a was a dense and random deployment of 20 readers within 100 m<sup>2</sup> localization area. The track starts at the point (10,20) and is rectangular with sharp turns to emphasize the effectiveness of IBST over IMU. Note that the drifting in the IMU readings is significant after two turns. Between steps 240 and 290, outage in RSSI localization occurs, causing an increased localization error as shown in Figure 11c. The average mean errors for IMU, RSSI, and IBST from the simulation scenario in Figure 11c are 10.25 m, 5.77 m, and 2.40 m, respectively.

In Figure 11b, a scenario of sparse and random deployment of 20 readers within the localization area. In this scenario, no points on the actual track are covered by three or more readers; hence, RSSI is in an outage for all steps in the track as the estimated location is the start point (10,20). This is reflected in high localization error as shown in Figure 11d. The error of IBST track is higher than the one in Figure 11c as this scenario suffers from the lack of more than three readers to enhance the location estimation. Nevertheless, the mean error is significantly lower than IMU and RSSI. The average mean errors for IMU, RSSI, and IBST from the simulation scenario in Figure 11d are 10.47 m, 51.69 m, and 1.69 m, respectively.



**Figure 11.** Rectangular back-and-forth track scenario with dense and sparse and random reader deployment (a) the resulting IMU, RSSI, and IBST tracks in dense deployment. (b) The resulting IMU, RSSI, and IBST tracks in sparse deployment. (c) Location error of IMU, RSSI, and IBST from actual track in Figure 11a. (d) location error of IMU, RSSI, and IBST from actual track in Figure 11b.



The above circular and rectangular tracks were executed 1000 times each for:

- (a) 5, 10, 20 readers
- (b) 5, 20 m reader ranges

The mean error in meters (for 1000 runs) in addition to standard deviation of such mean are provided in Tables 3 and 4, for circular and rectangular tracks, respectively. The superiority of the IBST technique is evident. The number and range of readers affect the accuracy of both RSSI and IBST. However, RSSI is more prone to localization errors in small and large numbers of readers with high variance around the mean error. This is because of the dependency on passing through an area covered by three or more readers. IBST, on the other hand, is less dependent on the number and range of readers. The reason behind this stability of IBST mean error is that passing by a single reader will provide a location that is near the actual track. IMU is independent of the number or range of readers. However, the drifting in IMU readings without referencing causes a consistent high error in both track scenarios.

**Table 3.** Average error (in meters) after  $10^3$  runs of circular track.

Reader Range ↓	Readers Number ↓	Location Estimation Method →	RSSI	IMU	IBST
5 m	5	Mean Error	31.9257	11.3720	5.5248
		$\sigma_{error}$	0.0000	1.5261	2.6642
	10	Mean Error	31.9257	11.2789	3.0226
		$\sigma_{error}$	0.0000	1.6527	1.9542
	20	Mean Error	31.8937	11.1036	2.2547
		$\sigma_{error}$	0.2023	1.1713	1.0244
20 m	5	Mean Error	31.3701	10.8083	2.3412
		$\sigma_{error}$	1.4106	1.5968	0.7357
	10	Mean Error	25.2819	11.3305	1.8249
		$\sigma_{error}$	5.7351	1.4866	0.5001
	20	Mean Error	6.4204	11.6785	1.3353
		$\sigma_{error}$	4.2260	2.1599	0.2438

**Table 4.** Average error (in meters) after  $10^3$  runs of rectangular track.

Reader Range ↓	Readers Number ↓	Location Estimation Method →	RSSI	IMU	IBST
5 m	5	Mean Error	51.7140	10.2056	6.3736
		$\sigma_{error}$	0.0002	1.5320	2.8419
	10	Mean Error	51.6840	9.6237	4.0300
		$\sigma_{error}$	0.0005	1.0847	1.1185
	20	Mean Error	49.4478	10.2517	3.5279
		$\sigma_{error}$	8.0887	1.5405	1.5634
20 m	5	Mean Error	39.5628	9.5347	2.0329
		$\sigma_{error}$	14.9362	1.6884	0.9439
	10	Mean Error	18.2600	10.0687	1.8215
		$\sigma_{error}$	4.9584	1.4449	0.4552
	20	Mean Error	5.7765	10.1530	1.6174
		$\sigma_{error}$	4.7883	1.4679	0.1680

We would like to stress here that IBST is a location estimation algorithm that combines asynchronous inertial and range readings to estimate the location; IBST is not an optimization algorithm of ranging nor inertial sensor readings. Therefore, any improvement in ranging methods can replace RSSI in this work (e.g., AoA, TDoA, hybrid RSSI-TDoA, etc.). Similarly, any enhancement in inertial sensors accuracy or drift-reduction algorithms can replace the “IMU” results. Note that any reduction in inertial sensor or ranging errors will improve the accuracy of IBST, as well.

## 6. Conclusions

Solutions proposed in the literature for of object localization require a minimum number of simultaneous range information about the object at any given time. These minimum “sufficient” readings are not always available, especially in IoT dynamic settings where providing centralized and fixed infrastructure is infeasible. In this paper, we proposed a novel cooperative tag-localization system called Inertial-Based Shifting and Trilateration (IBST). In IBST, we leverage crowdsourcing to estimate RFID tag locations in both indoor and outdoor environments, even at the absence of simultaneous reader detections to perform trilateration. IBST does not require any processing by the low power tags, maintaining the system scalability. In our proposed system, crowdsourcing fixed or mobile readers detect battery-assisted tags attached to mobile objects, read their current memory, write detection information on these tags, and report the detected tags’ information to a backend server responsible for users’ queries about objects of interest. IBST uses asynchronous detection information and internal inertial sensor information to enhance localization when the concurrent detection information is not sufficient to localize a tag. During any tag detection, IBST maintains into the tag’s memory: temporal and spatial information *w.r.t.* itself, Absolute Tag Location (ATL) identified by time of calculation, and Tag Displacement Vector (TDV) that are measured based on inertial sensors (IS) records. At insufficient spatial information, IBST shifts (expands the radius) asynchronous RSSI measures based on the tag’s IS records and the recent detection and uses those shifted measures, along with the recent detection, to localize this tag.

We evaluate IBST’s performance through extensive simulations using MATLAB, and our findings show that IBST outperforms other techniques in sparse and random reader deployment scenarios. IBST can consistently estimate location and give precise and stable localization results in the absence of synchronous detection information, typical scenarios in IoT settings. As future work, we plan to account for non-circular antenna radiation patterns at tags and readers and to consider AoA and TDOA ranging at readers. Furthermore, the incorporation of magnetic field sensors (in addition to inertial sensors) to assist in determining displacement vectors orientation is a future direction to enhance localization accuracy.

**Author Contributions:** Conceptualization, A.Y.A., L.M.E. and H.S.H.; Methodology, A.Y.A., L.M.E. and H.S.H.; Validation, A.Y.A.; Visualization, A.Y.A.; Writing—original draft, A.Y.A.; Writing—review & editing, L.M.E. and H.S.H.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Miorandi, D.; Sicari, S.; De Pellegrini, F.; Chlamtac, I. Internet of things: Vision, applications and research challenges. *Ad Hoc Netw.* **2012**, *10*, 1497–1516. [[CrossRef](#)]
2. Welbourne, E. Building the Internet of Things Using RFID: The RFID Ecosystem Experience. *IEEE Internet Comput.* **2009**, *13*, 48–55. [[CrossRef](#)]
3. Han, S.; Lim, H.; Lee, J. An Efficient Localization Scheme for a Differential-Driving Mobile Robot Based on RFID System. *IEEE Trans. Ind. Electron.* **2007**, *54*, 3362–3369. [[CrossRef](#)]
4. Wang, C.; Wu, H.; Tzeng, N.F. RFID-based 3-D positioning schemes. In Proceedings of the 26th IEEE International Conference on Computer Communications, Barcelona, Spain, 6–12 May 2007.

5. Zhu, W.; Cao, J.; Xu, Y.; Yang, L.; Kong, J. Fault-tolerant RFID reader localization based on passive RFID tags. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *25*, 2183–2191. [[CrossRef](#)]
6. Cai, S.; Liao, W.; Luo, C.; Li, M.; Huang, X.; Li, P. CRIL: An Efficient Online Adaptive Indoor Localization System. *IEEE Trans. Veh. Technol.* **2016**, *66*, 4148–4160. [[CrossRef](#)]
7. Wu, X.; Shen, R.; Fu, L.; Tian, X.; Liu, P.; Wang, X. iBILL: Using iBeacon and Inertial Sensors for Accurate Indoor Localization in Large Open Areas. *IEEE Access* **2017**, *5*, 14589–14599. [[CrossRef](#)]
8. Tiwari, S.; Jain, V.K. HILS: Hybrid indoor localisation system using Wi-Fi received signal strength and inertial sensor's measurements of smart-phone. *IET Commun.* **2019**, *13*, 1595–1606. [[CrossRef](#)]
9. Shen, L.; Zhang, Q.; Pang, J.; Xu, H.; Li, P.; Xue, D. ANTspin: Efficient Absolute Localization Method of RFID Tags via Spinning Antenna. *Sensors* **2019**, *19*, 2194. [[CrossRef](#)] [[PubMed](#)]
10. Passafiume, M.; Maddio, S.; Cidronali, A. An Improved Approach for RSSI-Based only Calibration-Free Real-Time Indoor Localization on IEEE 802.11 and 802.15.4 Wireless Networks. *Sensors* **2017**, *17*, 717. [[CrossRef](#)] [[PubMed](#)]
11. Zhang, R.; Liu, J.; Du, X.; Li, B.; Guizani, M. AOA-Based Three-Dimensional Multi-Target Localization in Industrial WSNs for LOS Conditions. *Sensors* **2018**, *18*, 2727. [[CrossRef](#)] [[PubMed](#)]
12. Junyi, Z.; Jing, S. RFID localization algorithms and applications—A review. *J. Intell. Manuf.* **2009**, *20*, 695–707.
13. Ciunzo, D.; Rossi, P.S.; Willett, P. Generalized Rao Test for Decentralized Detection of an Uncooperative Target. *IEEE Signal Process. Lett.* **2017**, *24*, 678–682. [[CrossRef](#)]
14. *IEEE Draft Standard for Low-Rate Wireless Networks Amendment for a Low Power Wide Area Network (LPWAN) Extension to the Low Energy Critical Infrastructure Monitoring (LECIM) Physical layer (PHY)*; IEEE P802.15.4w/D5; IEEE: New York, NY, USA, 2019.
15. EPC/RFID. Available online: <https://www.gs1.org/standards/epc-rfid> (accessed on 24 November 2019).
16. *IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*; IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007); IEEE: New York, NY, USA, 2012.
17. Woodman, O.J. *An Introduction to Inertial Navigation*; No. UCAMCL-TR-696; Computer Laboratory, University of Cambridge: Cambridge, UK, 2007.
18. Rappaport, T. *Wireless Communications: Principles and Practice*, 2nd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 2001.
19. Zhang, Z.-Q.; Meng, X. Use of an Inertial/Magnetic Sensor Module for Pedestrian Tracking During Normal Walking. *IEEE Trans. Instrum. Meas.* **2014**, *64*, 776–783. [[CrossRef](#)]
20. Sadowski, S.; Spachos, P. RSSI-Based Indoor Localization With the Internet of Things. *IEEE Access* **2018**, *6*, 30149–30161. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).