Mobile Sensor Navigation using Rapid RF-based Angle of Arrival Localization

Isaac Amundson, Xenofon Koutsoukos, Janos Sallai, and Akos Ledeczi
Institute for Software Integrated Systems (ISIS)
Department of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37235, USA

***Abstract*— Over the past decade, wireless sensor networks have advanced in terms of hardware design, communication protocols, resource efficiency, and other aspects. Recently, there has been growing interest in *mobile* wireless sensor networks, and several small-profile sensing devices that are able to control their own movement have already been developed. Unfortunately, resource constraints inhibit the use of traditional navigation methods, because these typically require bulky, expensive, and sophisticated sensors, substantial memory and processor allocation, and a generous power supply. Therefore, alternative navigation techniques are required. In this paper we present TripNav, a localization and navigation system that is implemented entirely on resource-constrained wireless sensor nodes. Localization is realized using radio interferometric angle of arrival estimation, in which bearings to a mobile node from a small number of infrastructure nodes are estimated based on the observed phase differences of an RF interference signal. The position of the mobile node is then determined using triangulation. A digital compass is also employed to keep the mobile node from deviating from the desired trajectory. We demonstrate using a real-world implementation that a resource-constrained mobile sensor node can accurately perform waypoint navigation with an average position error of 0.95 m.**

I. Introduction

Typically, autonomous navigation is performed by robots equipped with cameras, laser rangefinders, sonar arrays, and other sophisticated sensors for collecting range and bearing information. These sensor data are then used to compute spatial relationships such as position and proximity, which enable the robot to follow a given trajectory. However, these sensors are large, expensive, have considerable power requirements, and/or require a powerful computing platform to analyze sensor data. In recent years, mote-sized mobile sensor platforms have been developed that are unable to use traditional navigation methods because of their small size and limited resources [1], [2], [3], [4], [5]. This emerging class of mobile sensor would greatly benefit from navigation techniques geared towards resource-constrained devices.

In order to enable navigation in mobile wireless sensor networks (MWSNs), we must develop new methods for estimating position and deriving motion vectors that are rapid and accurate in spite of the limited resources available. Localization in wireless sensor networks has been studied extensively, and several techniques exist that provide sub-meter accuracy. However, these techniques are often unacceptable for mobile sensor localization due to algorithm complexity and cost. For example, although GPS receivers are available for mote-scale devices, they are still relatively expensive [6]. The Cricket location-support system requires customized hardware with ultrasonic sensors [7]. Other techniques such as RIPS do not require additional hardware support; however, localization latency is prohibitively high for mobile devices [8].

In this paper, we propose a localization and waypoint navigation system called TripNav, in which a mobile sensor node follows a path by navigating between position coordinates. Position estimates are obtained using a localization technique we developed that combines radio interferometric angle-of-arrival estimation [9] with least squares triangulation [10]. We use this approach because it provides rapid and accurate position estimates and runs on resource-constrained sensor nodes without the need for hardware modifications. These properties are desirable because they enable such a system to be assembled and deployed quickly and inexpensively using off-the-shelf components.

Way-finding represents a major category of navigational behavior [11]. Simple waypoint navigation scenarios include automated transportation routes and sentries that patrol a path along the perimeter of a secure area. For our research, the mobile sensor node is provided with a target speed and a set of waypoints, and is instructed to pass by each waypoint in the order they are given. The node is comprised of an XSM mote [12] mounted to an iRobot Create [13]. The Create is a programmable robot that hosts a small suite of sensors; however, we use it only as a mobile platform, and all localization and navigation control operations are performed on the attached mote. In addition, we employ a digital compass, which provides heading estimates, from which we can calculate the heading error of the mobile sensor with respect to the desired trajectory. A simple controller, implemented in software, is then used to derive the necessary wheel speeds for maintaining the correct heading.

In previous work [9], we developed a system for estimating the angle-of-arrival of an interference signal. The system was comprised entirely of COTS sensor nodes, it was completely distributed, bearing could be estimated rapidly, and no additional hardware was required. Our present research builds on this technique by estimating bearing to multiple anchors, and then determining position using triangulation. Because the technique is rapid, it is appropriate for mobile devices, which must continuously update their position estimates for

navigation. By implementing our angle-of-arrival technique on a mobile platform and using a simple waypoint navigation approach for determining motion vectors, we are able to satisfy the main criteria for a successful MWSN [14]. These criteria include 1) no hardware modifications, 2) sub-meter position accuracy, 3) position estimation on the order of seconds, and 4) implementation on a resource-constrained system.

The contributions of this work are:

1) We describe TripNav, a lightweight localization and waypoint navigation system for resource-constrained mobile wireless sensor networks, and demonstrate that our localization method is indeed suitable for mobile sensor navigation.
2) We perform error and timing analyses that show that location error, heading error, and latency do not significantly impact navigation.
3) Our experimental results show that TripNav works reliably and has a trajectory error of less than one meter.

The remainder of this paper is organized as follows. In Section II, we review other MWSN research that has recently appeared in the literature. In Section III, we describe the Radio Interferometric Positioning System and radio interferometric angle-of-arrival estimation, key components of our proposed navigation system. We then present the system design of our TripNav waypoint navigation method in Section IV. In Section V, we analyze the main sources of TripNav error. We describe our real-world implementation in Section VI, and evaluate the performance of the system in Section VII. Finally, in Section VIII, we conclude.

II. Related Work

To date, most mobile wireless sensor navigation applications deal with *tracking* a mobile embedded sensor (a mobile sensor that does not control its own movement) [15], [16], [17]. Tracking is the process of taking a series of measurements, and using that information to determine the history, current position, and potential future positions of the object. Tracking can be cooperative (i.e., the tracked object participates in its localization), or non-cooperative. Mobile *actuated* sensors, on the other hand, control their own movement. Navigation requires in-the-loop processing of location data to determine a motion vector that will keep the mobile entity on the desired trajectory. There are two main approaches for using mobile sensors for navigation: dead reckoning and reference-based [18].

Dead reckoning uses onboard sensors to determine the distance traveled over a designated time interval. Distance can be obtained using odometry via encoders, or by inertial navigation techniques using accelerometers and gyroscopes. The main benefit of using dead reckoning systems is that no external infrastructure is required. Position can be inferred by integrating velocity, or doubly integrating acceleration, with respect to time; however, error will accrue unbounded unless the mobile node can periodically reset the error by using known reference positions.

In reference-based systems, mobile entities use landmarks in the region for correct positioning and orientation. Landmarks can be active beacons, such as sensor nodes and satellites, or physical structures, such as mountains and buildings. A common use for reference-based systems is model-matching, also referred to as mapping [18]. Mapping requires the ability to detect landmarks in the environment, and match them to a representation of the environment that was obtained a priori and stored in the memory of the mobile device. For mobile robots, landmarks are typically detected using cameras. Landmarks do not need to be structural, however. Received signal strength (RSS) profiling is a type of model-matching technique, in which the observed signal strengths from multiple wireless access points are used to estimate position [19], [20]. Simultaneous localization and mapping (SLAM) is also a type of mapping, in which the mobile device builds a map of the environment at the same time as it determines its position [21]. Similarly, simultaneous localization and tracking (SLAT) is a technique to localize a mobile entity while keeping track of the path it has taken to arrive at its present position [22].

Most reference-based navigation techniques are difficult to implement on resource-constrained mobile sensors without increasing cost or modifying hardware. For example, in [4], position is determined using an overhead camera system. In many instances, the cost of the camera system alone can be higher than the rest of the sensor network, making this localization approach undesirable. Millibots use a combination of dead-reckoning and ultrasonic ranging [5]. Supporting ranging in this manner requires customized hardware with ultrasonic sensors, a feature typically not found on COTS sensor nodes. Another technique uses static sensors to guide the mobile sensor to a specific area [3], [23]; however, this approach can only achieve course-grained accuracy.

III. Background

A. The Radio Interferometric Positioning System

Our work is based on the Radio Interferometric Positioning System (RIPS), an RF-based localization method presented in [8]. RIPS was developed as a means for accurately determining the relative positions of sensor nodes over a wide area by only using the onboard radio hardware. It was originally implemented on the COTS Mica2 mote platform [24], which has a 7.4 MHz processor, 4 kB RAM, and a CC1000 tunable radio transceiver that operates in the 433 MHz range [25]. Although the radio hardware is quite versatile for its size and cost, 433 MHz is too high to analyze the received signal directly. Instead, RIPS employs transmitter pairs at close frequencies for generating an interference signal. The phase and frequency of the resulting envelope signal can be measured by making successive reads of the received signal strength indicator (RSSI).

The approach works as follows. Two nodes, A and B, transmit pure sinusoids at respective frequencies f_A and f_B, such that $f_B < f_A$. The two signals interfere, resulting in a beat signal with frequency $|f_A - f_B|$. The phase difference between receiver pairs is a linear combination of the distances between the four participating nodes:

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

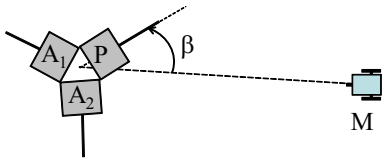

Fig. 1. Array containing a primary node (P) and two assistant nodes (A_1, A_2). A target node (M) computes its bearing (β) from the array.

where $\Delta\varphi$ is the phase difference, λ is the wavelength of the transmitted signal, and d_{AD}, d_{BD}, d_{BC}, and d_{AC} are the respective distances between node pairs (A, D), (B, D), (B, C), and (A, C).

The distance measurement $(d_{AD} - d_{BD} + d_{BC} - d_{AC})$ is referred to as a *quad-range*. Because phase wraps to 0 at 2π, an ambiguity exists where an observed signal phase difference could correspond to several different quad-ranges. To resolve this, RIPS samples at multiple frequencies and searches for a unique quad-range that satisfies Equation (1) for each measured phase difference and corresponding wavelength.

A single quad-range is not sufficient to determine the positions of the four nodes involved in the radio interferometric measurement. Instead, a genetic optimization algorithm is used that takes into consideration all participating nodes in the sensing region. The algorithm is able to simultaneously remove bad measurements while accurately estimating the position of the sensors. The quad-ranges between a sufficient number of participating nodes constrain each node to a unique position in the sensing region. RIPS was shown to have an accuracy of 3 cm at a range of up to 160 meters; however, it could take up to several minutes in large networks, and thus is not suitable for localizing mobile nodes.

B. Radio Interferometric Angle of Arrival Estimation

In [9], we presented a rapid technique for determining bearing to a target node at an unknown position from a stationary anchor node. The technique uses the same radio interferometric method as RIPS, but takes less than a second to complete.

The system consists of stationary antenna arrays and cooperating target (possible mobile) wireless sensor nodes. The array contains three nodes, a primary (P) and two assistants (A_1, A_2), as shown in Figure 1. At a predetermined time, the primary, P, and one of the assistants, A_1, transmit a pure sinusoidal signal at slightly different frequencies, which interfere to create a low-frequency beat signal whose phase is measured by the other assistant in the array, A_2, and a target node, M, at an unknown position. Such a measurement is termed a radio interferometric measurement (RIM).

The difference in phase measured by receiver nodes M and A_2 is a linear combination of the distances between the transmitters and receivers, and using Equation (1), we have

$$\Delta\varphi = \frac{2\pi}{\lambda}(d_{PA_2} - d_{A_1A_2} + d_{A_1M} - d_{PM}) \; (mod\; 2\pi), \quad (2)$$

where $\Delta\varphi$ is the phase difference, λ is the wavelength of the carrier frequency, d_{PM} is the distance between the primary node and mobile node, d_{A_1M} is the distance between the assistant transmitter and the target node, and d_{PA_1}, d_{PA_2}, and $d_{A_1A_2}$ are the respective distances between all pairs of nodes in the array.

Note that the nodes in the array are equidistant from each other, and therefore $d_{PA_2} - d_{A_1A_2} = 0$. In addition, we can eliminate the modulo 2π phase ambiguity by requiring the distance between antennas in the array to be less than half the wavelength. We can therefore rearrange Equation (2) so that known values are on the right hand side.

$$d_{A_1PM} = d_{A_1M} - d_{PM} = \frac{\Delta\varphi\lambda}{2\pi} \quad (3)$$

We refer to d_{A_1PM} as a *t-range* [16]. The t-range is significant because it defines the arm of a hyperbola that intersects the position of target node M, and whose asymptote passes through the midpoint of the line $\overline{A_1P}$, connecting the primary and assistant nodes. Figure 2 illustrates such a hyperbola with foci A_1 and P. From the figure, we see that the bearing of the asymptote is $\beta = tan^{-1}(\frac{b}{a})$, where $a = \frac{d_{A_1PM}}{2}$, $b = \sqrt{c^2 - a^2}$, and $c = \frac{d_{A_1P}}{2}$. In terms of known distances, the bearing of the asymptote is defined as

$$\beta = tan^{-1}\left(\frac{\sqrt{\left(\frac{d_{A_1P}}{2}\right)^2 - \left(\frac{d_{A_1PM}}{2}\right)^2}}{\left(\frac{d_{A_1PM}}{2}\right)}\right). \quad (4)$$

In [9], we demonstrated that we can estimate β with an average accuracy of 3.2° using this technique.

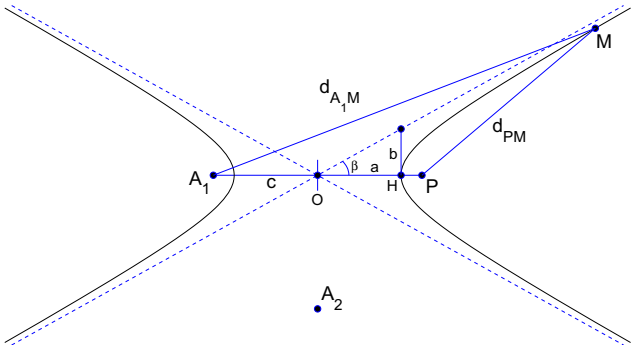

Fig. 2. The t-range defines a hyperbola that intersects target node M, and whose asymptote passes through the midpoint of the two transmitters in the array, A_1 and P.

IV. The TripNav Waypoint Navigation System

The TripNav waypoint navigation system consists of anchor nodes as described in Section III-B, and a mobile sensor that traverses a region in order to perform some task. In order for a mobile node to travel between waypoints, it is necessary to know the node's current position. By approximating the bearing of the mobile node from a sufficient number of landmarks, node position can be estimated using triangulation. Figure 3 illustrates a simple waypoint navigation scenario.

Determining spatial relationships for mobile sensor nodes is non-trivial due to the extreme resource limitations inherent in these types of devices. Designing appropriate localization

Fig. 3. Waypoint Navigation. A mobile device traverses the sensing region by navigating between position coordinates.

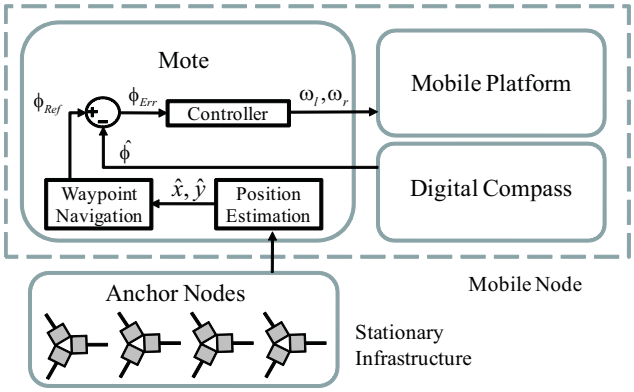

Fig. 4. Control loop for waypoint navigation.

and navigation algorithms becomes challenging in what would otherwise be a fairly straight-forward process. Consequently, we make some assumptions about the system. We assume that all participating sensor nodes have wireless antennas, and that we can use these to observe the phase of a transmitted sinusoidal interference signal. We also assume that the mobile platform is equipped with a digital compass or similar device from which it can estimate its current orientation. Finally, we assume that a sufficient number of anchors are within range of the mobile node at all times. A minimum of two anchor bearings are required for triangulation; however, a greater number of bearings will result in a more accurate position estimate.

Figure 4 is a diagram of the control loop, which illustrates how the waypoint navigation system works. A mobile sensor node traverses the sensing region by moving from waypoint to waypoint. The waypoint coordinates are stored in the mote's memory. The mobile node observes the phase of interference signals transmitted sequentially by anchor nodes at known positions within the sensing region. Anchor bearings are estimated, from which the position of the mobile node, $(\widehat{x}, \widehat{y})$, is calculated using triangulation. These coordinates are then used by the waypoint navigation logic to determine if the mobile node has reached the next waypoint. If this is the case, a new heading ϕ_{Ref} is computed and a course correction is determined based on the difference between the current estimated heading (obtained by the onboard digital compass) and the new computed heading. This heading offset, ϕ_{Err}, is input into a simple controller, which appropriately updates the angular velocities of the wheels, ω_l and ω_r, in order to keep the mobile node on the correct trajectory to intercept the waypoint. This process runs continuously until the last waypoint is reached. We describe each step of this process in detail below.

A. Mobile Platform Kinematics

We use the following equations to describe the kinematic model of our two-wheeled mobile platform with differential steering [26]:

$$\dot{x} = \frac{r(\omega_r + \omega_l)}{2} cos\phi \quad (5)$$

$$\dot{y} = \frac{r(\omega_r + \omega_l)}{2} sin\phi \quad (6)$$

$$\dot{\phi} = \frac{r(\omega_r - \omega_l)}{2b} \quad (7)$$

where x and y constitute the mobile node's position, ϕ is the heading, r is the wheel radius, b is the distance between the hub center of the driving wheel and mobile platform axis of symmetry, and ω_r and ω_l are the right and left wheel angular velocities, respectively. The speed of the mobile node is the magnitude of the velocity, and in terms of angular velocity is represented as $|v| = \frac{r(\omega_r + \omega_l)}{2}$.

B. Position and Heading Estimation

In order for a mobile node to travel between arbitrary waypoints, it is necessary to know its current position and heading. Having approximated the bearing of the mobile node from a sufficient number of anchors, we can estimate its position using triangulation. Triangulation is the process of determining the position of an object by using the bearings from known reference positions. When two reference points are used (Figure 5a), the target position will be identified as the third point in a triangle of two known angles (the bearings from each reference point), and the length of one side (the distance between reference points).

The intersection of bearings can be calculated using the following equations:

$$x = x_2 + cos(\alpha_2) \frac{y_2 - y_1 - tan(\alpha_1)(x_2 - x_1)}{cos(\alpha_2)tan(\alpha_1) - sin(\alpha_2)} \quad (8)$$

$$y = y_2 + sin(\alpha_2) \frac{y_2 - y_1 - tan(\alpha_1)(x_2 - x_1)}{cos(\alpha_2)tan(\alpha_1) - sin(\alpha_2)} \quad (9)$$

where (x, y) are the coordinates of the intersecting bearings (i.e., the position estimate of the mobile node), (x_i, y_i) are the coordinates of the reference position (i.e., the anchor), and α_i is the bearing of the mobile node relative to the anchor.

When the position of the mobile node is directly between the two reference points (Figure 5b), two bearings are not

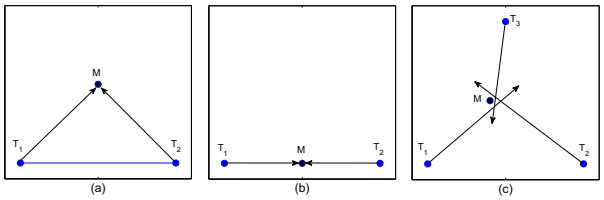

Fig. 5. Triangulation. (a) As few as two bearings from known positions are required to estimate the position of a target. (b) Degenerate case where a third bearing is needed to disambiguate position. (c) Three bearings may not intersect at the same position.

sufficient to determine position, because the mobile node could be located at any point on that axis. Therefore, a third bearing is required to disambiguate. However, three bearings may not intersect at the same point if any one bearing is inaccurate (Figure 5c). Triangulation techniques are presented in [27], [28], [29], and [30], in which position estimation using more than two bearings is considered. The method we use is a least squares orthogonal error vector solution based on [10] and [31], which is rapid, has low complexity, and still provides accurate position estimates from noisy bearing measurements.

Least squares triangulation using orthogonal error vectors works as follows. Figure 6 illustrates a simplified setup with a single anchor (T_i) and mobile node (M). The actual bearing from the anchor to the mobile node is denoted by β_i, and the estimate by $\widehat{\beta}_i$. Similarly, the vector pointing from the anchor position to the actual mobile node position is denoted by $\mathbf{v}_i$ and the vector pointing to the estimated mobile node position by $\widehat{\mathbf{v}}_i$. Finally, we denote the difference between the actual and estimated bearing vectors as the orthogonal error vector $\mathbf{e}_i$, such that ${\mathbf{e}_i}^{\mathsf{T}}\widehat{\mathbf{v}}_i = 0$.

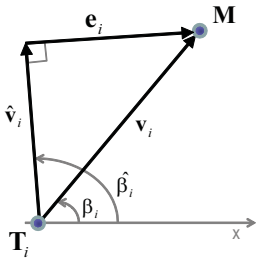

Fig. 6. Least squares triangulation using orthogonal error vectors.

If we let $\mathbf{a}_i = \begin{bmatrix} sin\widehat{\beta}_i \\ -cos\widehat{\beta}_i \end{bmatrix}$, then the orthogonal error vector is formally defined as

$$\mathbf{e}_i = \|\mathbf{M} - \mathbf{T}_i\| \, sin(\widehat{\beta}_i - \beta_i)\mathbf{a}_i,$$

where $\|\mathbf{M} - \mathbf{T}_i\|$ is the distance between the mobile node and anchor position vectors, $\widehat{\beta}_i - \beta_i$ is the Gaussian bearing noise with zero mean and variance σ_i^2, and $\mathbf{a}_i$ is the unit vector orthogonal to $\widehat{\mathbf{v}}_i$.

The position of the mobile node can be represented as $\mathbf{M} = \mathbf{T}_i + \widehat{\mathbf{v}}_i + \mathbf{e}_i$. To remove $\widehat{\mathbf{v}}_i$, we multiply with the transpose of $\mathbf{a}_i$, resulting in

$${\mathbf{a}_i}^{\mathsf{T}}\mathbf{M} = {\mathbf{a}_i}^{\mathsf{T}}\mathbf{T}_i + \eta_i,$$

where $\eta_i = \|\mathbf{M} - \mathbf{T}_i\| \, sin(\widehat{\beta}_i - \beta_i)$. Considering all anchors ($i = 1, \cdots, N$), we have a system of equations that takes the form

$$\mathbf{AM} = \mathbf{b} + \eta$$

where $\mathbf{A} = [{\mathbf{a}_1}^{\mathsf{T}}, {\mathbf{a}_2}^{\mathsf{T}}, \cdots, {\mathbf{a}_N}^{\mathsf{T}}]^{\mathsf{T}}$ and $\mathbf{b} = [{\mathbf{a}_1}^{\mathsf{T}}\mathbf{T}_1, {\mathbf{a}_2}^{\mathsf{T}}\mathbf{T}_2, \cdots, {\mathbf{a}_N}^{\mathsf{T}}\mathbf{T}_N]^{\mathsf{T}}$. A least squares solution for estimating $\mathbf{M}$ is given by

$$\widehat{\mathbf{M}} = (\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{b}.$$

where $\widehat{\mathbf{M}}$ is the position estimate returned by the triangulation using noisy bearing measurements from N anchors.

Using this method, a node can determine its position with as little as two anchors, the minimum required for triangulation. The localization algorithm outputs the estimated position of the mobile node, $\widehat{x}$ and $\widehat{y}$; however, it is unable to estimate orientation. Therefore, to obtain the heading estimate $\widehat{\phi}$, we use a digital compass attached to the mobile platform.

C. Waypoint Navigation

The mobile node needs to follow a trajectory (reference heading) that will lead it to the next waypoint. The bearing from the node's current position to the waypoint is one such trajectory. However, when the mobile node is close to the waypoint, a small localization error can contribute to large reference heading error. Instead, we define the reference heading as the bearing from the previous waypoint (or the initial position estimate of the mobile node) to the next waypoint:

$$\phi_{Ref} = \begin{cases} tan^{-1}\left(\frac{w_{y_i} - \widehat{y}}{w_{x_i} - \widehat{x}}\right) & \text{if } i = 1 \\ tan^{-1}\left(\frac{w_{y_i} - w_{y_{i-1}}}{w_{x_i} - w_{x_{i-1}}}\right) & \text{if } i > 1 \end{cases}$$

where w_{x_i} and w_{y_i} are the coordinates of waypoint i and $\widehat{x}$ and $\widehat{y}$ are the estimated position of the mobile node. Initially, ϕ_{Ref} is computed based on the position of the mobile node and the first waypoint. After the mobile node has reached the first waypoint, ϕ_{Ref} is calculated once for each waypoint i at the time waypoint $i - 1$ is reached.

Heading error is then determined by subtracting the mobile node's heading estimate, $\widehat{\phi}$, from the reference heading, ϕ_{Ref}:

$$\phi_{Err} = \phi_{Ref} - \widehat{\phi} \tag{10}$$

D. Mobile Sensor Control

To arrive at the wheel angular velocities that will keep the mobile sensor on the reference trajectory, we use a PI controller that takes the heading error ϕ_{Err} as input. Because the heading wraps to 0 at 2π, we shift the heading error to fall between $-\pi$ and π:

$$\phi_{Err} = \begin{cases} \phi_{Err} - 2\pi & \text{if } \phi_{Err} > \pi \\ \phi_{Err} + 2\pi & \text{if } \phi_{Err} < -\pi \\ \phi_{Err} & \text{otherwise} \end{cases}$$

The controller then takes the following form:

$$\dot{\phi} \quad = \quad K_p\phi_{Err}(T) + K_iT_e\sum_{t=1}^{T}\phi_{Err}(t) \tag{11}$$

where K_p and K_i are constant proportional and integral gains, respectively, T is the current sample number, $\phi_{Err}(t)$ is the heading error for sample t, and T_e is the time elapsed from the previous sample. The output of the controller, $\dot{\phi}$, is the updated angular velocity of the mobile node; however, the mobile platform is commanded by specifying an angular velocity for each wheel. Consequently, we convert $\dot{\phi}$ into individual wheel angular velocities, ω_l and ω_r, as follows:

$$\omega_l = \frac{|v| - b\dot{\phi}}{r} \quad (12)$$

$$\omega_r = \frac{|v| + b\dot{\phi}}{r} \quad (13)$$

Here, r and b (defined in Section IV-A) are system parameters with known values. $|v|$ is an input parameter to this system and does not change even though the mobile platform may not actually achieve the desired value. This is because we are only interested in regulating the heading, and not the speed, of the mobile platform.

The effect of the above transformation is that both wheels will be set with an equal desired base speed. If heading error exists, the controller will minimize it by turning one wheel faster than the base speed, and the other wheel slower, which will result in the mobile node turning in the correct direction as it moves forward. This type of controller has low run-time complexity and does not require a substantial amount of memory.

V. Error Analysis

In this section, we analyze the main sources of error in TripNav. We do this by generating a simulated setup and observing how various error sources affect the results. The simulation engine models the dynamics of the mobile node and computes the ideal bearings from each anchor at each timestep. Triangulation is then performed using the computed bearings. For the error analysis, Gaussian noise is added to the heading and position estimates, as described below for each source of error. In the simulation, we position anchors at the corners of a 20 x 20 meter region. The mobile node follows a path that takes it around a 10 x 10 meter square within the sensing region. The desired speed of the mobile node is fixed first at 100 mm/s and then at 400 mm/s. This setup is identical to our real-world experimental evaluation, described in detail in Section VII and illustrated in Figure 11.

A. Position Estimation Error

Although position error can reach as high as several meters in the worst case, it contributes relatively little to TripNav error. This is because position estimates are only used to recognize waypoint proximity. The rest of the time, the digital compass is used to maintain the desired trajectory. To analyze the effect of localization accuracy on TripNav, we simulate the system under ideal conditions, while adding Gaussian noise to the position with zero mean and varying the standard deviation between zero and five meters. Figure 7 shows the simulated paths of the mobile node with different localization accuracies. We see from the figure that even with large position error, the mobile node will still complete the circuit; however, the path it follows can be offset significantly from the desired path. Note that there are a greater number of data points for the 100 mm/s simulation because the mobile sensor is moving slower and can therefore perform more measurements.

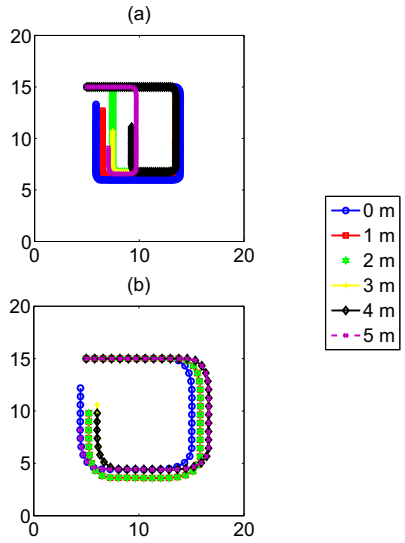

Fig. 7. TripNav trajectories due to position error when the mobile node speed is (a) 100 mm/s, and (b) 400 mm/s.

B. Digital Compass Measurement Noise

In order to compute the heading error of the mobile node, its current orientation must be known. To determine this, we use a digital compass. To understand how noisy compass sensor data affects navigation, we performed 100 simulated runs under ideal conditions, introducing a Gaussian noise to the compass heading with zero mean and a standard deviation of $0.5°$, $1°$, $2°$, $3°$, $4°$, and $5°$. Figure 8 shows the average associated position error for each. From the figure, we see that even a compass heading error as high as $5°$ does not contribute significantly to the position error.

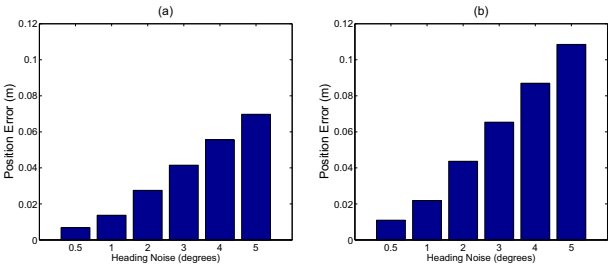

Fig. 8. TripNav average position error due to digital compass sensor noise when the mobile node speed is (a) 100 mm/s, and (b) 400 mm/s.

C. Latency

Because the mobile node is in motion while performing localization, the accuracy of the position estimate will depend

on the speed of the mobile node and the latency of the localization algorithm. Bearings from each anchor are estimated sequentially. Triangulation is then performed to determine position by finding the intersection of these bearing vectors. However, even if all other sources of error were absent from this system, these bearing vectors would still not intersect at a common point because each measurement is made from a slightly different physical location. In addition, once all measurements have been taken, the mobile node continues to change its location while phase data is being transmitted from the anchor nodes and the position estimate is computed. Therefore, the faster the TripNav control loop runs, the more accurate the position estimates will be, because the mobile node will not have had a chance to move far from the position where the localization algorithm was initiated.

To analyze how this affects the accuracy of TripNav, we simulate the system under ideal conditions while varying the number of anchors. We performed 100 simulated runs, and averaged the position error for each localization. Figure 9 shows the average position error we can expect due to latency when we use two, three, and four anchors. From the figure we can see that the latency incurred by increasing the number of anchors affects TripNav position accuracy on the order of centimeters.

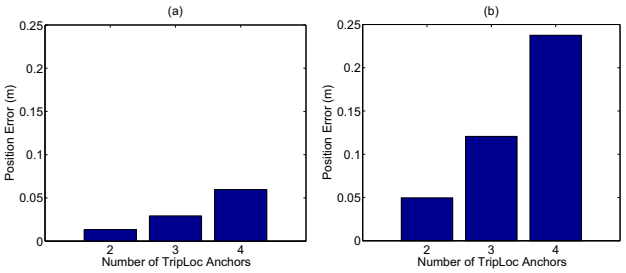

Fig. 9. TripNav average position error due to latency using 2, 3, and 4 anchors when the mobile sensor speed is (a) 100 mm/s, and (b) 400 mm/s.

VI. Implementation

Our mobile sensor is comprised of an XSM mote [12] attached to an iRobot Create mobile platform [13], as pictured in Figure 10. All localization and control operations are performed on the mote, which communicates with the Create microcontroller over a serial interface. Mobile sensor heading is determined using a Honeywell HMR3300 digital compass [32]. The Create acts solely as a mobile platform and does not perform any computation or control independently of the mote. The anchor implementation is described in [9].

The Create is a small-profile mobile platform, only 7.65 cm tall. Fixing the XSM mote to the Create body becomes problematic because the localization transmission signal is affected by ground-based reflections. We built a mount out of lightweight PVC pipe that places the mote 85 cm off the ground. We determined this height was sufficient to minimize the effect of ground-based reflections. The mount is fixed to the Create body, and houses the XSM mote, the digital compass, the connecting cable assembly for communicating with the Create, and a battery pack.

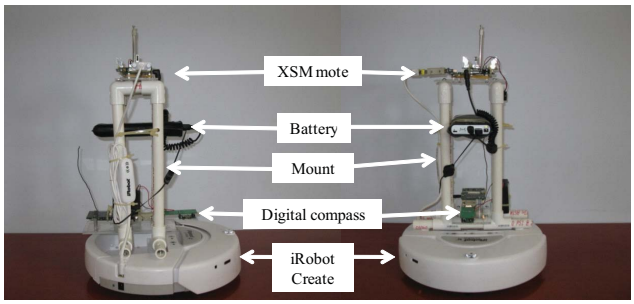

Fig. 10. The TripNav mobile platform.

One of the main implementation challenges for TripNav is designing an accurate rapid localization system, as well as waypoint navigation and mobile control logic that is small enough to fit in the memory of a single mote. Our TripNav implementation consumes approximately 3.1 kB RAM and 60 kB of programming memory.

VII. Evaluation

We place four anchors at the corners of a 20 x 20 meter region in a non-multipath outdoor environment. The mobile node is given a series of four waypoint coordinates within the region, and instructed to drive along the square route that connects the waypoints. Once the mobile node reaches the last waypoint (i.e., completes the circuit) it is instructed to come to a stop. Figure 11 illustrates this setup.

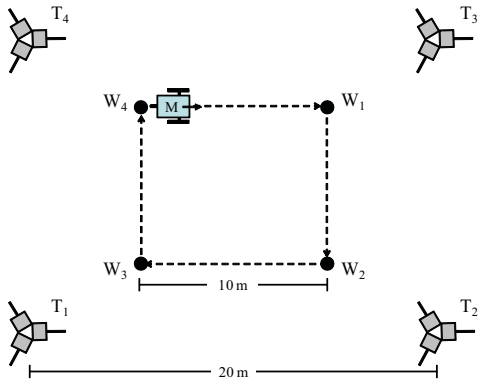

Fig. 11. Waypoint navigation experimental setup. Anchors ($T_1 \ldots T_4$) surround the sensing region. The mobile node (M) is instructed to drive in a square, passing through each waypoint ($W_1 \ldots W_4$) before proceeding to the next.

A. Performance Analysis

There are several tunable parameters for waypoint navigation using TripNav. Because TripNav only controls the heading of the mobile node and not its speed, an important system parameter is the *target drive speed* (the translational speed of the mobile node). The maximum speed of the Create is 500 mm/s. However, because we attached a sensor mount to the body of the Create, the increased weight (as well as

uneven terrain) limit the speed to about 450 mm/s. Because the controller specifies wheel speeds such that one wheel may rotate faster than the target speed and the other slower, we set our maximum target drive speed to be 400 mm/s. For our experiments, we performed waypoint navigation with target drive speeds of 100 mm/s and 400 mm/s.

Because of localization error and continuous movement, the mobile sensor will not always be able to land exactly on the waypoint. We therefore select a *waypoint range* that specifies how close the mobile sensor must be to a waypoint before being allowed to proceed to the next. The size of the waypoint range is adjusted based on the speed of the mobile node and the latency of the localization. If the mobile node's speed is slower, we can reduce the size of the waypoint range. If the mobile node's speed is faster, we must increase the size of the waypoint range, otherwise the mobile node will not realize that it reached the waypoint. Because we make the design decision to not slow down as the mobile node nears the waypoint, or stop at the waypoint, turn, then start forward motion again, we must resort to using the waypoint range. For our experiments, we ultimately chose waypoint ranges of two meters when moving at 100 mm/s and three meters when moving at 400 mm/s. We found that if we increased the waypoint range beyond these values, the mobile node still completed its circuit; however, the path it followed had a high average position error.

Finally, to filter out inaccurate position estimates, we use a simple validation gate that approximates the distance traveled since the last position estimate by multiplying the elapsed time by the average wheel speed. If the distance difference between the current and previous position estimates is greater than the estimated travel distance plus a *position error constant* (to account for positioning and drive error), then the current position estimate is discarded. We chose a value of 2.5 meters for the position error constant.

We performed five waypoint navigation runs for both target drive speeds using TripNav. Figure 12 shows the average path of the mobile node over all runs. Note that the mobile node's path does not intersect with the waypoints, and seems to stop short of the final waypoint. This is due to the waypoint range setting, where the mobile node considers the waypoint reached if it comes within the specified range. On average, position and heading accuracy with respect to the desired trajectory was 0.95 m and 4.75° when traveling at 100 mm/s and 1.08 m and 5.05° when traveling at 400 mm/s.

Figure 13 displays the outermost and innermost positions along the circuit of the mobile node over all runs. These are not individual paths, but bounds on the mobile sensor's movement over all five runs. This shows that one TripNav run does not significantly vary from another.

B. Latency Analysis

Because the mobile sensor is moving while estimating its position, localization must be performed rapidly, otherwise the mobile node will be in a significantly different location by the time a result is returned. The speed of the entire localization process depends on the latency of each component within the TripNav system, and so we provide a timing analysis of those components here. A latency analysis of the individual components involved in bearing estimation is presented in [9].

Figure 14 shows a sequence diagram for each step in the TripNav control loop, in which two anchors (dotted boxes) and a single mobile node are used. Because phase difference is used to determine bearing, each node must measure the signal phase at the same time instant. This requires synchronization with an accuracy on the order of microseconds or better. A SyncEvent message [33] is broadcast by the primary transmitter, and contains a time in the future for all participating nodes to start the first RIM. Each array then performs two RIMs, one for each primary-assistant pair. Signal transmission involves acquiring and calibrating the radio, transmitting the signal, then restoring the radio to enable data communication. The assistant nodes in the array store their phase measurements until both primary-assistant pairs have finished their RIMs, at which point they broadcast their phase measurements to the mobile node. The mobile node calculates its bearing from each array, determines its position using triangulation, obtains its heading from the digital compass, and then uses this information to move in the appropriate direction.

Table I lists the average and maximum execution times over 100 iterations for the components pictured in Figure 4. Note that TripNav execution time depends on the number of participating anchors, because bearing from each anchor is estimated sequentially. A minimum of two anchors is required for triangulation; however, the accuracy of the localization will improve with the addition of more participating anchors. We therefore provide execution times for three scenarios, in which we vary the number of participating anchors between two (the minimum required) and four (the number we use in our real-world evaluation).

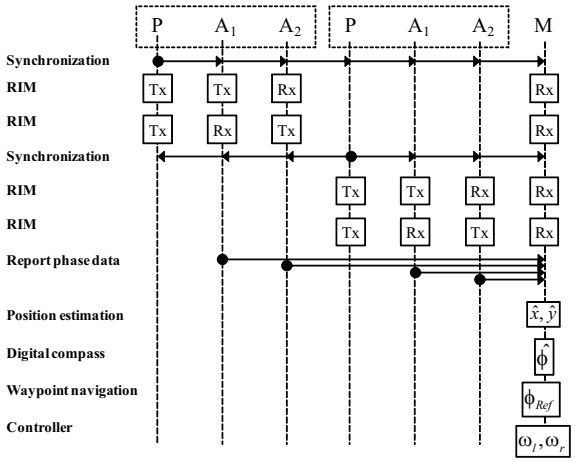

Fig. 14. Sequence diagram of the TripNav control loop in which two anchors are used.

On average, the digital compass takes approximately 50 ms to estimate heading. This is in fact a limitation of the compass hardware, which provides heading estimates at a rate of approximately 8 Hz, or 125 ms. The 50 ms latency reflects the average time we must wait for the next heading estimate to be returned.

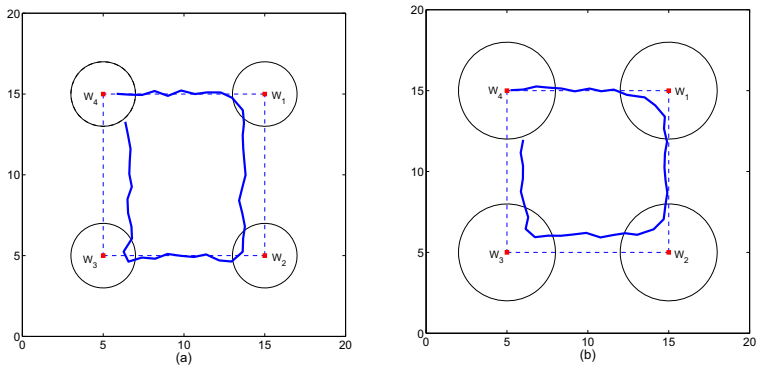

Fig. 12. Waypoint navigation average position results when mobile sensor speed is (a) 100 mm/s, and (b) 400 mm/s. The dotted line represents the desired path. Waypoints are marked $W_1 \ldots W_4$, and the the surrounding circles represent the waypoint range of (a) 2 m and (b) 3 m.

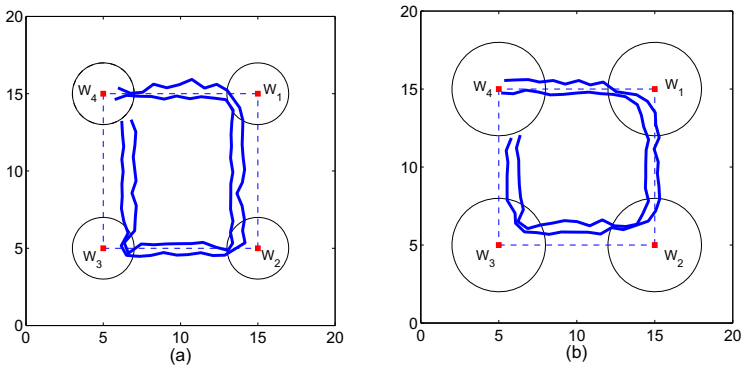

Fig. 13. Waypoint navigation outermost and innermost path when mobile sensor speed is (a) 100 mm/s, and (b) 400 mm/s. The dotted line represents the desired path. Waypoints are marked $W_1 \ldots W_4$, and the the surrounding circles represent the waypoint range of (a) 2 m and (b) 3 m.

Component	Average latency (ms)	Maximum latency (ms)
Digital compass	49.61	89.48
Waypoint Navigation	0.45	0.52
Controller	0.64	0.68
Localization (2 anchors)	888.72	956.22
Localization (3 anchors)	1283.81	1334.99
Localization (4 anchors)	1667.76	1734.48

TABLE I
LATENCY OF TRIPNAV COMPONENTS.

It is worth noting that because the mobile node acts solely as a receiver in this process, system latency is not affected by introducing more mobile nodes to the sensing region. TripNav is fully scalable in this respect; however, latency will increase as more anchors are employed, which will ultimately limit the size of the sensing region.

C. The Effect of TripNav Mobility on Position Accuracy

We performed our localization technique on a stationary sensor network deployment. Similar to the TripNav mobility experiments, four anchors were placed at the corners of a 20 x 20 meter region in an outdoor environment. Twelve stationary target nodes, placed at least 2.5 m inside the sensing region, performed 50 position estimates each. The average localization error was 0.62 m.

The experiment demonstrates the effect of TripNav mobility on the accuracy of our localization technique. When the mobile node is moving at a speed of 100 mm/s, the average position error due to mobility is 0.33 m. At a speed of 400 mm/s, the average position error is 0.46 m.

VIII. CONCLUSION

Spatio-temporal awareness in mobile wireless sensor networks entails new challenges that result from integrating resource-constrained wireless sensors onto mobile platforms. The localization methods and algorithms that provide greater accuracy on larger-footprint mobile entities with fewer resource limitations are no longer applicable. Similarly, centralized and high-latency localization techniques for static WSNs are undesirable for the majority of MWSN applications. In this paper, we presented a waypoint navigation method for resource-constrained mobile wireless sensor nodes. The method is rapid, distributed, and has sub-meter accuracy.

One of the biggest challenges we face with RF propagation is multipath fading. Currently, TripNav will not work

acceptably in multipath environments. Outdoor urban areas and building interiors are both major sources of multipath, and yet these are places where MWSNs have the greatest utility. An RF-based localization system that provides accurate results in these environments would be a major step forward. This is a future direction for our MWSN localization and navigation research, and we have already obtained encouraging preliminary results. In [34], we were able to demonstrate that precise RF indoor 1-dimensional tracking is indeed possible, and we are currently investigating how we can extend this technique to two and three dimensions. Such fine-grained RF-based localization would enable mobile sensors to navigate through hallways of burning buildings, help to evacuate shopping malls in the event of an emergency, and monitor the health of patients in every room of their house.

Acknowledgments. This work was supported by ARO MURI grant W911NF-06-1-0076, NSF grant CNS-0721604, and NSF CAREER award CNS-0347440. The authors would also like to thank Peter Volgyesi for his assistance, and Metropolitan Nashville Parks and Recreation, and Edwin Warner Park for use of their space.

REFERENCES

[1] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005, p. 55.

[2] J. Friedman, D. C. Lee, I. Tsigkogiannis, S. Wong, D. Chao, D. Levin, W. J. Kaisera, and M. B. Srivastava, "Ragobot: A new platform for wireless mobile sensor networks," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS)*, vol. LNCS 3560. Springer, 2005, p. 412.

[3] S. Bergbreiter and K. S. J. Pister, "CotsBots: An off-the-shelf platform for distributed robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003, pp. 27–31.

[4] M. B. McMickell, B. Goodwine, and L. A. Montestruque, "MICAbot: a robotic platform for large-scale distributed robotics," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2003, pp. 1600–1605.

[5] J. H.B. Brown, J. V. Weghe, C. Bererton, and P. Khosla, "Millibot trains for enhanced mobility," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 452–461, 2002.

[6] I. Getting, "The global positioning system," *IEEE Spectrum 30*, p. 3647, Dec. 1993.

[7] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 32–43, Aug. 2000.

[8] M. Maróti, B. Kusý, G. Balogh, P. Völgyesi, A. Nádas, K. Molnár, S. Dóra, and A. Lédeczi, "Radio interferometric geolocation," *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 1–12, 2005.

[9] I. Amundson, J. Sallai, X. Koutsoukos, and A. Ledeczi, "Radio interferometric angle of arrival estimation," in *Proceedings of the 7th European Conference on Wireless Sensor Networks (EWSN)*, vol. LNCS 5970. Springer, 2010, pp. 1–16.

[10] J. N. Ash and L. C. Potter, "Robust system multiangulation using subspace methods," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN)*, 2007, pp. 61–68.

[11] M. Franz and H. A. Mallot, "Biomimetic robot navigation," *Robotics and Autonomous Systems*, vol. 30, pp. 133–153, 2000.

[12] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005.

[13] iRobot, "Create programmable robot," http://www.irobot.com/.

[14] I. Amundson and X. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in *Proceedings of the 2nd International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*, R. Fuller and X. Koutsoukos, Eds., vol. LNCS 5801. Springer, 2009, pp. 235–254.

[15] B. Kusý, A. Lédeczi, and X. Koutsoukos, "Tracking mobile nodes using RF doppler shifts," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys)*, 2007, pp. 29–42.

[16] B. Kusý, J. Sallai, G. Balogh, A. Lédeczi, V. Protopopescu, J. Tolliver, F. DeNap, and M. Parang, "Radio interferometric tracking of mobile wireless nodes," *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys)*, pp. 139–151, 2007.

[17] L. Fang, P. J. Antsaklis, L. Montestruque, M. B. Mcmickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie, "Design of a wireless assisted pedestrian dead reckoning system - the NavMote experience," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2342–2358, 2005.

[18] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning sensors and techniques," *Journal of Robotic Systems*, vol. 14, no. 4, pp. 231–249, 1997.

[19] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user-location and tracking system," *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pp. 775–784, 2000.

[20] K. Lorincz and M. Welsh, "MoteTrack: a robust, decentralized approach to RF-based location tracking," *Personal and Ubiquitous Computing*, vol. 11, no. 6, pp. 489–503, 2007.

[21] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[22] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue, "Simultaneous localization, calibration, and tracking in an ad hoc sensor network," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN)*, 2006, pp. 27–33.

[23] A. Verma, H. Sawant, and J. Tan, "Selection and navigation of mobile sensor nodes using a sensor network," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PERCOM)*, 2005, pp. 41–50.

[24] UC Berkeley, "Mica2," http://www.tinyos.net/scoop/special/hardware/#mica2.

[25] Texas Instruments, "CC1000: Single chip very low power RF transceiver," http://focus.ti.com/docs/prod/folders/print/cc1000.html.

[26] I. Amundson, X. Koutsoukos, and J. Sallai, "Mobile sensor localization and navigation using RF doppler shifts," in *Proceedings of the 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*, 2008, pp. 97–102.

[27] C. McGillem and T. Rappaport, "A beacon navigation method for autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 38, no. 3, pp. 132–139, 1989.

[28] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003, pp. 1734–1743.

[29] A. Nasipuri and R. el Najjar, "Experimental evaluation of an angle based indoor localization system," in *Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2006.

[30] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 251–263, 1997.

[31] K. Doğançay, "Bearings-only target localization using total least squares," *Signal Processing*, vol. 85, no. 9, pp. 1695–1710, 2005.

[32] Honeywell, "HMR3300 digital compass," http://www.magneticsensors.com/.

[33] B. Kusý, P. Dutta, P. Levis, M. Maróti, A. Lédeczi, and D. Culler, "Elapsed time on arrival: a simple and versatile primitive for canonical time synchronization services," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 1, pp. 239–251, 2006.

[34] J. Sallai, I. Amundson, A. Ledeczi, X. Koutsoukos, and M. Maroti, "Using RF received phase for indoor tracking," in *Proceedings of the Workshop on Hot Topics in Embedded Networked Sensors (HotEmNets)*, 2010.