A Method for Estimating Angular Separation in Mobile Wireless Sensor Networks

**Isaac Amundson · Manish Kushwaha ·
Xenofon Koutsoukos**

Abstract Resource-constrained mobile sensors require periodic position measurements for navigation around the sensing region. Such information is often obtained using GPS or onboard sensors such as optical encoders. However, GPS is not reliable in all environments, and odometry accrues error over time. Although several localization techniques exist for wireless sensor networks, they are typically time consuming, resource intensive, and/or require expensive hardware, all of which are undesirable for lightweight mobile devices. In this paper, we describe a technique for determining spatial relationships that is suitable for resource-constrained mobile sensors. Angular separation between multiple pairs of stationary sensor nodes is derived using wheel encoder data in conjunction with the measured Doppler shift of an RF interference signal. Our experimental results demonstrate that using this technique, a robot is able to determine the angular separation between four pairs of sensors in a 45 × 35 m sensing region with an average error of 0.28 rad. in 0.68 s.

1 Introduction

Until recently, mobile wireless sensors had little control over their own movement, and were typically mounted on mobile objects for purposes of identification, tracking, and monitoring. This is now no longer the case; with the emergence of small-footprint mobile wireless sensors such as [10] and [14], sensor nodes are able to traverse the sensing region under their own control. This has numerous advantages, such as enabling targeted coverage [35] and connecting disjoint sensor networks [33].

Arguably one of the biggest challenges for mobile sensor nodes is *navigation*, where the mobile node must reach point *B* from point *A*. Navigation is a fairly straight-forward procedure for mobile robots equipped with resource-intensive devices such as cameras, laser rangefinders, sonar, and GPS receivers. However, when the size and available resources of the mobile device are severely limited, these kinds of sensors are either too large, heavy, expensive, or require too much power to operate over extended periods of time. Therefore, new localization and navigation methods must be developed that enable the the mobile

I. Amundson (✉) · M. Kushwaha · X. Koutsoukos
Institute for Software Integrated Systems,
Department of Electrical Engineering
and Computer Science, Vanderbilt University,
Nashville, TN, USA
e-mail: iamundson@gmail.com

device to determine where it is and where it needs to go.

For the most basic wheeled mobile robots (WMRs), navigation is typically accomplished by dead reckoning using odometry, whereby the robot monitors the angular velocity of each wheel to approximate the distance traveled over a given time period. Angular velocity is commonly measured using optical encoders mounted on each wheel. The advantage of optical encoders is that they are small and can be mounted on almost any type of WMR. When operating on a clean, level surface, optical encoders can be quite accurate. However, most environments contain dust that can interfere with the encoder readings. Additionally, odometry rapidly accrues error on uneven terrain due to wheel slippage and low tire pressure. Consequently, odometry alone is insufficient for mobile wireless sensor navigation [2].

An alternative to dead reckoning is reference-based navigation, whereby the robot determines its current position and trajectory by computing its range or bearing to landmarks within its sensing field of view. Landmarks can either be physical objects such as mountains and buildings on the horizon, or active beacons such as lighthouses and satellites. Referenced-based navigation can achieve greater accuracy than dead reckoning, especially over long distances; however, obtaining reference information in mobile wireless sensor networks is highly non-trivial. A common reference-based navigation approach is angle-based navigation [7, 8, 19, 21, 26, 28]. Angular separation between pairs of landmarks is determined using signal angle-of-arrival methods such as those presented in [9, 13, 31], and [6]. With a sufficient number of angular separations, position can be determined using triangulation [12]. However, localization is not always necessary in angle-based navigation. In [25] and [1], methods are presented for arriving at a target position by only observing the angular separation between two pairs of landmarks.

In this paper, we present a method for determining the angular separation between stationary sensor nodes that only requires the sensor radio and wheel encoders, both of which are common to robotic wireless sensors, and hence no additional hardware is required. Our method uses the Doppler shift in frequency of a radio interferometric signal and the instantaneous velocity of a mobile node transmitting a sinusoidal signal to derive the angular separation between anchor nodes surrounding the sensing region.

Our method does not require the positions of the anchor nodes, or the initial position of the mobile node, to be known. Because this method is intended for use with resource-constrained mobile sensors, it is rapid and "mote-able" (i.e., the algorithm runs entirely on the mote; no offline or PC-based processing is involved). We show using real-world experimental results and in simulation that this method is accurate with an average angular separation error of 0.28 rad.

The contributions of this work are as follows [5]:

1. We develop a method for determining the angular separation between stationary wireless sensors using resource constrained mobile devices.
2. We perform an extensive analysis that shows how each source of error contributes to the overall accuracy of the system.
3. We implement this method on a resource-constrained mobile sensor and analyze the latency and memory requirements.
4. We show using real-world experimental results and in simulation that this method is accurate with a moderate angular separation error.

The remainder of this paper is organized as follows. We describe related work in Section 2. In Section 3, we present our method for angular separation estimation, and analyze the various sources of system error in Section 4. Our implementation on a mobile wireless sensor platform is described in Section 5. Experimental results are then presented in Section 6. We conclude the paper in Section 7.

2 Related Work

2.1 Mobile Sensor Localization and Navigation

In recent years, several mote-sized mobile wireless sensing devices have been developed. These

devices are all highly resource-constrained and therefore cannot rely on traditional localization and navigation methods used by larger-scale robots. The localization trade-offs that need to be considered when designing such systems include accuracy, latency, hardware support, and compact implementation [2]. This generally implies that the mobile sensor design will need to compromise on at least one of these attributes. Typically, the project requirements will dictate how these trade-offs are made (for example, a project may require the device cost to be low, limiting hardware support, but course-grained localization accuracy will be sufficient).

The MICAbot platform was developed in order to perform large-scale distributed robotic research [29]. These devices are built on the MICA mote hardware [17], so it is fairly straight-forward to program applications and add sensor boards. The mobile node uses odometry for navigation by evenly spacing small magnets around the wheels and using a Hall-effect sensor to detect the magnets as they rotate around. This produces an average of 5.7 cm position error over a distance of 125 cm. Because dead reckoning error is unbounded, this error would continue to accrue unless periodically reset using some known reference position. The authors discuss using a ceiling-mounted camera system as a possible solution; however, the cost of the camera system alone could be higher than the rest of the sensor network, making this approach undesirable.

Millibots are modular mobile wireless sensors that can host a suite of interchangeable sensors and drive platforms [16]. Localization is accomplished using a combination of dead reckoning, ultrasound, and radio. Groups of Millibots determine their relative positions by emitting an RF pulse followed by an ultrasound pulse, and measuring the time difference of arrival of the two signals to determine range. Position is then estimated using a maximum likelihood estimation trilateration technique. Navigation is achieved by designating a subset of the mobile nodes as stationary anchors, from which the other mobile nodes use to advance. The stationary nodes then advance using a different subset of mobile nodes as anchors. Although novel, the localization technique requires specialized ultrasound transducer hardware and suffers from the relatively low range (on the order of meters) of the ultrasound signal.

The SensorFly system is a mobile aerial sensor network platform designed for the exploration and collection of situational information in indoor environments [32]. This functionality is important, especially in emergency situations, when conditions within a structure are unknown to emergency personnel. The SensorFly is lightweight, enabling three-dimensional movement, and is equipped with several sensors for spatial coordination including an accelerometer, digital compass, gyroscope, and radio. Although the system is lightweight and can operate indoors, it only achieves course-grained localization accuracy (e.g. localization resolution the size of a room).

2.2 Radio Interferometric Localization and Navigation

Typical low-cost sensor hardware supports radios that transmit in the 400 MHz–2.4 GHz range. These radios have a received signal strength indicator (RSSI) pin that can be accessed from software; however, the RSSI cannot be sampled fast enough using inexpensive mote radio hardware to determine the frequency and phase of the signal. The ability to accurately measure the frequency and phase provides a means of estimating the range or bearing to the transmitter. To enable the measurement of these signal characteristics, we can use radio interferometry, in which a second node transmits a signal at a slightly lower frequency such that the two transmitted signals interfere, creating a low-frequency beat signal (see Fig. 1). The secondary transmitter can be positioned anywhere in or near the sensing region, as long as it is stationary and its signal can reach all receiver nodes. The beat signal, which can be as low as a few hundred Hertz, is sampled by making successive reads of the RSSI. The measured phase and frequency of the beat signal at a specific time can be used to estimate range and bearing, as described in Section 3, as well as in [23, 27], and [4].

The Radio Interferometric Positioning System (RIPS) [27] was the first wireless sensor implementation to use radio interferometry for localization. It provides centimeter-accurate localization up to distances greater than 150 m, but runs on the
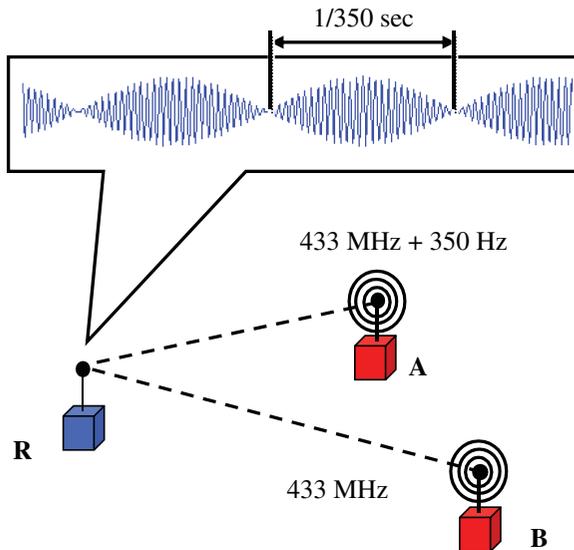
Fig. 1 Radio interferometry. Two nodes transmit sinusoidal signals at slightly different frequencies, which interfere to create a low frequency beat signal that can be measured by resource-constrained sensor nodes

order of minutes in large networks. mTrack [24], an enhancement of RIPS for tracking mobile nodes, is able to reduce this latency to tens of seconds, but still requires routing data to a base station for computation.

dTrack and dNav [22] use the Doppler shift of an RF interference signal to determine position and velocity. An extended Kalman filter is employed to handle noisy frequency measurements. Like mTrack, dTrack requires data to be routed to a base station for computation. Although dNav was implemented using only commercial-off-the-shelf (COTS) sensors, each mobile entity requires two motes each because a single mote is too memory-constrained to run the navigation application on its own.

In [6], we developed a radio interferometric technique for determining angle of arrival from several beacons. The approach is rapid, accurate, and does not require additional hardware support. We combined this technique with a mobile platform called TripNav and demonstrated that we could use such a system to perform simple waypoint navigation [4].

The main objective of our current work is to develop a mechanism for mobile wireless sensor navigation that is accurate, fast, does not require additional hardware, and can be implemented entirely on the resource-constrained mote. The above radio interferometric techniques have some of these properties, but not all. For example, RIPS is very accurate, but takes too long to run. mTrack requires data to be routed through the network for base station processing, which takes time and can suffer from single point of failure. dNav requires an extra mote in order to use an extended Kalman filter for noisy measurement data. Finally, Trip-Nav is accurate, fast, and can be implemented on COTS sensors; however, it requires three sensor nodes for each anchor placement. In contrast, our current navigation approach is extremely fast, moderately accurate, uses one COTS sensor per anchor, and can be implemented entirely on the resource-constrained node without requiring additional base station processing.

3 System Design

In this section, we describe the theory and design of our system for determining the angular separation between pairs of stationary sensor nodes. This system provides a means for angle-based navigation on mobile wireless sensors. The technique is rapid, can be implemented entirely on the resource-constrained mobile node, and does not require hardware modifications.

We consider a sensing region that contains multiple anchor nodes, as well as a mobile sensor that needs to travel from point *A* to point *B*. This scenario is illustrated in Fig. 2. In order to navigate toward point *B*, we need to know which direction to drive in, and for that we need to have some idea of the spatial relationship between the current position of the mobile node (*A*) and the goal position (*B*). Determining angular separation between pairs of nodes will provide us with such a spatial relationship. Often, angle information is determined using cameras, microphone arrays, or light pulses, all of which are not ideal for light-weight mobile sensors. We would like to estimate angular separation using only hardware that is widely available on sensor nodes. Specifically, we obtain this angle information using the sensor node radio and the optical encoders on the wheels.
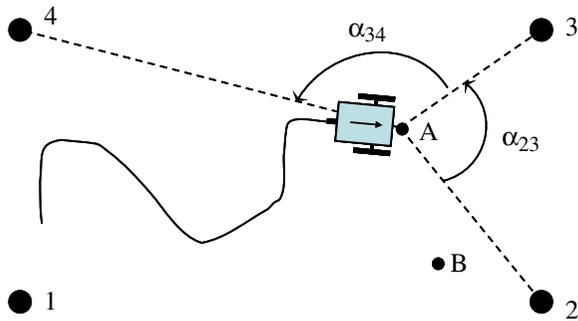
Fig. 2 A mobile sensor moves through the sensing region. The node navigates based on the angular separation between nodes (numbered 1 through 4)

3.1 Estimation of Angular Separation

The system works as follows. A mobile node, T, moving through the sensing region with velocity **v**, collects wheel angular velocity data from its encoders. For mobile platforms with 2-wheel differential steering, the relationship between the translational speed and the wheel angular velocities is

$$|v| = \frac{r(\omega_r + \omega_l)}{2} \tag{1}$$

where the speed $|v|$ is the magnitude of the velocity **v**, r is the wheel radius, and ω_r and ω_l are the right and left wheel angular velocities, respectively.

As the mobile node moves, it transmits an RF sinusoidal signal. Because the mobile node is moving with respect to the stationary receivers, the RF signal will be Doppler-shifted. The amount of Doppler shift depends on the relative speed of the mobile and anchor nodes, as well as the wavelength and carrier frequency of the signal.

However, since we cannot measure the frequency of the signal due to the inexpensive radio hardware, we use radio interferometry by placing a second stationary transmitter in the sensing region. The mobile node and secondary transmitter simultaneously transmit sinusoidal signals at slightly different frequencies such that the signals interfere on the receivers, generating a low frequency beat signal (see Fig. 1). Because the mobile node is moving through the sensing region, the resulting beat signal will also be Doppler-shifted.

The relationship between the observed Doppler-shifted frequency and the velocity of the mobile node is formalized as

$$f_i = f_{\text{beat}} - \frac{v_i}{\lambda} \tag{2}$$

where f_i is the observed Doppler-shifted frequency at receiver R_i, f_{beat} is the beat frequency of the interference signal, λ is the wavelength of the transmission, and v_i is the relative speed of mobile node M with respect to receiver R_i.

Figure 3 illustrates the geometry of a simplified setup. For now we will only consider two receiver nodes, R_i and R_j. The problem is to estimate the angular separation α_{ij} between the two receiver nodes based on the measured values of ω_r, ω_l, f_i, and f_j, and the known values f_{beat} and λ.

The relative speed, v_i, between the mobile node and receiver R_i is the scalar value resulting from the projection of **v** onto the position vector $\overrightarrow{MR_i}$, as

$$v_i = |v|\cos\beta_i \tag{3}$$

where the speed of the mobile node, $|v|$, has a negative sign if M is moving toward R_i and positive sign otherwise, and β_i is the angle between the velocity vector **v** and the position vector $\overrightarrow{MR_i}$.
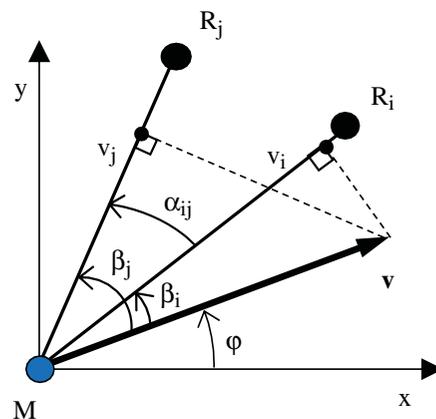
Fig. 3 Geometry of simplified setup for determining angular separation between two receivers

The relative speed is related to the received Doppler-shifted signal. By rearranging Eq. 2, we have

$$v_i = \lambda(f_{\text{beat}} - f_i). \qquad (4)$$

Combining Eqs. 3 and 4, and rearranging, we can calculate β_i by

$$\beta_i = \cos^{-1}\left(\frac{\lambda(f_{\text{beat}} - f_i)}{|v|}\right). \qquad (5)$$

Angular separation between two receiver nodes R_i and R_j can then be computed by subtracting one bearing from the other, as

$$\alpha_{ij} = \beta_j - \beta_i. \qquad (6)$$

One drawback with such a system is that it requires knowledge of the current speed of the mobile node. We argue that such information can easily be obtained by using wheel-mounted optical encoders, which are common components found on most mobile robots. Such sensors are not always adequate on their own for dead reckoning because they accrue error over time. However, encoders are inexpensive, and provide good approximations of *instantaneous velocity*, which make them suitable for this type of system. We demonstrate in Section 4.3 that the error associated with optical encoder measurements is minimal.

3.2 Frequency Estimation Using Resource Constrained Hardware

One problem with the inexpensive radio chip is that the transmission frequency can differ from the nominal frequency by up to 65 Hz due to the tuning precision of the hardware [34]. For this reason, we treat the transmission frequency as a random variable, which results in the beat frequency being a random variable as well. This poses a challenge, because we require knowledge of the beat frequency to compute the receiver bearings. Therefore, in order to determine receiver bearing, we use maximum likelihood (ML) estimation [20].

For ML estimation, we rewrite Eq. 5 as

$$f_i = F(\beta_i, f_{\text{beat}}) + \epsilon_i$$

$$= f_{\text{beat}} - \frac{|v|}{\lambda}\cos\beta_i + \epsilon_i$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma_f)$ is the Gaussian noise in the observed Doppler-shifted frequency. The negative log-likelihood for f_i is given by

$$\ell_i(f_{\text{beat}}, \beta_i) = -\ln p(f_i | f_{\text{beat}}, \beta_i)$$

$$= \frac{\|f_i - F(\beta_i, f_{\text{beat}})\|^2}{\sigma_f^2} - \ln\frac{1}{\sqrt{2\pi\sigma_f^2}}.$$

Assuming N receivers, the combined negative log-likelihood for $f_i, i = 1, \cdots, N$ is given by

$$\ell(f_{\text{beat}}, \beta_1, \cdots, \beta_N) = -\ln p(f_1, \cdots, f_N |$$

$$f_{\text{beat}}, \beta_1, \cdots, \beta_N)$$

$$= -\ln\prod_{i=1}^{N} p(f_i | f_{\text{beat}}, \beta_i)$$

$$= \sum_{i=1}^{N} \ell_i(f_{\text{beat}}, \beta_i)$$

$$= \sum_{i=1}^{N} \frac{\|f_i - F(\beta_i, f_{\text{beat}})\|^2}{\sigma_f^2}$$

$$- N\left(\ln\frac{1}{\sqrt{2\pi\sigma_f^2}}\right).$$

The ML estimate can be obtained by minimizing the negative log-likelihood using the following

$$\frac{\partial\ell(f_{\text{beat}}, \beta_1, \cdots, \beta_N)}{\partial f_{\text{beat}}} = 0.$$

The partial derivative leads to the following result for the ML estimate of the beat frequency

$$\widehat{f}_{\text{beat}} = \frac{1}{N}\sum_{i=1}^{N} f_i + \frac{|v|}{\lambda N}\sum_{i=1}^{N}\cos\beta_i.$$

Note that the ML estimate, $\widehat{f}_{\text{beat}}$, is in terms of $\beta_i, i = 1, \cdots, N$. To solve for the angles, we iteratively compute the ML estimate and the angles. The two iterative steps are given below.

1. Computing the angles:

$$\beta_{i,k} = \cos^{-1}\left(\frac{\lambda(\widehat{f}_{\text{beat}_{k-1}} - f_i)}{|v|}\right)$$

2. Computing the ML estimate for the beat frequency:

$$\widehat{f}_{\text{beat}_k} = \frac{1}{N} \sum_{i=1}^{N} f_i + \frac{|v|}{\lambda N} \sum_{i=1}^{N} \cos \beta_{i,k}$$

where $k = 1, \cdots, 10$ is the iteration index, and the ML estimate is initialized with the average of the observed Doppler-shifted frequencies, $\widehat{f}_{\text{beat}_0} = \frac{1}{N} \sum_{i=1}^{N} f_i$.

The initial beat frequency estimate averages all frequency measurements. At each subsequent iteration, only those Doppler frequency measurements are considered in the equation for which the cosine falls in the $[-1, +1]$ interval. The number of such measurements can be understood as those falling in a window around the previous beat frequency. At each iteration, the window is centered around the previous beat frequency estimate. Since at each iteration a few measurements that are outside the window are discarded, after a few iterations all the frequency measurements outside of the window will be discarded and the beat frequency estimate will converge.

We show typical convergence results for the beat frequency in Fig. 4. The data in the figure were obtained from an actual test on our experimental platform (see Section 6). We observed that the beat frequency estimate converges within a small number of iterations, hence we conservatively chose 10 iterations for the iterative algorithm. A theoretical analysis of convergence of the algorithm is beyond the scope of this work.

4 Error Analysis

We have identified five main sources of error in this system. In order to understand the effect each source of error has on the overall performance of the system, we use a simulated experimental setup that consists of a single mobile sensor and single anchor node. Because the relationship between observed frequency and bearing is dependent on the relative speed between the two nodes (see Eq. 5), we perform repeated simulations, rotating the anchor node around the mobile node at 1° increments. The mobile node is considered to be moving at a speed of 1 m/s with an orientation of zero degrees, and the distance between the mobile node and anchor is 10 m. Figure 5 illustrates this setup.

4.1 Nonlinearity of the Bearing Estimation Equation

The relationship between bearing and observed frequency is

$$\beta_i = \cos^{-1} \left(\frac{\lambda (f_{\text{beat}} - f_i)}{|v|} \right) \qquad (7)$$

Fig. 4 Convergence results for the beat frequency estimate with the maximum likelihood estimation algorithm
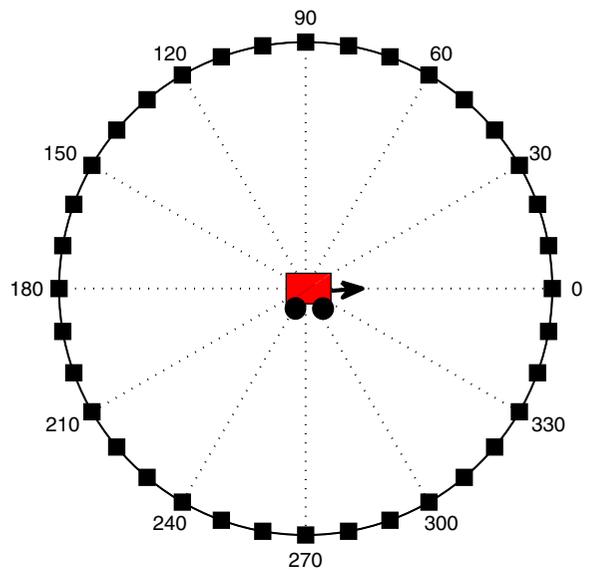
Fig. 5 Simulation setup. Anchor nodes are placed at 1° increments around the mobile node (some nodes not pictured for clarity)

where β_i is the bearing to the mobile node from anchor node i, f_{beat} is the frequency of the beat signal, f_i is the Doppler-shifted frequency observed at anchor node i, and $|v|$ is the speed of the mobile node.

However, the error in computing the bearing β will vary due to the nonlinearity of Eq. 7. Figure 6a shows the structure of the inverse cosine function ($y = \cos^{-1}(x)$), and its derivative is pictured in Fig. 6b. We can see that in general a small error in x will result in a large error in $\cos^{-1}(x)$. This is especially true at the limits ($-0.8 \geq x \geq 0.8$). To avoid this problem, we can examine the argument to the inverse cosine, and if too large or small, discard the sensor data for the current measurement round. In practice, we found this gives us a marginal error reduction of approximately 11 %, or 0.035 rad.

Figure 7 plots the expected arccosine argument under ideal conditions at different anchor node bearings. We can see that the arccosine argument will exceed the bounds illustrated in Fig. 6 when $325° \geq \beta \leq 35°$ and $145° \leq \beta \leq 215°$. In light of this, for the remainder of this error analysis, we will discard all samples that fall within these ranges. Under non-ideal conditions (when there is noise in the system), the range of unusable anchor node measurements will increase. Therefore, we must deploy a redundant number of anchor nodes around the sensing region to ensure a sufficient amount of acceptable measurements at bearings that do not fall within these undesirable ranges.
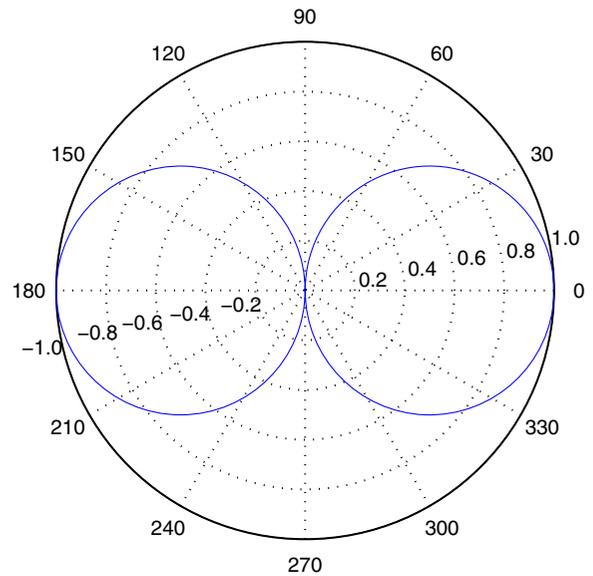
Fig. 7 Expected arccosine argument under ideal conditions at different anchor node bearings

4.2 Measurement Noise

It was reported in [23] that the standard deviation of the measured beat frequency is 0.21 Hz. The primary sources of randomness are timing variations due to clock drift and non-deterministic radio effects. Because our environment and mote hardware configuration are equivalent to that in [23], we consider this metric valid for use in our error analysis. We therefore generate a dataset of 1,000 samples for each anchor position. The

Fig. 6 **a** The inverse cosine function and **b** its derivative
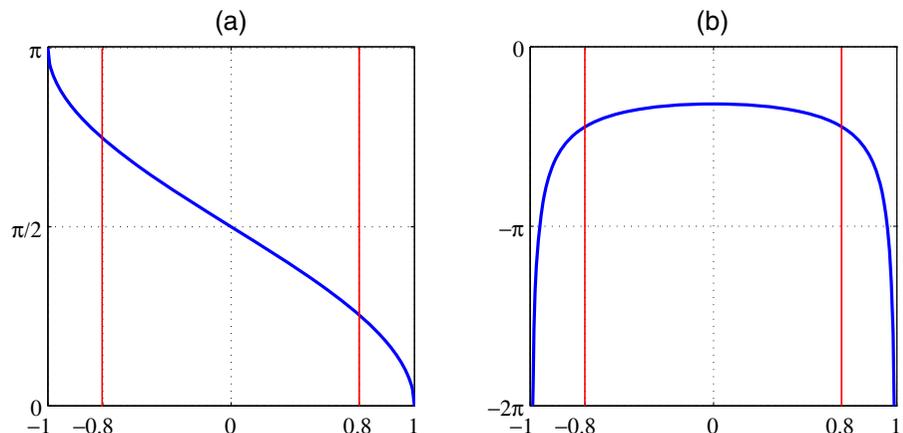
dataset has a mean of the expected frequency and standard deviation of 0.21 Hz. For each anchor position, we would like to know the average error due to measurement noise we expect to see in the system. On average, the bearing error due to measurement noise is 0.12 rad. Figure 8 plots the error.

4.3 Noisy Encoder Data

Typical optical encoders have an instantaneous velocity error distribution with a standard deviation of approximately 1 % of the speed. We examine how encoder error affects the bearing estimate. The results are displayed in Fig. 9. On average, the bearing error due to encoder noise is 0.023 rad.

4.4 Unknown Beat Frequency

Because we are using low-cost hardware, one problem we encounter is that we are unable to know the exact beat frequency, f_{beat}. Although we instruct the nodes to transmit at specific frequencies, the actual transmission can differ from the nominal value by as much as 2 kHz. Because ultimately it is the beat frequency that is impor-

Fig. 9 Bearing error (in radians) due to wheel encoder noise

tant, we can tune the transmitters to transmit at frequencies such that the interference signal (as measured by a participating receiver node) will have a beat envelope at the desired frequency. However, the radio hardware has a tuning resolution of 65 Hz, so the actual beat frequency may differ from the desired frequency by up to 65 Hz. In addition, the transmission frequencies will drift from their tuned frequencies over time due to environmental factors such as temperature, humidity, and supply voltage, as well as imprecision in the radio crystal. We see the effects of a variable f_{beat} in Fig. 10, in which we plot the beat frequency observed by a stationary receiver node over 100 successive measurements, 10 s apart. The figure illustrates the degree to which we are unable to estimate the beat frequency.

Because we do not know the beat frequency, we must solve Eq. 7 using maximum likelihood estimation, where f_{beat} is the unknown parameter. ML estimation approximates the unknown beat frequency by considering measurements made at *all* participating receivers, therefore we cannot determine ML estimation error by simply rotating a single anchor receiver around the mobile node. Consequently, we must fix the positions of several receivers. We do this based on the error analysis

Fig. 8 Bearing error (in radians) due to measurement noise
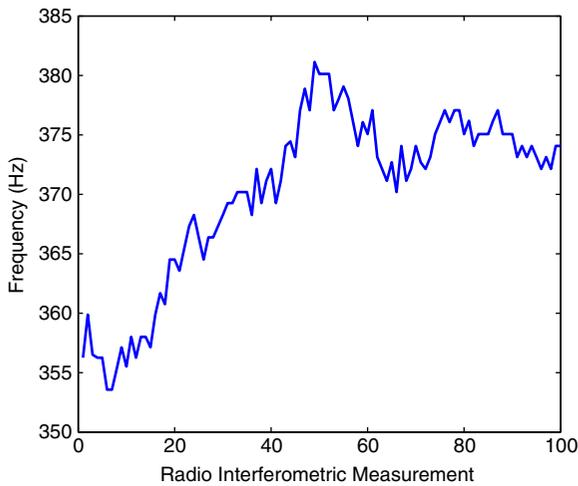
Fig. 10 Observed beat frequency variation over 100 successive transmissions 10 s apart

Table 1 Execution time for each step

Step	Average (ms)	Maximum (ms)
Signal transmission	415	417
Routing	242	561
Angular separation algorithm	28	46
Total	685	1024

4.5 System Latency

Mobile sensors require a rapid positioning algorithm, otherwise by the time the algorithm completes, the mobile node may be in a completely different location. We therefore provide a timing analysis of our algorithm implementation to demonstrate that its latency is acceptable for mobile sensor navigation. Table 1 lists the average and maximum observed execution times for each step involved in the angular separation estimation algorithm. On average, the algorithm takes 685 ms to run.

Based on the fixed speed of the robot at 1 m/s, we can determine the bearing error caused by algorithm latency. In 685 ms, the position of the mobile node will have changed by 0.685 m. The bearing error caused by this position change is 0.035 rad. and is plotted in Fig. 12.

we have performed so far. From the above analysis, we can see the best place to take measurements is by receivers on the flanks of the mobile node. We therefore select four anchor nodes at random, two on each side of the mobile node. We repeat this process 1,000 times and determine the average bearing error. For each iteration, we generate a random beat frequency between 300 and 400 Hz. The error distribution is shown in Fig. 11. The average error is 0.085 rad.
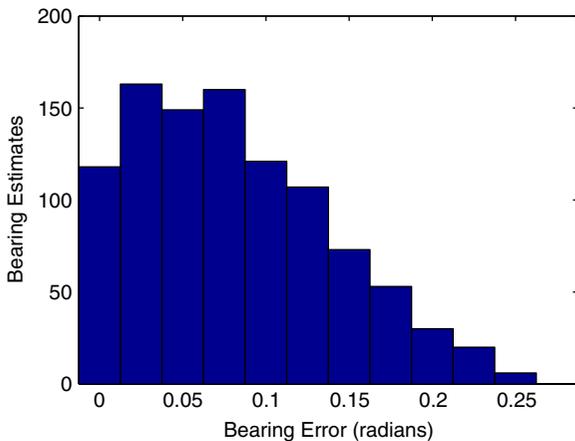
Fig. 11 Bearing error (in radians) due to maximum likelihood estimation

Fig. 12 Bearing error (in radians) due to system latency

4.6 Discussion

Overall, the majority of the angular separation estimation error comes from measurement noise and the ML estimation, accounting for more than 0.2 rad. of error. The other sources, encoder noise and latency, do not contribute significantly to the error. For all receivers, the error due to encoder noise, latency, and the unknown beat frequency will be the same, introducing a systematic bias. Only the error due to the frequency measurements will be different between receivers. Further study is needed to determine whether the overall error can be reduced by taking the systematic bias into consideration.

5 Implementation

Our mobile wireless sensor platform consists of an ExScal mote (XSM) [11] mounted to a MobileRobots Pioneer 3DX [30] robot (see Fig. 13). The anchor nodes are XSM motes as well. All code was written in nesC [15] for the TinyOS operating system [18]. The XSMs use the Texas Instruments CC1000 radio chip [34], and transmit in the 433 MHz band. Note that although the Pioneer comes equipped with an onboard embedded PC, as well as a wide variety of sensors, only the instantaneous velocity obtained from encoder data is used, and all computation is performed on the attached mote. The mote communicates with the robot microcontroller over a serial interface
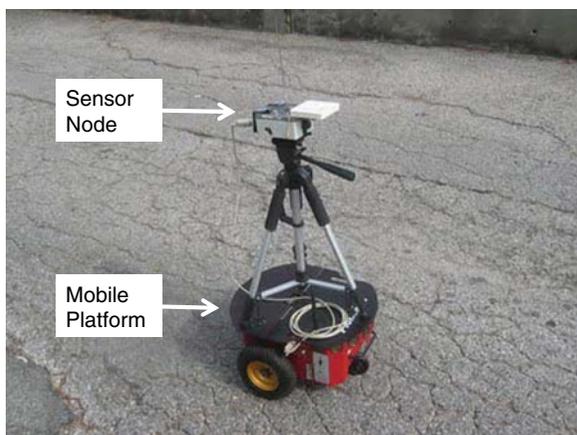
Fig. 13 Mobile sensor implementation

in order to provide angular velocity commands to each wheel.

Because we are using resource-constrained sensor nodes, we are interested in minimizing the memory required to run the algorithm. Our previous work on dNav required the use of two motes on the mobile platform, one hosting the controller, and the other hosting the EKF, leaving little space for the user application [3]. Our current approach requires significantly less memory, using 2.9 kB of RAM and 49.6 kB of program memory (ROM).

6 Evaluation

6.1 Experimental Setup

Our setup consists of six XSM nodes, four of which act as stationary receivers and surround a 45 m × 35 m sensing region. Another stationary node is designated the secondary transmitter, and is placed just outside the sensing region. The final mote is attached to the mobile platform. The mobile node moves around an uneven paved surface in an outdoor environment, mostly free of trees, buildings, and other obstacles. Figure 14 illustrates the experimental setup.

We direct the mobile node to move through the sensing region while transmitting a pure sinusoidal signal. Simultaneously, the second
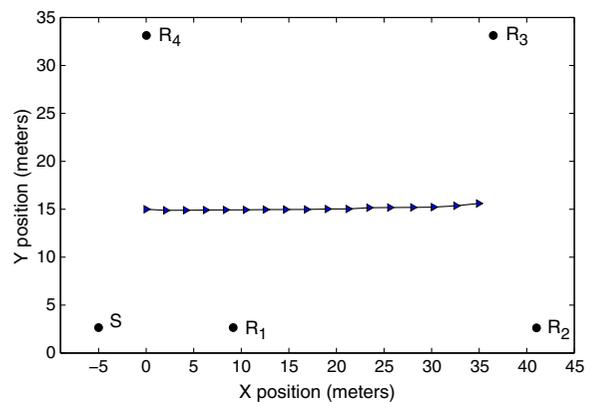
Fig. 14 Experimental setup. Four anchor nodes ($R_1 \ldots R_4$) and the secondary transmitter (S) surround the sensing region. *Triangles* show the direction of travel of the mobile node at each timestep
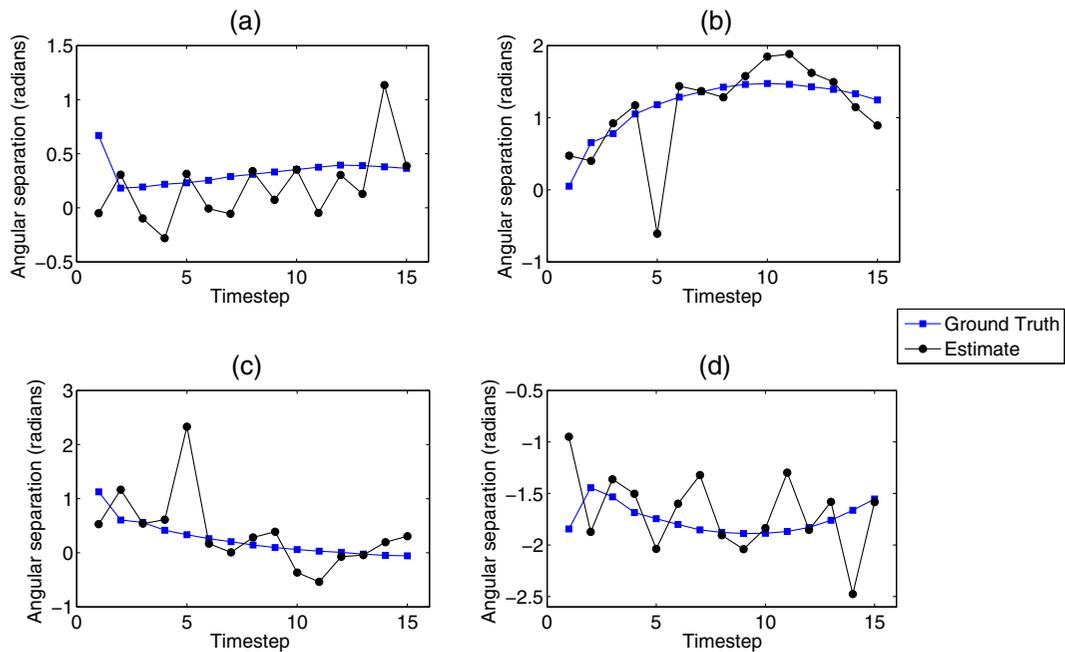
Fig. 15 Ground truth versus estimated angular separation between receiver nodes **a** 1 and 2, **b** 2 and 3, **c** 3 and 4, and **d** 4 and 1, for each measurement as the mobile node traverses the sensing region

transmitter node transmits a pure sinusoidal signal at a slightly lower frequency. The anchor nodes measure the frequency of the resulting interference signal and report their observations back to the mobile node. At the beginning of each measurement round, the mobile node records its instantaneous velocity, obtained from the wheel encoders. This information is then used to derive the angular separation of the anchor nodes. Ground truth is manually measured at each timestep (the time at the beginning of each measurement).

6.2 Experimental Results

Figure 15 shows the estimated versus ground truth angular separations for all pairs of adjacent anchor nodes over the entire course. As the results show, this technique produces a moderate average error of 0.28 rad.

6.3 Latency Analysis

Our method for determining angular separation involves three major steps: (1) signal transmission/

reception, (2) sending observed frequencies from the anchor nodes to the mobile node, and (3) running the angular separation estimation algorithm. We list the average and maximum latencies for these steps in Table 1. The most unpredictable of these steps is the time it takes the anchor nodes to send their observed frequencies to the mobile node. This latency can grow relatively large because the nodes are all attempting to send messages at roughly the same time, resulting in back-off delays. However, even with this unpredictability, we can, on average, obtain angular separation information at a rate of 1.46 Hz, which is sufficient for mobile sensor navigation.

7 Conclusion

In this paper we presented a method for determining angular separation using RF Doppler shifts and wheel encoder data in mobile sensor networks. Angular separation between multiple pairs of anchor nodes can be used for navigation, without the need for localization.

Several implementation challenges were encountered while designing this system. (1) We

initially experimented with a robot that had an effective maximum speed of 400 mm/s; however, this was not fast enough to produce a reliably measurable Doppler shift using the onboard radio hardware. We found that by increasing the robot speed to 1 m/s, we were able to obtain a more accurate measurement. (2) Our algorithm performs better for anchor nodes flanking the robot. Therefore, a sufficient (possibly redundant) number of stationary receivers are required to ensure the necessary number of anchor bearings are obtained. (3) Measuring Doppler-shifted frequencies in resource-constrained mote hardware requires the use of radio interferometry. The addition of the secondary transmitter to the sensing region must be positioned such that its signal reaches all participating stationary receivers. For larger sensing regions, more than one secondary transmitter will be required, along with the switching logic to activate the appropriate one and deactivate all others. In addition, we noticed that in situations where one of the transmitters was much closer to a stationary receiver relative to the other transmitter, the proximal transmission signal would dominate, effectively drowning out the complimentary signal. This further necessitates the need for a redundant number of stationary receivers. (4) Radio hardware limitations caused the actual transmission frequency to be unknown. Because knowledge of the beat frequency was necessary for our algorithm, we use maximum likelihood estimation.

Experimental results obtained using our method had an average error of 0.28 rad, which will provide course-grained navigation. However, in situations where such navigation is acceptable, our approach is fast and requires less memory than other RF-based methods (e.g., [3, 23, 24]). This is because our algorithm is distributed, and therefore we expend no time routing data to a base station for analysis. In addition, determining angular separation from Doppler shifts and instantaneous velocity does not require complex statistical tools, such as a Kalman filter, reducing the overall memory footprint of the application.

References

1. Altun, K., Koku, A.: Evaluation of egocentric navigation methods. In: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), pp. 396–401 (2005)
2. Amundson, I., Koutsoukos, X.: A survey on localization for mobile wireless sensor networks. In: Fuller, R., Koutsoukos, X. (eds.) Proceedings of the 2nd International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT). LNCS, vol. 5801, pp. 235–254. Springer (2009)
3. Amundson, I., Koutsoukos, X., Sallai, J.: 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), pp. 97–102. ACM, San Francisco, CA, USA (2008)
4. Amundson, I., Koutsoukos, X., Sallai, J., Ledeczi, A.: Mobile sensor navigation using rapid RF-based angle of arrival localization. In: Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 316–325. Chicago, IL, USA (2011)
5. Amundson, I., Kushwaha, M., Koutsoukos, X.: On the feasibility of determining angular separation in mobile wireless sensor networks. In: Fuller, R., Koutsoukos, X. (eds.) Proceedings of the 2nd International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT). LNCS, vol. 5801, pp. 115–127. Springer (2009)
6. Amundson, I., Sallai, J., Koutsoukos, X., Ledeczi, A.: Radio interferometric angle of arrival estimation. In: Sá Silva, J., et al. (eds.) Proceedings of the 7th European Conference on Wireless Sensor Networks (EWSN). LNCS, vol. 5970, pp. 1–16. Springer, Coimbra, Portugal (2010)
7. Bekris, K.E., Argyros, A.A., Kavraki, L.E.: Angle-based methods for mobile robot navigation: reaching the entire plane. In: Proceedings of the International Conference on Robotics and Automation (ICRA), pp. 2373–2378. New Orleans, LA (2004)
8. Betke, M., Gurvits, L.: Mobile robot localization using landmarks. IEEE Trans. Robot. Autom. **13**(2), 251–263 (1997)
9. Chen, J., Yao, K., Hudson, R.: Source localization and beamforming. IEEE Sig. Proc. Mag. **19**(2), 30–39 (2002)
10. Dantu, K., Rahimi, M., Shah, H., Babel, S., Dhariwal, A., Sukhatme, G.S.: Robomote: enabling mobility in sensor networks. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN), p. 55 (2005)

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

12. Esteves, J., Carvalho, A., Couto, C.: Generalized geometric triangulation algorithm for mobile robot absolute self-localization. In: Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE), pp. 346–351 (2003)

13. Friedman, J., Charbiwala, Z., Schmid, T., Cho, Y., Srivastava, M.: Angle-of-arrival assisted radio interferometry (ARI) target localization. In: Proceedings of the Military Communications Conference (MILCOM), pp. 1–7 (2008)

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

15. Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., Culler, D.: The nesC language: a holistic approach to networked embedded systems. In: Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), pp. 1–11 (2003)

16. Brown, Jr., H.B., Vandeweghe, J.M., Bererton, C.A., Khosla, P.K.: Millibot trains for enhanced mobility. IEEE/ASME Trans. Mechatronics **7**(4), 452–461 (2002)

17. Hill, J., Culler, D.: Mica: a wireless platform for deeply embedded networks. IEEE Micro **22**(6), 12–24 (2002)

18. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for networked sensors. In: Proceedings of the Ninth International Conference on Achitectural Support for Programming Languages and Operating Systems (ASPLOS-IX), pp. 93–104 (2000)

19. Hong, J., Tan, X., Pinette, B., Weiss, R., Riseman, E.: Image-based homing. Cont. Syst. Mag. IEEE **12**(1), 38–45 (1992)

20. Kay, S.M.: Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory. Prentice Hall (1993)

21. Koku, A., Sekmen, A., Wilkes, D.: A novel approach for robot homing. In: Proceedings of 2003 IEEE Conference on Control Applications, 2003. CCA 2003, vol. 2, pp. 1477–1482 (2003)

22. Kusy, B., Amundson, I., Sallai, J., Volgyesi, P., Ledeczi, A., Koutsoukos, X.: RF doppler shift-based mobile sensor tracking and navigation. ACM Trans. Sensor Netw. **7**(1), 1–32 (2011)

23. Kusý, B., Lédeczi, A., Koutsoukos, X.: Tracking mobile nodes using RF doppler shifts. In: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys), pp. 29–42. Sydney, Australia (2007)

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

25. Ladd, A.M., Bekris, K.E., Rudys, A., Marceau, G., Kavraki, L.E., Wallach, D.S.: Robotics-based location sensing using wireless ethernet. In: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom), pp. 227–238 (2002)

26. Loizou, S., Kumar, V.: Biologically inspired bearing-only navigation and tracking. In: 46th IEEE Conference on Decision and Control, pp. 1386–1391 (2007)

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

28. McGillem, C., Rappaport, T.: A beacon navigation method for autonomous vehicles. IEEE Trans. Veh. Technol. **38**(3), 132–139 (1989)

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

30. MobileRobots: Pioneer P3-Dx. http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx (2012). Accessed 3 Oct 2012

31. Niculescu, D., Nath, B.: Ad hoc positioning system (APS) using AOA. In: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 1734–1743 (2003)

32. Purohit, A., Sun, Z., Salas, M., Zhang, P.: SensorFly: Controlled-mobile sensing platform for indoor emergency response applications. In: Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN) (2011)

33. Shah, R., Roy, S., Jain, S., Brunette, W.: Data MULEs: modeling a three-tier architecture for sparse sensor networks. In: Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, pp. 30–41 (2003)

34. Texas Instruments: CC1000: Single chip very low power RF transceiver. http://focus.ti.com/docs/prod/print/cc1000.html (2007). Accessed 3 Oct 2012

35. Wang, G., Cao, G., Porta, T., Zhang, W.: Sensor relocation in mobile sensor networks. In: Proceedings of the IEEE International Conference on Computer Communication (INFOCOM), pp. 2302–2312 (2005)