

# On the Feasibility of Determining Angular Separation in Mobile Wireless Sensor Networks

Isaac Amundson, Manish Kushwaha, and Xenofon Koutsoukos

Institute for Software Integrated Systems (ISIS)  
Department of Electrical Engineering and Computer Science  
Vanderbilt University  
Nashville, TN 37235, USA  
`isaac.amundson@vanderbilt.edu`

**Abstract.** 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 nodes. We propose a technique for obtaining angle-of-arrival information that uses the wheel encoder data from the mobile sensor, and the RF Doppler-shift observed by stationary nodes. These sensor data are used to determine the angular separation between stationary beacons, which can be used for navigation. Our experimental results demonstrate that using this technique, a robot is able to determine angular separation between four pairs of sensors in a 40 x 40 meter sensing region with an average error of 0.28 radian.

## 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 wireless sensors such as [1] and [2], nodes are able to traverse the sensing region under their own control. This has numerous advantages, such as enabling targeted coverage [3] and connecting disjoint sensor networks [4].

Arguably one of the biggest challenges for mobile sensors is navigation, where the mobile node must reach point  $B$  from point  $A$ . For the most basic wheeled mobile robots (WMRs), navigation is typically solved using odometry, whereby the robot monitors the angular velocity of each wheel to approximate the distance traveled over a given time period. The angular velocity is often determined from feedback from 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. Most other sensors used for navigation (e.g., GPS, laser rangefinders, sonar, etc.) are either too large, heavy, expensive, or require too much power to

operate over extended periods of time. 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.

Recent work [5] has shown that navigation is possible without knowing the current position of the mobile sensor. In fact, all points in a plane are reachable if the angular separation between two pairs of beacons can be determined. In many situations, navigating without having to determine position is advantageous because most localization methods require extensive PC processing, have high localization latency, and require the positions of infrastructure nodes to be known. In previous work [6], we used the Doppler shift in radio transmission frequency as control feedback to drive a robot. Although the robot was able to accurately navigate the sensing region, the localization algorithm relied on the use of an extended Kalman filter (EKF) for noisy frequency measurements. The EKF was run in realtime on the robot, but its large size required execution on a separate mote, which communicated with the robot controller over a wireless interface.

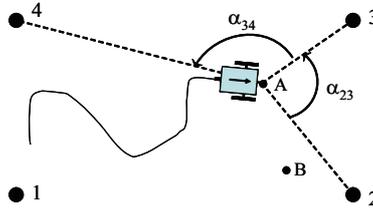
In this paper, we present a method for determining angular separation that only requires the sensor radio and wheel encoders, both of which are common to robotic wireless sensors, and hence no additional hardware is required. Our method uses the Doppler shift in radio frequency and the instantaneous velocity of a mobile node transmitting a pure sinusoidal signal to derive the angular separation between infrastructure nodes surrounding the sensing region. Our method does not require the positions of the infrastructure 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 radian.

The remainder of this paper is organized as follows. In Section 2, we describe our problem statement, followed by our method for angular separation estimation in Section 3. Our implementation on a mobile wireless sensor platform is described in Section 4. Experimental results are then presented in Section 5. Section 6 concludes.

## 2 Problem Formulation

Consider a sensing region that contains multiple infrastructure nodes, as well as a mobile sensor that needs to travel from point  $A$  to point  $B$ . This scenario is illustrated in Figure 1.

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 where we are. Localization in wireless sensor networks is a well-studied problem, and several different approaches have been developed [7], [8]. Typically, localization is accomplished by using a set of coordinated nodes, whereby one or more nodes emit a signal, such

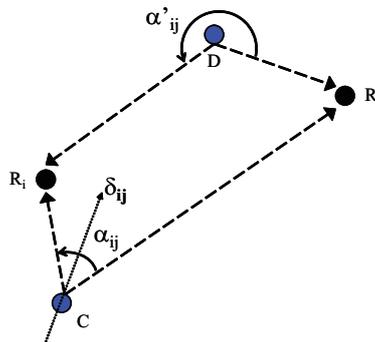


**Fig. 1.** A mobile sensor moves through the sensing region. The mobile node navigates based on angular separation of beacons (numbered 1 through 4).

as radio, ultrasound, infrared, or acoustic. Node position is then computed based on properties of signal arrival, such as time-of-arrival, time-difference-of arrival, received signal strength, angle-of-arrival, or by other methods.

However, localization is not necessary for navigation. In [5], a navigation method is presented that enables mobile entities to reach any position on a plane based on the angular separation between two pairs of beacons. Figure 2 illustrates the setup, simplified for clarity. In the actual setup, at least three beacons are required for navigation. In the figure, the current position of the mobile node is denoted by C, and the destination position by D. We do not know these actual positions, nor do we know the positions of nodes  $R_i$  and  $R_j$ . The only information we have is the angle  $\angle R_i C R_j$  at our present position, denoted by  $\alpha_{ij}$ , as well as the angle  $\angle R_i D R_j$  at our destination, denoted by  $\alpha'_{ij}$ . What we require is a control law that takes these angles as input and provides us with the necessary motion vector to reach our destination.

With current and destination angle information, the objective is to minimize the difference,  $\Delta\alpha_{ij} = \alpha'_{ij} - \alpha_{ij}$ , between the two. When  $\Delta\alpha_{ij} = 0$ , the mobile node will have reached its destination. For example, if  $\Delta\alpha_{ij} > 0$ , this means that  $\angle R_i D R_j > \angle R_i C R_j$ . Therefore, the mobile node should move along a vector that increases  $\alpha_{ij}$ . The unit length bisector vector  $\delta_{ij}$  is one such vector. Based on this reasoning, the complete motion vector developed in [5] takes the form



**Fig. 2.** Navigation without localization based on beacon angular separation

$$\mathbf{M}_{ij} = \begin{cases} \Delta\alpha_{ij} \cdot \delta_{ij}, & -\pi \leq \Delta\alpha_{ij} \leq \pi \\ (2\pi - \Delta\alpha_{ij}) \cdot \delta_{ij}, & \Delta\alpha_{ij} > \pi \\ (-2\pi - \Delta\alpha_{ij}) \cdot \delta_{ij}, & \Delta\alpha_{ij} < -\pi \end{cases}$$

As the mobile node moves along the bisector  $\delta_{ij}$ , it is actually following a segment of a hyperbolic curve, defined by the two foci  $R_i$  and  $R_j$ . By following such a curve, the mobile node is guaranteed to reach the arc  $R_iDR_j$ . With only two beacons, position D cannot be identified on this arc because for every point,  $\mathbf{M}$  becomes 0. However, with three beacons, D is constrained to lie on two or more arcs. In other words,  $\mathbf{M} = \mathbf{M}_{ij} + \mathbf{M}_{jk} + \mathbf{M}_{ki}$ , and the destination D is reached when  $\forall ij : \Delta\alpha_{ij} = 0$ .

Often, angle information is determined using cameras, microphone arrays, or light pulses, all of which are not ideal for lightweight mobile sensors. We would like to estimate angular separation using only hardware that is widely available on sensor nodes. Specifically, we would like to be able to obtain this angle information using the sensor node radio and the optical encoders on the wheels. Although we have not implemented the above navigation method for our current work, we refer to it here as motivation for our angular separation estimation technique, and have plans to integrate it in the near future.

### 3 Estimation of Angular Separation

In this section, we describe the design of our angular separation estimation method for mobile sensors.

A mobile node,  $T$ , moving through the sensing region with velocity  $\mathbf{v}$  and heading  $\varphi$ , collects angular velocity data from its wheel encoders. For WMRs with 2-wheel differential steering, the relationship between the robot speed and the wheel angular velocities is

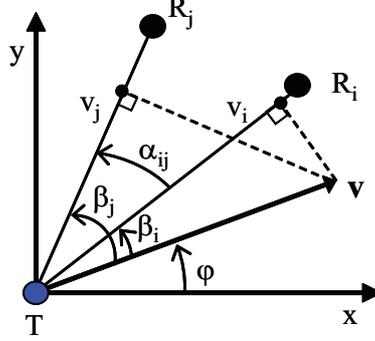
$$|v| = \frac{r(\omega_r + \omega_l)}{2} \quad (1)$$

where the speed  $|v|$  is the magnitude of the velocity  $\mathbf{v}$ ,  $r$  is the wheel radius, and  $\omega_r$  and  $\omega_l$  are the right and left wheel angular velocities, respectively.

As the mobile node moves, it transmits an RF sinusoidal signal, which is observed by the receiver nodes. 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. This relationship is formalized as

$$f_i = f_{carrier} - \frac{v_i}{\lambda} \quad (2)$$

where  $f_i$  is the observed Doppler-shifted frequency at receiver  $R_i$ ,  $f_{carrier}$  is the transmission frequency of the carrier signal with wavelength  $\lambda$ , and  $v_i$  is the relative speed of mobile node  $T$  with respect to receiver  $R_i$ .



**Fig. 3.** Geometry of simplified setup for determining angular separation between two receivers

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  $\alpha_{ij}$  between the two receiver nodes based on the measured quantities of  $\omega_r$ ,  $\omega_l$ ,  $f_i$ , and  $f_j$ , and known values  $f_{carrier}$  and  $\lambda$ .

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

$$v_i = |v|\cos\beta_i \quad (3)$$

where the speed of the mobile node,  $|v|$ , has a negative sign if  $T$  is moving toward  $R_i$  and positive otherwise, and  $\beta_i$  is the angle between the velocity vector  $\mathbf{v}$  and the position vector  $\overrightarrow{TR_i}$ .

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

$$v_i = \lambda(f_{carrier} - f_i) \quad (4)$$

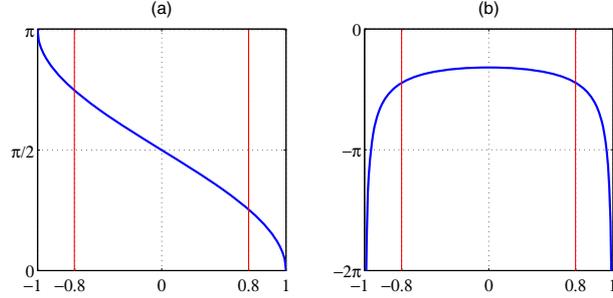
Combining Equations (3) and (4), and rearranging, we can calculate  $\beta_i$  by

$$\beta_i = \cos^{-1}\left(\frac{\lambda(f_{carrier} - f_i)}{|v|}\right) \quad (5)$$

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

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

The error in computing the bearing  $\beta$  will vary due to the nonlinearity of Equation (5). Figure 4a shows the structure of the inverse cosine function ( $y = \cos^{-1}(x)$ ), and its derivative is pictured in Figure 4b. We can see that at the limits ( $-0.8 \geq x \geq 0.8$ ), a small error in  $x$  will result in a large error in  $\cos^{-1}(x)$ . To avoid this problem, we examine the argument to the inverse cosine, and if



**Fig. 4.** (a) The inverse cosine function and (b) its derivative

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  $2^\circ$ , or 11%.

### 3.1 Frequency Estimation Using Resource-Constrained Hardware

Typical low-cost sensor hardware supports radios that transmit in the 400 MHz — 2.6 GHz range. These radios have a received signal strength indicator (RSSI) pin which can be accessed from software, however, we cannot sample the RSSI fast enough to determine the carrier frequency,  $f_{carrier}$ , of the signal. Instead, we use radio interferometry [9], 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,  $f_{beat}$ . The assistant 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 (350 Hz in our case), can be sampled by making successive reads of the RSSI.

Another issue with the inexpensive radio chip is that the transmission frequency can vary from the desired frequency by up to 65 Hz. For this reason, we treat the transmission frequency as a random variable, which results in the beat frequency,  $f_{beat}$ , being a random variable as well. This poses a challenge, because we require knowledge of the beat frequency to compute the receiver bearings. In order to solve Equation (5), we use maximum likelihood (ML) estimation [10].

For ML estimation, we rewrite Equation (5) as

$$\begin{aligned} f_i &= F(\beta_i, f_{beat}) + \epsilon_i \\ &= f_{beat} - \frac{|v|}{\lambda} \cos \beta_i + \epsilon_i \end{aligned}$$

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

$$\ell_i(f_{beat}, \beta_i) = -\ln p(f_i | f_{beat}, \beta_i) = \frac{\|f_i - F(\beta_i, f_{beat})\|^2}{\sigma_f^2}$$

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

$$\begin{aligned} \ell(f_{beat}, \beta_1, \dots, \beta_N) &= -\ln p(f_1, \dots, f_N | f_{beat}, \beta_1, \dots, \beta_N) \\ &= -\ln \prod_{i=1}^N p(f_i | f_{beat}, \beta_i) \\ &= \sum_{i=1}^N \ell_i(f_{beat}, \beta_i) \\ &= \sum_{i=1}^N \frac{\|f_i - F(\beta_i, f_{beat})\|^2}{\sigma_f^2} \end{aligned}$$

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

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

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

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

Note that the ML estimate,  $\hat{f}_{beat}$  is in terms of  $\beta_i, i = 1, \dots, N$ . 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:

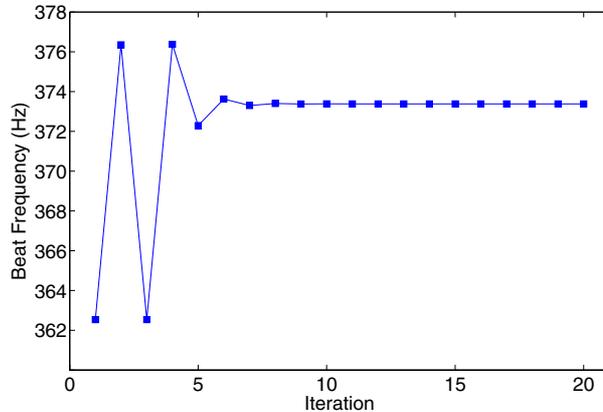
$$\beta_{i,k} = \cos^{-1} \left( \frac{\lambda(\hat{f}_{beat_{k-1}} - f_i)}{|v|} \right)$$

2. Computing the ML estimate for the beat frequency:

$$\hat{f}_{beat_k} = \frac{1}{N} \sum_{i=1}^N f_i + \frac{|v|}{\lambda N} \sum_{i=1}^N \cos \beta_{i,k}$$

where  $k = 1, \dots, 20$  is the iteration index, and the ML estimate is initialized with the average of the observed Doppler-shifted frequencies,  $\hat{f}_{beat_0} = \frac{1}{N} \sum_{i=1}^N f_i$ .

We show the convergence results for the beat frequency in Figure 5. We observed that the beat frequency estimate converges within a couple of iterations, hence we conservatively chose 20 iterations for the iterative algorithm. A theoretical analysis of convergence of the algorithm is beyond the scope of this paper.



**Fig. 5.** Convergence results for beat frequency estimate with the maximum likelihood estimation algorithm

## 4 Implementation

Our wireless sensor platform consists of ExScal motes (XSMs) [11] and a Mobile-Robots Pioneer 3DX [12] robot. All code was written in nesC [13] for the TinyOS operating system [14]. The XSMs use the Texas Instruments CC1000 radio chip [15], 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.

### 4.1 Implementation Benchmarking

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. Our method for determining angular separation involves three major steps: (1) signal transmission/reception, (2) sending observed frequencies from the infrastructure 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 infrastructure 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.

Because we are using resource-constrained sensor nodes, we are also interested in minimizing the memory required to run the algorithm. Our previous

**Table 1.** Execution time for each step

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

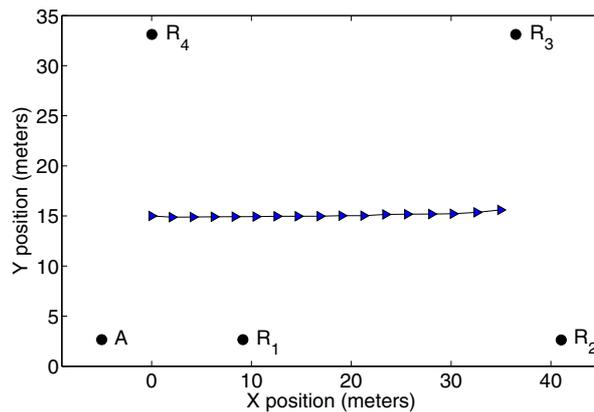
approach [6] required the use of two nodes on the mobile platform, one hosting the controller, and the other hosting the EKF, leaving little space for the user application. Our current approach requires significantly less memory, using 2.9 kB of RAM and 49.6 kB of program memory (ROM).

## 5 Evaluation

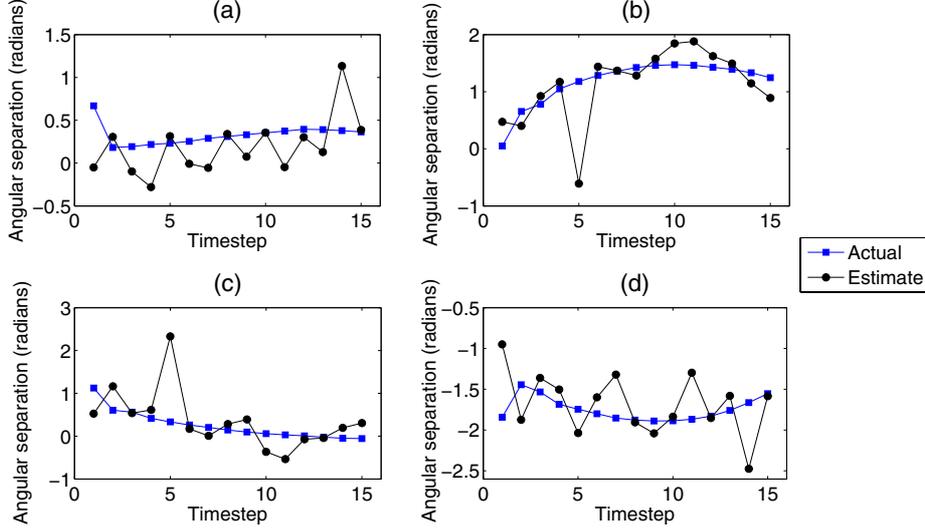
### 5.1 Experimental Setup

Our setup consists of 6 XSM nodes, four of which act as stationary receivers and surround the sensing region. Another stationary node is designated the assistant transmitter, and is placed just outside the sensing region. The final node is attached to the robot. The robot moves around an uneven paved surface in an outdoor environment, mostly free of trees, buildings, and other obstacles. Figure 6 illustrates the experimental setup.

We direct the mobile node to move through the sensing region while transmitting a pure sinusoidal signal. The infrastructure nodes measure the frequency of the signal and report their observations back to the mobile node. At the beginning of each measurement round, the mobile node records its instantaneous



**Fig. 6.** Experimental setup. Four infrastructure nodes ( $R_1 \dots R_4$ ) and the assistant transmitter (A) surround the sensing region. Triangles show the direction of travel of the mobile node at each timestep.



**Fig. 7.** Actual versus estimated angular separation between receiver nodes (a)  $R_1$  and  $R_2$ , (b)  $R_2$  and  $R_3$ , (c)  $R_3$  and  $R_4$ , and (d)  $R_4$  and  $R_1$ , for each measurement as the mobile node traverses the sensing region

velocity, obtained from the wheel encoders. This information is then used to derive the angular separation of the infrastructure nodes. Ground truth is manually measured at each timestep (the time at the beginning of each measurement).

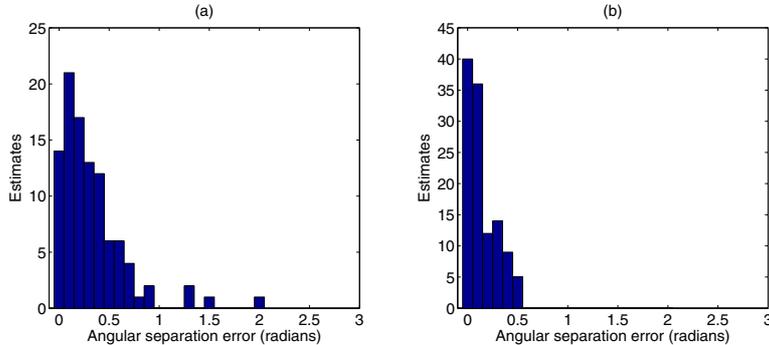
## 5.2 Experimental Results

As the mobile node makes its way through the sensing region, it periodically computes the angular separation between pairs of infrastructure nodes using the technique presented in Section 3. Figure 7 shows the estimated versus actual angular separations for all pairs of adjacent beacons over the entire course. The average error is 0.28 radian.

## 5.3 Discussion

The error in our estimation method comes from three main sources: (1) noisy encoder readings, (2) noisy frequency measurements, and (3) the unknown beat frequency. For all receivers, the error due to encoder noise 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.

An average angular separation error of 0.28 radian will result in course-grained navigation. Because a significant part of this error comes from the estimation



**Fig. 8.** Error distribution for all adjacent pairs of receivers for each robot position using (a) maximum likelihood estimation, and (b) simulating a known beat frequency

of the beat frequency, we would like to know how this system would respond if we were using radio hardware that transmitted exactly at the desired frequency. With such a system we will still expect measurement noise, whose standard deviation was previously reported to be 0.21 Hz [16], and so we account for this. For the simulation, we use the same infrastructure and mobile node positions as our robot experiment. The average error for this simulation is much lower at 0.14 radian. The error distributions for all pairs of receivers for each robot position are shown in Figure 8.

## 6 Conclusion

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

Several implementation challenges were encountered while designing this system. Measuring Doppler-shifted frequencies required the use of radio interferometry. Radio hardware limitations caused the actual transmission frequency to be unknown. Because knowledge of the beat frequency was necessary for our algorithm, we used maximum likelihood estimation, however, the accuracy of the results was lower. Experimental results obtained using our method had an average error of 0.28 radian, which will provide course-grained navigation. However, in situations where such navigation is acceptable, our approach is faster, and requires less memory than other RF-based methods (e.g., [16], [17], [6]). 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.

The goal of our research is to enable mobile sensor navigation using methods that are distributed, require no additional hardware, have low computational complexity, and do not rely on GPS. For future work, we intend to continue this pursuit by examining how we can reduce the angular error for better navigation.

**Acknowledgements.** This work was supported in part 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 Akos Ledecz, Janos Sallai, Peter Volgyesi, Metropolitan Nashville Parks and Recreation, and Edwin Warner Park.

## References

1. Dantu, K., Rahimi, M., Shah, H., Babel, S., Dhariwal, A., Sukhatme, G.S.: Robomote: enabling mobility in sensor networks. In: The Fourth International Symposium on Information Processing in Sensor Networks, IPSN (2005)
2. 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: Prasanna, V.K., Iyengar, S.S., Spirakis, P.G., Welsh, M. (eds.) DCOSS 2005. LNCS, vol. 3560, p. 412. Springer, Heidelberg (2005)
3. Wang, G., Cao, G., Porta, T., Zhang, W.: Sensor relocation in mobile sensor networks. In: IEEE INFOCOM (2005)
4. Shah, R., Roy, S., Jain, S., Brunette, W.: Data mules: modeling a three-tier architecture for sparse sensor networks. In: Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (2003)
5. Bekris, K.E., Argyros, A.A., Kavradi, L.E.: Angle-based methods for mobile robot navigation: Reaching the entire plane. In: International Conference on Robotics and Automation (2004)
6. Amundson, I., Koutsoukos, X., Sallai, J.: Mobile sensor localization and navigation using RF doppler shifts. In: 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, MELT (2008)
7. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. *IEEE Computer* 34(8), 57–66 (2001)
8. Mao, G., Fidan, B., Anderson, B.D.O.: Wireless sensor network localization techniques. *Computer Networks* 51(10), 2529–2553 (2007)
9. Maróti, M., Kusý, B., Balogh, G., Völgyesi, P., Nádas, A., Molnár, K., Dóra, S., Lédeczi, A.: Radio interferometric geolocation. In: Proc. of ACM SenSys (November 2005)
10. Kay, S.M.: *Fundamentals of Statistical Signal Processing, Estimation Theory*, vol. I. Prentice-Hall, Englewood Cliffs (1993)
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: Proc. of IPSN/SPTS (April 2005)
12. MobileRobots: Pioneer P3-DX, <http://www.activrobots.com/ROBOTS/p2dx.html>
13. Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., Culler, D.: The nesC language: a holistic approach to networked embedded systems. In: Proc. of Programming Language Design and Implementation (PLDI) (June 2003)

14. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., Pister, K.: System architecture directions for networked sensors. In: Proc. of ASPLOS-IX (November 2000)
15. Texas Instruments: CC1000: Single chip very low power RF transceiver (2007), <http://focus.ti.com/docs/prod/folders/print/cc1000.html>
16. Kusý, B., Lédeczi, A., Koutsoukos, X.: Tracking mobile nodes using RF doppler shifts. In: SenSys 2007: Proceedings of the 5th international conference on Embedded networked sensor systems, pp. 29–42. ACM, New York (2007)
17. Kusý, B., Sallai, J., Balogh, G., Lédeczi, A., Protopopescu, V., Tolliver, J., DeNap, F., Parang, M.: Radio interferometric tracking of mobile wireless nodes. In: Proc. of MobiSys (2007)