Node Localization and Tracking Using Distance and Acceleration Measurements

Benjamin R. Hamilton, *Xiaoli Ma*,
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia, USA 30332

Robert J. Baxley, and *Brett Walkenhorst*
Information Technology and Telecommunications Laboratory
Georgia Tech Research Institute
Atlanta, Georgia, USA 30318

***Abstract*—Advances in miniaturized wireless and sensing technologies have enabled the construction of cheap, low-powered, portable wireless devices capable of forming ad hoc networks. While these networks have shown enormous potential in applications such as remote sensing and target tracking, these applications require the devices to determine their own location. Additionally, devices capable of self-localization can also be used to implement location-based services or to improve coordination between first-responders to disaster sites or infantry in tactical situations. Existing techniques such as GPS may not be available due to design or environmental constraints, so other methods need to be devised.**

Previous works have proposed methods for wireless devices to self-localize based on received signal strength (RSS), but these methods offer limited accuracy due to the large error in RSS measurements. Recognizing the trend for these portable wireless devices to contain acceleration sensors, we propose an algorithm to combine these acceleration measurements with RSS readings to achieve accurate localization. We apply a distributed extended Kalman filter to track position based on these two measurements and a kinematic node movement model. This algorithm is able to take advantage of correlations between successive location estimates to improve estimation accuracy. We calculate the posterior Cramér-Rao bound for this algorithm and analyze it through simulation. Our analysis shows that by utilizing the acceleration information, the network is able to self-localize despite the large inaccuracy in RSS readings.

I. Introduction

Progressive advances in semiconductor technologies have decreased the size, power requirements and cost of wireless devices. Such advances have allowed modern wireless devices to be increasingly mobile and capable. These advances have also enabled the use of Micro-electromechanical systems (MEMS) technology to construct a variety of cheap and efficient sensors. These sensors can be combined with wireless mobile devices to construct cheap, low-powered, portable wireless devices capable of forming ad-hoc networks.

These networks have shown enormous potential in applications such as environmental and habitat monitoring, intrusion detection, and target tracking [1]. Many of these applications are greatly enhanced when the nodes in the network are able to determine their own location. Nodes that are capable of self-localization can also be used for location-based services or to improve coordination between first-responders at disaster sites or infantry in tactical situations. Alternative localization techniques such as GPS may not be available in some areas: obstructions such as buildings or interference such as that caused by hostile jamming can render GPS useless. Additionally GPS receivers require a significant amount of power and complexity that is too expensive for some applications.

Despite these difficulties, numerous recent works have focused on this problem. Methods have been proposed to use angle, distance or delay measurements [2]–[4]. Some methods even use connectivity information, such as hop counts. Examples include LSVM [5], which uses the support vector machine learning method to determine node locations and Sequential Montecarlo Localization [6], [7] which uses a particle filter based on connectivity information to track node positions.

Several proposed methods use distance or received signal strength (RSS) measurements to determine node location. Methods have been proposed using techniques such as Semidefinite Programming [8], Particle Filters [9] and Probability-based Maximum Likelihood [10]. Multilateration [11], [12] uses the observation update portion of an extended Kalman filter to estimate node locations. The Cramér Rao bound (CRB) for the location estimation accuracy of these methods has also been calculated [13]–[15]. These methods do not take advantage of the accelerometers available on many wireless platforms, which could be used to further increase localization accuracy.

Previous works have also looked into using the accelerometers available on many wireless devices. A “Smart Kindergarten” [16] used the accelerometers to determine the orientation of a subject. Another work [17] used the accelerometer and a magnetometer to aid in localization by detecting node orientation and whether it has been moved.

In this paper we present a novel, hybrid location tracking system combining distance and acceleration measurements to produce more accurate location estimates than otherwise possible. We analyze this method theoretically and derive the Posterior Cramér Rao lower bound. We analyze the expected performance through simulation.

This paper is organized as follows. In Section II, we describe the system model. In Section III we apply this model to the location tracking problem to produce our proposed hybrid location tracking algorithm. We also derive the posterior Cramér Rao bound on its performance. We verify the proposed algorithm’s performance through simulation in Section IV, and conclude in Section V.

II. System Model

We consider a network consisting of N_r static reference nodes and N_m mobile nodes distributed in a two dimensional region such that the density of mobile nodes is D. This network consists of reference nodes that are stationary with globally known locations and mobile nodes with unknown locations. The mobile nodes move according to the movement model described in Section II-A. All nodes are assumed to have a maximum transmission range R.

A. Movement Model

We consider a movement model where the velocity is modeled as an autoregressive (AR) random process. The velocity is assumed to have a state equation such that $v[t+1] = v[t]+w[t]$, where $v[t]$ is the velocity at time t and $w[t]$ is Gaussian random noise of variance σ_w^2. The position is updated using the discrete kinematic equation: $x[t+1] = x[t] + v[t]$.

B. Measurement Model

We assume that nodes are able to obtain a measure of the distance to each of their neighbors. We consider two measurement models: one where distance is measured, and another that uses the received signal strength. Additionally, we assume mobile nodes are capable of measuring their absolute acceleration.

1) Using distance measurements: In the first model, the node is assumed to simply record a noisy measurement of the distance:

$$\hat{d} = d + \gamma,$$

where $\hat{d}$ is the measurement, d is the actual distance, and γ is Gaussian distributed noise. The distance for a given link is calculated as the euclidean distance between the nodes involved: $d_{nm} = \sqrt{(s_n^x - s_m^x)^2 + (s_n^y - s_m^y)^2}$, where (s_n^x, s_n^y) is either the position of mobile node n or the position of the reference node n, if n is a reference node.

2) Using RSS as a proxy for distance: Most common devices do not have specialized hardware capable of directly determining distance. Instead, the distance is usually calculated from the received signal strength (RSS) or the height of the peak of the correlation from the correlator in the receiver. The primary difference between the two is that the correlation peak is more resistant to interference.

The RSS and peak correlation measure the received signal magnitude. We model this received signal magnitude as a pure function of distance disturbed by additive Gaussian noise with zero mean and variance σ_γ^2. So the RSS and peak correlation can be calculated as $\hat{z} = d^{-\alpha} + \gamma$, where α is the distance attenuation exponent. Note that the distance measurement model described previously can be thought of as simply a special case of this model with $\alpha = -1$.

3) Acceleration Measurements: Many common portable wireless devices have begun incorporating acceleration sensors. The readings from these sensors can be combined with either readings from a rotation sensor or compass to measure the absolute acceleration. The methods for achieving this are beyond the scope of this paper. Instead we assume that nodes directly measure the absolute acceleration with additive Gaussian noise η with variance σ_η^2: $\hat{a} = a + \eta$.

III. Location Tracking

We propose a hybrid location tracking system which uses an extended Kalman filter to track the position and velocity of the mobile nodes. In this algorithm, the acceleration measurement from the accelerometer will be added as a control input to the extended Kalman Filter, with the variance from this measurement being used as the state noise. We assume that, since accelerometer measurements require significantly less power than distance measurements, nodes check the accelerometer some k times in the interval between synchronizations over the wireless interface, and that the interval between these checks is T. An alternative formulation would be to track the acceleration as part of the system state, but since the distance synchronization period should be large enough that the correlation in acceleration between these periods is quite small, we can avoid the complexity of additional state variables while still achieving similar performance.

The extended Kalman filter consists of two main parts: the state update and innovation update. In the state update, the state update matrix $\mathbf{F}$ is applied to the previous state $\theta[t|t]$ and the previous mean squared error (MSE) $\mathbf{E}[t|t]$. The control input $\mathbf{c}[t]$ is also added:

$$\theta[t+1|t] = \mathbf{F}\theta[t|t] + \mathbf{c}[t], \tag{1}$$

$$\mathbf{E}[t+1|t] = \mathbf{F}\mathbf{E}[t|t]\mathbf{F}^H + \mathbf{W}. \tag{2}$$

For our filter, the state θ consists of the position (s^x, s^y) and velocity (v^x, v^y) of all the mobile nodes in the network:

$$\theta = \begin{bmatrix} s_0^x & s_0^y & v_0^x & v_0^y & \cdots & s_{N_m-1}^x & s_{N_m-1}^y & v_{N_m-1}^x & v_{N_m-1}^y \end{bmatrix}^T.$$

We will derive the state update matrix $\mathbf{F}$, the state covariance matrix $\mathbf{W}$, and the control input $\mathbf{c}$ in Section III-A.

The innovation update consists of incorporating the measurement with the current estimate. The measurement $\hat{\mathbf{z}}$ is assumed to be a function $h(\theta)$ of the state plus some noise $\gamma[t]$ (with covariance $\mathbf{\Gamma}$):

$$\hat{\mathbf{z}}[t] = h(\theta[t|t]) + \gamma[t]$$

In the case of localization, the measurement vector contains the individual measurements for each of the L links in the network: $\hat{\mathbf{z}} = \begin{bmatrix} z_0 & z_1 & \cdots & z_{L-1} \end{bmatrix}$. Each of these individual measurements are calculated as:

$$z_l = \left((s_n^x - s_m^x)^2 + (s_n^y - s_m^y)^2\right)^{\frac{-\alpha}{2}},$$

where the l^{th} link is between node n and node m.

Since the function $h(\theta)$ is nonlinear in this case, we approximate it with its Jacobian evaluated at the estimated state, forming the Jacobian matrix $\mathbf{H}[t+1]$. The observation can then be incorporated into the state estimate using the Kalman gain $\mathbf{K}[t]$ as a weighting factor.

$$\mathbf{S}[t] = \mathbf{H}[t+1]\mathbf{E}[t+1|t]\mathbf{H}[t+1]^H + \mathbf{\Gamma} \tag{3}$$

$$\mathbf{K}[t] = \mathbf{E}[t+1|t]\mathbf{H}[t+1]^H\mathbf{S}[t]^\dagger \tag{4}$$

$$\theta[t+1|t+1] = \theta[t+1|t] + \mathbf{K}\left(\hat{\mathbf{z}} - h(\theta[t+1|t])\right) \tag{5}$$

$$\mathbf{E}[t+1|t+1] = \left(\mathbf{I} - \mathbf{E}\mathbf{H}[t+1]\right)\mathbf{E}[t+1|t] \tag{6}$$

For the localization problem, this Jacobian is shown in Eq. (7).

A. State Update

In order to determine the state update matrix $\mathbf{F}$, state covariance matrix $\mathbf{W}$, and the control input $\mathbf{c}$, we look briefly into how state changes due to a measured acceleration will affect the state transition from $\theta[t|t]$ to $\theta[t+1|t]$.

While the acceleration has a linear effect on velocity, its affect on position is nonlinear function of time. This means that we cannot directly integrate the measurements of acceleration as the control input. For simplicity, we consider the 1-dimensional case at a single node. Since each of the dimensions and each of the nodes progress independently, this derivation can be easily generalized to a full 2-dimensional network. Nodes use the kinematic equations to update their position each time they checks the accelerometer:

$$\begin{aligned} x[n] &= x[n-1] + v[n-1]T + 0.5a[n]T^2 \\ v[n] &= v[n-1] + a[n]T, \end{aligned}$$

where $x[n-1]$ and $v[n-1]$ are the current position and velocity estimates respectively, and $a[n]$ is the n^{th} acceleration measurement.

In matrix form, we can write:

$$\mathbf{x}[n] = \mathbf{F}_a\mathbf{x}[n-1] + \begin{bmatrix} \frac{a[n]T^2}{2} \\ a[n]T \end{bmatrix},$$

with $\mathbf{F}_a = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$.

Since $\mathbf{x}[n]$ is updated k times between each synchronization interval, we express $\mathbf{x}[k]$ in terms of $\mathbf{x}[0]$ as:

$$\mathbf{x}[k] = \mathbf{F}_a^k\mathbf{x}[0] + \sum_{m=0}^{k-1}\mathbf{F}_a^m \begin{bmatrix} \frac{a[k-m]T^2}{2} \\ a[k-m]T \end{bmatrix}. \tag{8}$$

If we let $\mathbf{x}[0]$ be a node's state in the Kalman filter state vector $\theta[t|t]$ and $\mathbf{x}[k]$ be its state in $\theta[t+1|t]$ and compare Eq. (8) to the state update equation used by the Kalman filter (Eq. (1)), we find that $\mathbf{F} = \mathbf{F}_a^k = \begin{bmatrix} 1 & kT \\ 0 & 1 \end{bmatrix}$ and $\mathbf{w}[n] = \sum_{m=0}^{k-1}\mathbf{F}_a^m \begin{bmatrix} \frac{a[n-m]T^2}{2} \\ a[n-m]T \end{bmatrix}$.

The mean of the state noise $\mathbf{w}$ is then:

$$\begin{aligned} \bar{\mathbf{w}} &= \begin{bmatrix} T\sum_{n=0}^{k-1}\frac{v[n]+v[n-1]}{2} \\ T\sum_{n=0}^{k-1}a[n] \end{bmatrix} && (9) \\ &= \begin{bmatrix} T^2\sum_{n=1}^{k-1}\left(\frac{a[n]}{2} + (k-n)a[n-1]\right) \\ T\sum_{n=0}^{k-1}a[n] \end{bmatrix}, && (10) \end{aligned}$$

and its covariance becomes:

$$\mathbf{W} = \sigma_\eta^2 T^2 \begin{bmatrix} T^2(\frac{k^3}{3} - \frac{k}{12}) & T\frac{k^2}{2} \\ T\frac{k^2}{2} & k \end{bmatrix}.$$

We use the mean of the state noise from Eq. (10) as the control input $\mathbf{c}[t]$. Instead of performing the summation, we maintain an accumulator $\mathbf{c}_a$ which keeps a running total of the offset in position and velocity due to the acceleration measurements. This vector is updated according to:

$$\mathbf{c}_a[n+1] = \mathbf{c}_a[n] + \begin{bmatrix} \mathbf{a}[n]T^2(k-n+0.5) \\ \mathbf{a}[n]T \end{bmatrix},$$

where $\mathbf{a}[n]$ is the measured acceleration at the n^{th} time the accelerometer was checked during the current interval and $\mathbf{c}_a[0] = \mathbf{0}$. This accumulator is then used as the control input (i.e. $\mathbf{c} = \mathbf{c}_a[k]$) each time the node synchronizes over the wireless interface.

Note however that this technique will tend to underestimate the error due to the observations of the acceleration. The true acceleration is a continuous process, and only the instantaneous acceleration is measured. This measurement is used as the average acceleration over the entire interval between checks. Since the acceleration actually changes during this interval, there will be an additional amount of error. If the checking interval is made sufficiently small, the acceleration will be rougly constant, and this error will become negligible.

B. Modified Multilateration

For comparison purposes we define a trivial modification of the Multilateration algorithm [11], [12] allowing it to use a full extended Kalman filter to track the position. The original method assumed a static network, and only used the innovation update portion of the Kalman filter. We extend this into a full extended Kalman filter by assuming the position can be modeled as an AR process. This means that the state update equation will be very similar to Eq. (1) in our hybrid location tracking algorithm:

$$\theta[t+1|t] = \mathbf{F}\theta[t|t] + \mathbf{w}[t],$$

where the state θ only contains node positions and $\mathbf{F}$ is the identity matrix. We will compare our hybrid location tracking algorithm to this method in Section III-C and Section IV.

C. Theoretical Bounds on Performance

In [18] the Posterior Cramér Rao Bound (PCRB) for nonlinear filtering was proposed. In this section we apply this bound to the proposed tracking algorithms.

The Posterior Cramér Rao Bound (PCRB) [18] provides a lower bound for the mean-squared error (MSE) due to estimation or a non-linear dynamic system. This error lower bounded by the diagonal elements of the inverse of the Fisher information matrix $\mathbf{J}$. Since we assume the state and observation noise are both additive Gaussian and the state update is linear, we can express the Fisher information matrix for the system state at time $t+1$ based on the first t observations using the recursion:

$$[\mathbf{H}[t]]_{ik} = \begin{cases} -\alpha(s_n^x - s_m^x) z_k^{\frac{\alpha+2}{\alpha}} & \text{when the } k^{\text{th}} \text{ link is between nodes } n \text{ and } m \text{ and } s_n^x \text{ is the } i^{\text{th}} \text{ component of } \theta \\ -\alpha(s_n^y - s_m^y) z_k^{\frac{\alpha+2}{\alpha}} & \text{when the } k^{\text{th}} \text{ link is between nodes } n \text{ and } m \text{ and } s_n^y \text{ is the } i^{\text{th}} \text{ component of } \theta \\ 0 & \text{else} \end{cases} \quad (7)$$

$$\mathbf{J}[t+1] = \mathbf{D}^{22}[t] - \mathbf{D}^{21}\left(\mathbf{J}[t] + \mathbf{D}^{11}\right)^{-1}\mathbf{D}^{12},$$

where

$$\begin{aligned} \mathbf{D}^{11} &= \mathbf{F}^H \mathbf{W}^{-1} \mathbf{F} \\ \mathbf{D}^{12} &= -\mathbf{F}^H \mathbf{W}^{-1} \\ \mathbf{D}^{21} &= \left(\mathbf{D}^{12}\right)^H \\ \mathbf{D}^{22}[t] &= \mathbf{W}^{-1} + \mathbf{H}[t]^H \mathbf{\Gamma}^{-1} \mathbf{H}[t]. \end{aligned}$$

The PCRB for the covariance of the estimate of the system state at time $t+1$ is then given as $\mathbf{J}[t+1]^{-1}$. In the case where the matrix $\mathbf{H}[t]$ is relatively constant and the system has reached a steady-state, the steady state estimation error can be calculated as the solution for $\mathbf{J}_\infty$ in the Riccati equation:

$$\mathbf{J}_\infty = \mathbf{D}^{22} - \mathbf{D}^{21}\left(\mathbf{J}_\infty + \mathbf{D}^{11}\right)^{-1}\mathbf{D}^{12}.$$

This solution can be found numerically in MatLab with the 'dare' function. The steady-state PCRB is: $\text{PCRB} = \mathbf{J}_\infty^{-1}$.

We apply this PCRB to both the hybrid location tracking algorithm we described previously and the modified multilateration algorithm described in Section III-B. The network (shown in Figure 1) was chosen to be similar to the hexagonal networks previously used [14] to evaluate the Cramér Rao Bound for localization in static networks. In order to avoid edge effects, we only consider the PCRB of the position of a single node located near the center of the network (marked in the figure with an 'x'). We examine the PCRB using both the distance ($\alpha = -1$) and the RSS ($\alpha = 4$) measurement models. Figure 2 shows the PCRB as a function of the observation noise power σ_w^2, and Figure 3 shows the PCRB as a function of the accelerometer noise power σ_η^2. From the plots it is apparent that the hybrid location tracking algorithm outperforms multilateration. When the observation noise power is high relative to the accelerometer noise power, the hybrid location tracking algorithm has a bound relatively independent of the observation noise. When the observation noise becomes small enough relative to the accelerometer noise power, the hybrid location tracking algorithm approaches the bound achieved by multilateration until they are roughly collinear. This means that the hybrid location tracking algorithm is able to take advantage of the low noise power from either measurement source.

The two operating regions are also apparent in Figure 4. This figure contains two plots of the PCRB versus network density. In Figure 4(a), the observation noise is high relative to the accelerometer noise, so the hybrid location tracking algorithm's bound primarily determined by the accelerometer noise, and is relatively independent from network density. This is contrasted with Figure 4(b), where the hybrid location tracking algorithm and multilateration exhibit similar bounds. This once again shows that the hybrid location tracking algorithm is able to take advantage of the relatively higher accuracy available from the acceleration measurements to compensate for the higher accuracy RSS measurements.

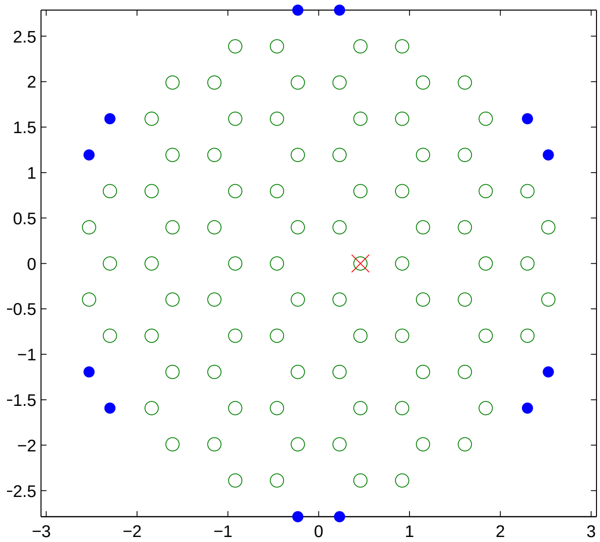


Fig. 1. Network used to estimate PCRB. Reference nodes are solid circles. Mobile nodes are open circles. The open circle containing an 'x' represents the location where the PCRB shown in the following plots was taken.

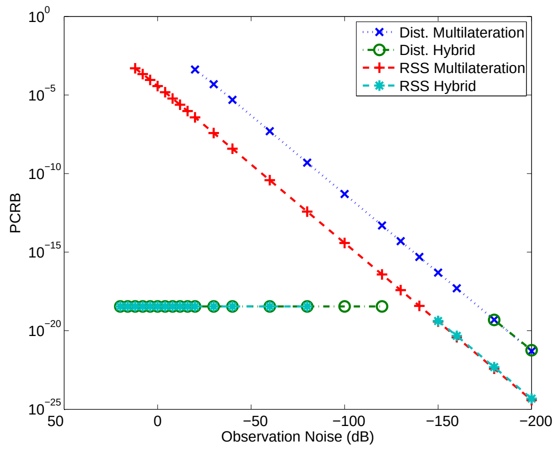


Fig. 2. Steady-state PCRB vs. Observation Noise (σ_w^2 dB)

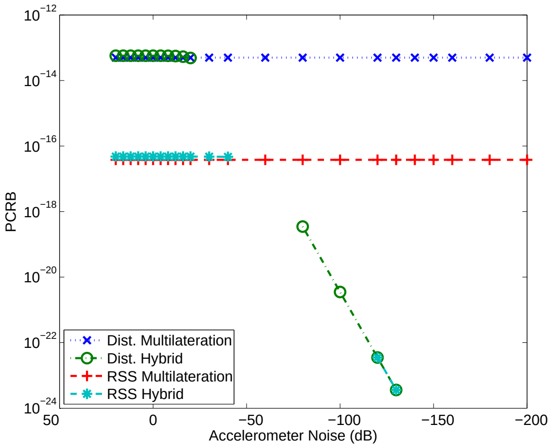
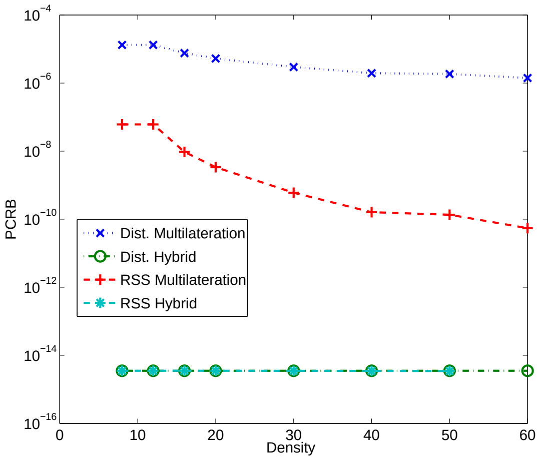
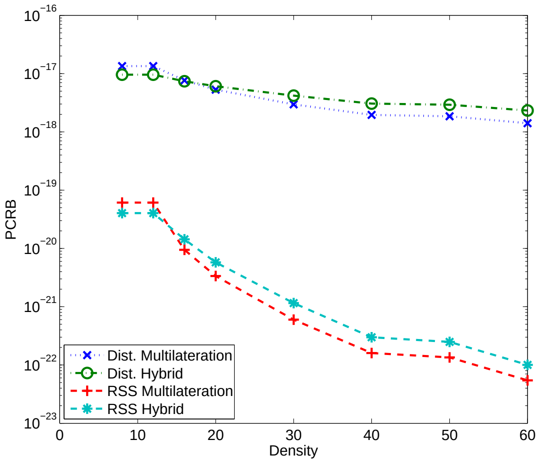


Fig. 3. Steady-state PCRB vs. Accelerometer noise (σ_η^2 dB)



(a) High Observation Noise (-40dB)



(b) Low Observation Noise (-160dB)

Fig. 4. Steady-state PCRB vs. Density (nodes/ sq. unit)

D. Distributed Calculation

The calculation of $\mathbf{K}$ in Eq. (4) for a large network of nodes can be very computationally intensive. To reduce the computational cost, we split the problem and solve each node separately using the estimated positions of the other nodes as reference nodes as in [11]. Then we iterate several times to allow the propagation of information between nodes.

One problem with treating all of the neighboring nodes as if they are reference nodes is that it does not consider the error in the node's position estimate, causing it to underestimate the resulting error in the the location estimate. This location error will cause the Kalman filter to converge to a suboptimal location estimate. We have attempted to remedy this by using a modified calculation of the Kalman gain from Eq. (4):

$$\mathbf{K} = \left(\mathbf{E}\mathbf{H}^H\right)\left(\mathbf{H}\mathbf{E}\mathbf{H}^H + \mathbf{\Gamma} + \mathbf{Q}\right)\right)^{-1},$$

where $\mathbf{Q}$ is a diagonal matrix containing the effective variance in the observed measurement due to uncertainty in the position of this reference node. This can be found to be one of the diagonal elements in that reference node's $\mathbf{H}\mathbf{E}\mathbf{H}^H$ which uses the current node as a reference node. These values can be exchanged by the mobile nodes at the same time they make their measurements.

This method will cause the location estimates to converge faster because it puts more weight on the measurements from reference nodes, whose location is known. Since this method still does not accurately represent the correlation in location error that develops between mobile nodes, this may cause them to overestimate their location error. In our simulations, we have not found this overestimation to be a problem.

IV. Simulation

We simulate our hybrid location tracking algorithm and the modified multilateration location tracking algorithm on a network with $N_m = 60$ mobile nodes and $N_r = 4$ reference nodes to compare the accuracy of the localization and tracking estimates. The mobile nodes are randomly placed in a square region. This region is sized such that the nodes have a density between 1 and 30 nodes per square unit. Each node is assumed to have a communication range of 1 unit, and be capable of making measurements to nodes within this distance. Reference nodes are placed at the 4 corners of the square.

Mobile nodes move according to a movement model similar to the random waypoint model [19]. Since the random waypoint model has been shown to suffer from speed decay, where the average speed of nodes approaches 0 as time progresses [20], we implement a modified version. In our random direction model, nodes select a random direction (uniform from $-\pi$ to π radians), a random speed (uniform from 0 to 1), and two random intervals (both 0 to $T_{\max} = 0.5$s). The first time length describes the duration the node will accelerate from its current velocity to the new velocity. The second describes the duration the node will have constant velocity. The nodes are constrained to lie within the square. Mobile nodes that attempt to leave the square have an appropriate acceleration applied to

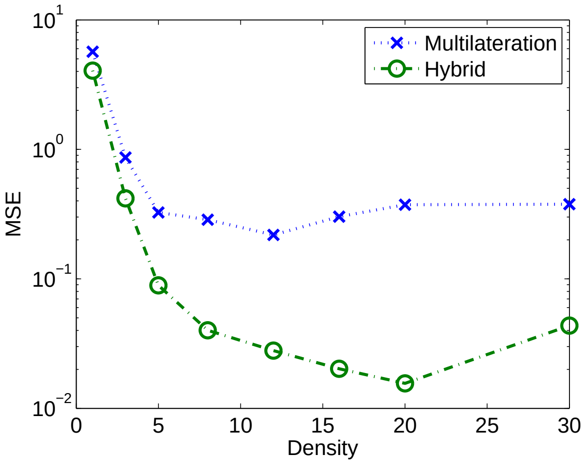


Fig. 5. Position MSE vs Network Density (at 0dB observation noise)

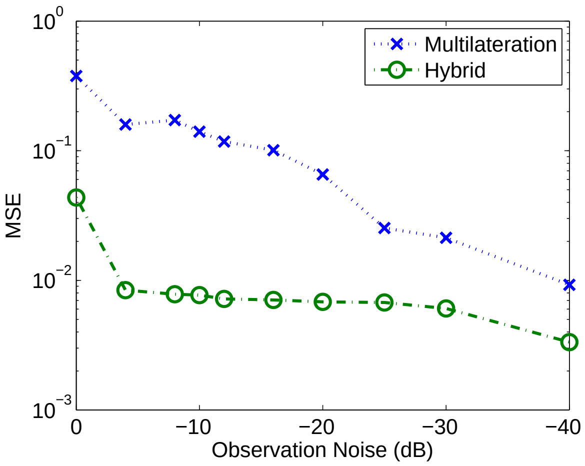


Fig. 6. Position MSE vs Observation Noise (with density of 3)

keep them within the square. Nodes check their acceleration every 0.0001 seconds and synchronize every 0.01 seconds. The simulation is run for 500 synchronization periods to allow it to converge. The results of several simulation runs are combined to estimate the performance at each of several noise levels and network densities.

Figure 5 shows the mean-squared error location tracking performance for 0dB noise power as density increases. Our hybrid location tracking algorithm is able to achieve relatively high performance despite the relatively large noise power. Additionally Figure 6 shows that our tracking algorithm maintains the lower MSE even when the noise power decreases.

V. Conclusion

In this paper we proposed a new hybrid location tracking algorithm based on distance and location measurements. We determined a lower bound on its performance and showed that its bound is lower than that of multilateration for networks with large measurement noise. In simulation we verified that our hybrid location tracking algorithm is able to use the additional acceleration measurements to generate higher accuracy location estimates despite the low distance measurement accuracy common with RSS measurements.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and R. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] N. Patwari, J. N. Ash, S. Kyperountas, A. O. I. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[3] G. Mao, B. Fidan, and B. D. Anderson, "Wireless Sensor Network Localization Techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[4] I. Guvenc and C.-C. Chong, "A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[5] D. A. Tran and T. Nguyen, "Localization In Wireless Sensor Networks Based on Support Vector Machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 7, pp. 981–994, 2008.

[6] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. IEEE/ACM MobiCom*. New York, NY, USA: ACM, 2004, pp. 45–57.

[7] A. Baggio and K. Langendoen, "Monte Carlo localization for mobile wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 5, pp. 718–733, Jul. 2007.

[8] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proc. Third International Symposium on Information Processing in Sensor Networks IPSN 2004*, Apr. 26–27, 2004, pp. 46–54.

[9] V. Seshadri, G. V. Zaruba, and M. Huber, "A Bayesian sampling approach to in-door localization of wireless devices using received signal strength indication," in *Proc. IEEE PerCom*, Mar. 8–12, 2005, pp. 75–84.

[10] C.-H. Chang and W. Liao, "Revisiting Relative Location Estimation in Wireless Sensor Networks," in *Proc. IEEE International Conference on Communications*, Jun. 2009, pp. 1–5.

[11] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proc. ACM Wireless sensor networks and applications*. New York, NY, USA: ACM, 2002, pp. 112–121.

[12] ——, "The n-hop multilateration primitive for node localization problems," *Mobile Network Applications*, vol. 8, no. 4, pp. 443–451, 2003.

[13] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M. B. Srivastava, "On the Error Characteristics of Multihop Node Localization in Ad-hoc Sensor Networks," in *Proc. Information Processing in Sensor Networks*, vol. 2634, 2003, p. 555.

[14] A. Savvides, W. L. Garber, R. L. Moses, and M. B. Srivastava, "An analysis of error inducing parameters in multihop sensor node localization," *IEEE Trans. Mobile Comput.*, vol. 4, no. 6, pp. 567–577, Nov. 2005.

[15] N. Patwari, A. O. I. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, 2003.

[16] S. Park, I. Locher, A. Savvides, M. B. Srivastava, A. Chen, R. Muntz, and S. Yuen, "Design of a wearable sensor badge for smart kindergarten," in *Proc. International Symposium on Wearable Computers*, Oct. 7–10, 2002, pp. 231–238.

[17] D. Niculescu and B. Nath, "Localized positioning in ad hoc networks," in *Proc. IEEE Sensor Network Protocols and Applications*, May 11, 2003, pp. 42–50.

[18] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[19] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, Sep. 2002.

[20] J. Yoon, M. Liu, and B. Noble, "Sound Mobility Models," in *Mobicom*, Sep. 2003.