Vehicle Tracking based on Kalman Filter Algorithm

Tuan Le, Meagan Combs, and Dr. Qing Yang (Computer Science Department at Montana State University)

***Abstract*—Received signal strength indicator (RSSI) is a difficult technique to accurately estimate the distance between two participating entities because of the obscure environmental factors that distort the signal's strength. In this study, we demonstrate that RSSI can be used in combination with the Kalman Filter to identify the position of a node in a wireless vehicular network. By observing a series of measurements and utilizing a model of the node's trajectory, we can filter the noisy RSSI measurements to obtain a more accurate estimation of the node's position. In our experiment, we gathered RSSI measurements from a mobile node, and used this data in combination with the Kalman Filter to estimate the position of the mobile node within 10 feet of the true position. Throughout this report, we demonstrate our implementation of the Kalman Filter, which is conceptually two Kalman Filters condensed into a single filter. Furthermore, we present the results of our experiments that display accuracy as close as 4 ft. from the true position.**

***Index Terms*—Kalman Filter, RSSI, Vehicular Tracking, Localization**

I. Introduction

In a vehicular environment, knowledge of the location of cars and other objects is powerful information that can help prevent accidents, reduce traffic, and lead to overall safer roads. Other applications include: emergency vehicle management, train crossing, tolling, and taxi management. Tracking applications are fundamental to the future of vehicular safety. Currently, the U.S. Department of Transportation is developing DSRC (Dedicated Short Range Communication) as the foundation for a national network among vehicles and roadside access points. For vehicular safety applications, accuracy in vehicles' locations is an essential requirement. As a possible solution, we present our RSSI-based vehicular tracking algorithm built on the Kalman Filter.

The RSSI is a measurement of the power of a radio signal. A main challenge with RSSI ranging is that the effect of reflecting and attenuating objects in the environment can radically distort the received RSSI, making it difficult to infer distance without a detailed model of the physical environment. In our study, we use the Kalman Filter to combat the error inherent within RSSI readings. The Kalman filter is a recursive algorithm that provides an efficient, computational method to estimate the state of a process in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown. The ultimate objective in our study is to use RSSI as the modality for estimating distance in combination with the Kalman Filter to achieve accuracy in multilateration that is viable for vehicular safety.

We performed tests using two wireless sensing devices, one roadside unit (RSU) and one onboard unit (OBU). We moved the OBU in timesteps to simulate a vehicle in motion and obtained the RSSI data received at each position by the OBU. We then used this data along with the Kalman Filter to calculate the position estimates and compared them to the actual path of the device to assess the accuracy of our calculations.

II. Background

There is an array of RF (Radio Frequency) signal properties used to determine the distance between a transmitter and receiver. The most commonly used and explored techniques are RSSI (Received Signal Strength Indicator), TOA (Time of Arrival), TDOA (Time Difference of Arrival), and AOA (Angle of Arrival) [4]. This study utilizes the RSSI of received packets because of its tractability. RSSI is a measure of the voltage or power received by the antenna, reported in dBm. The salient feature of RSSI is that it does not require additional hardware, software, or computational cost; however, in exchange for ease of use, the RSSI is extremely susceptible to noise as reported in literature and supported by our empirical tests. This noisiness in the RSSI measurements is a result of the fading and shadowing of the RF signal, which is caused by the signal's surroundings [6]. More explicitly, fading and shadowing encompass the reflection, refraction, diffraction, and obstruction of the RF signal. Furthermore, multipath propagation explicates that the observed signal at the receiver is the summation of these reflected, refracted, and diffracted RF signals, distorted by the environment; these signals can add constructively or destructively, skewing the expected power of the received RF signal. Ultimately, all these effects make the RSSI measurement an erroneous indication of the distance between the transmitter and receiver.

A. RSSI-Based Ranging Model

The path-loss channel model below is the most commonly used model to correlate the received radio signal strength (RSS) to the distance between the transmitter and receiver [5].

$$P_r = P_0 - n \log \frac{d}{d_0} + x \tag{1}$$

P_r represents the RSSI. d_0 and P_0 are the reference distance and the received signal strength at that reference distance respectively. x is a random variable that accounts for the effects of shadowing and fading. The fading of the signal envelope has been shown to generally adhere to the Nakagami distribution in [3]; however the Kalman filter algorithm requires a Gaussian distribution to model measurement errors and white noise. Eq. (1) can be generalized with empirical constants, replacing the logarithmic reference distance terms.

$$g(d) = P_r = A - n \log d - nB \tag{2}$$

In this form, Eq. (2), the generalized path-loss channel model, can be fitted to a dataset of distance-RSSI pairs, solving for the parameters, A, B, and n, that best characterize the path loss of the data. The parameters can be found by solving the least squares regression equation, similar to the method used in [5].

$$\min_{A,B,n} \sum_{i} (P_{ri} - g(d))^2 \tag{3}$$

i is a sample distance-RSSI pair in our dataset. We applied gradient descent to (3) to solve for the optimal parameters.

B. Multilateration

The goal of multilateration is to determine the position of a node, which, in this context, is a mobile vehicle. We will refer to the mobile vehicle as the mobile node. The idea is to employ RSUs to send messages containing each RSUs absolute coordinates to the mobile node. The mobile node will receive these coordinates along with a corresponding RSSI to calculate an estimate of the distance between the mobile node and the RSU. The localization algorithm we use requires three or more RSUs to calculate an estimate of the mobile node's position.

In a wireless sensor network (WSN), there are stationary anchor nodes with known positions; these are the RSUs. The distance between the mobile node and an anchor node can be determined using the path-loss channel model, Eq. (2). Furthermore, in our application, we will be using multiple distances between a mobile node and a set of anchor nodes to determine the absolute location of the mobile node via multilateration algorithms.

A common and intuitive method to determine the position of the mobile node given the measured distances between the mobile node and 3 or more anchor nodes is to calculate the position on the coordinate system that minimizes the least squares equation. To estimate the position, we minimize the square of the difference between the measured distances and the calculated distances, where the calculated distances are the distances between the possible position and the anchor nodes. [2] utilizes the equation below to estimate the position of the mobile node.

$$l(r_1, ..., r_i) = min_{x,y} \sum_{i} ((x - x_i)^2 + (y - y_i)^2 - r_i^2)^2 \tag{4}$$

We employ gradient descent again to determine the position of the mobile node. Unfortunately, the gradient descent algorithm requires an initial state for the parameters, and, because Eq. (4) contains local minima, the algorithm must be executed multiple times with varying initial states to attain the global or near global minimum.

C. The Kalman Filter

The Kalman Filter is a recursive filtering algorithm that is used to optimally estimate the current state of a process in the presence of noisy measurements by minimizing the mean of the squared error. The Kalman Filter is commonly applied and researched in the area of navigation [7]. We extend upon our multilateration method and apply this algorithm to track the position of a mobile node. Whereas multilateration intends to estimate the position at a single point in time, the Kalman Filter goes beyond that single observation and makes an optimal estimate of the position based on a sequence of localization measurements and an underlying model of the system, i.e. the mobile nodes trajectory. The goal of the Kalman Filter algorithm is to estimate the state of a discrete-time controlled process or system represented by $x \in \Re^n$. The Kalman Filter stipulates that the underlying process must be modeled by a linear dynamical system and that the measurements and the error terms express a Gaussian distribution [7] [8]. The state of x is governed by the following linear stochastic difference equation [7].

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{5}$$

The measurement term, $z \in \Re^m$ is expressed as

$$z_k = Hx_k + v_k \tag{6}$$

The additional terms, w and v are random variables that represent the process and measurement noise respectively. They are assumed to be zero-mean Gaussian random variables, and their variances are defined by the covariance matrices, Q and R respectively. A, the transition matrix, is a $n \times n$ matrix that correlates the previous state to the current state. The term Bu_{k-1} is the optional driving function where u_{k-1} represents control inputs and B is a matrix that relates u_{k-1} to the x_k. H is a $m \times n$ matrix that relates the current state, x to the observed measurement at time k.

The Kalman Filter is a feedback control process that loops through two stages: time update and measurement update [7]. During each loop, the time update step predicts the next state using the model of the system; this step's prediction is called the *a priori*, denoted by $\hat{x}_k^-$. In the same loop iteration, the measurement update then accepts a new measurement and integrates it into the *a priori*, resulting in an improved estimation of the current state called the *a posteriori* as denoted by $\hat{x}_k$ [7]. Moreover, both the *a priori* and *a posteriori* have estimate errors that are defined by their covariances. These covariances are labeled P_k^- and P_k. The equations below characterize the time update step of the Kalman Filter.

$$\hat{x}_k^- = A\hat{x}_{k-1}^- \tag{7}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{8}$$

P_k^- represents the covariance of the *a priori*. The measurement update step is described below.

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{9}$$

$$\hat{x}_k = \hat{x}_{k-1} + K_k(z_k - H\hat{x}_k^-) \tag{10}$$

$$P_k = (I - K_k H)P_k^- \tag{11}$$

The terms A, B, u, Q, and R are the same terms that are expressed in Eq. (5) and (6).

The Kalman Filter can further be extended with the Kalman Smoother. Whereas the Kalman Filter works recursively forward in time, improving the estimate of the current state based on the previous state, the Kalman Smoother utilizes the future time-step states to improve the previous states.

Fig. 1. The figure displays the OBU used in our experiment. More information can be found at http://www.aradasystems.com/locomate-obu/.

Fig. 2. The figure displays the RSU used in our experiment. More information can be found at http://www.aradasystems.com/locomate-rsu/.

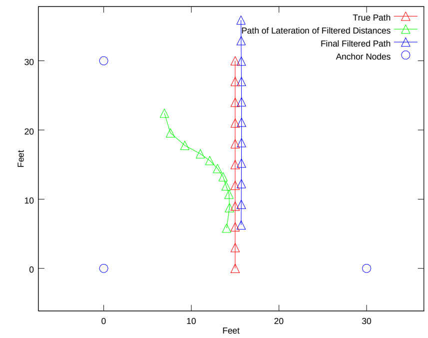

Fig. 3. The figure displays the local coordinate system used in the experiment. The red path represents the true path of the mobile node. The green path is the localized filtered distance estimates at each time step. The blue path is the final filtered path that encompasses information from both the filtered distances and the mobile node's trajectory.

III. Experimental Setup

We designed an experiment to gather RSSI readings received at the OBU from the RSUs, and we use this data in combination with the Kalman Filter to estimate the node's position. We used two devices made by Arada, the Locomate Classic On Board Unit and the Locomate Road Side Unit, both equipped with a full DSRC WAVE software solution. The OBU functioned as a mobile node that would be placed in a vehicle and the RSU functioned as an anchor node. The RSSI data was always collected by the OBU.

The experiment was conducted outdoors in an open parking lot, assumed line of sight. We used a 30 by 30 ft. grid to help with localization precision. In a vehicular environment, any vehicle being localized would be in motion more often than not, so to simulate a vehicle in motion we moved the OBU along a specified path and gathered RSSI data in 3-ft. increments. Each increment represented a time step as if the device were moving. For each 3-ft. increment we gathered 500 samples of RSSI measurements from each anchor node. We collected this much data because the fading, attenuation and multipath propagation causes the RSSI to fluctuate, and taking the mean of 500 RSSI values gave us the most precise RSSI for each position.

Our experiment was limited in that we only had one RSU, whereas in a more realistic vehicular environment many RSU would be present. In order to emulate this, we placed the RSU in three different positions: (0,0), (30,0), and (0,30) and gathered RSSI data from each anchor node position every time the OBU was moved. With the RSSI data gathered from the anchor node positions, we used the path loss model to estimate the distance based on the RSSI alone. Using the aggregate anchor distance estimates, we could estimate the position of mobile node using (4).

IV. Application of the Kalman Filter

In our application of the Kalman Filter, our underlying model of the mobile nodes trajectory is driven by the nodes velocity. Here, we make the assumption that the mobile vehicles velocity is accessible by our algorithm. This is not such an unfair assumption because it is fairly plausible for future vehicles to have embedded computer systems with velocity sensor readings available to them. In each of our applications of the Kalman Filter, we also utilized a Kalman smoother to further improve the position estimates.

Our initial attempt to apply the Kalman Filter was unsuccessful. We implemented a Kalman Filter with a simple kinematic model that described the trajectory of the mobile node. We filtered on the mobile node's position using measurements from the multilateration algorithm. However, our implementation proved to be ineffective.

Our second application of the Kalman Filter is a two-pass filter that performs two executions of the Kalman Filter. The first Kalman Filter is used to rarefy the distance estimates given by each anchor node. Once the filtered distances are obtained, our multilateration algorithm calculates the optimal position at each time step, creating an estimation of the mobile

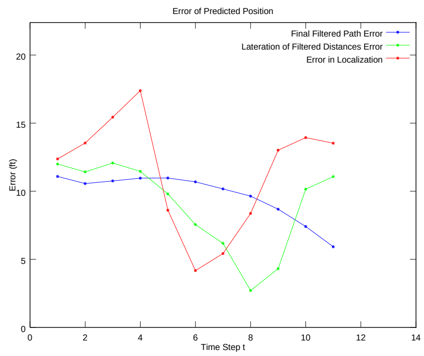

Fig. 4. The figure displays the errors of the steps of Kalman Filter implementation. The red path represents the error in multilateration of the crude distances received without any filtering. The green path displays the error in the multilateration of the filtered distances without knowledge of the true trajectory model. The blue path displays the error in the final filtered path which uses both the filtered distances and a model of the trajectory.

nodes path. These new positions are then passed to our first application of the Kalman Filter to then further refine the path.

The idea behind our second application is to first improve the distance estimates before using them for multilateration. In our first application, we filtered on the mobile node's position. However, because the distance estimates are inherently noisy, the multilateration algorithm combines the errors of the distances and produces an inaccurate estimate of the mobile node's position. Therefore, to counteract this issue, we decided to implement a two-pass Kalman Filter that would refine the distance estimates before multilateration. Our third and final version of our Kalman Filter tracking implementation is a minor improvement of the second in terms of efficiency; we compacted the two-pass Kalman Filter into one to improve the efficiency and accuracy of the algorithm.

Our final implementation models the system state as

$$x = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ x \\ y \\ v_x \\ v_y \end{pmatrix} \tag{12}$$

Where r_1, r_2, r_3, are the estimated distances between the anchor nodes and the mobile node. The state size can be increased to incorporate more anchor nodes. x and y denote the estimated position of the mobile node, and v_x and v_y represent the velocity.

The Kalman Filter is bifurcated into the time update step and the measurement update step. In the time update step, we predict the distances and the position while the velocity is assumed to be constant in this stage. The anchor node distances can be modeled with a time-discrete formula by taking the derivative of r_i with respect to time. The model can then be linearized as a linear combination of the velocity and current distance. Our model for distance is displayed below.

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{13}$$

$$a = \frac{(x - x_i)}{d_i} \tag{14}$$

$$b = \frac{(y - y_i)}{d_i} \tag{15}$$

$$\frac{dr_i}{dt} = av_x dt + bv_y dt \tag{16}$$

$$r_i(k) = r_i(k-1) + \frac{dr_i}{dt} \tag{17}$$

Here $r_i(k)$ denotes x's r_i at time k. The i subscript distinguishes between the anchor nodes' positions and their corresponding distances. In the same fashion, the position can be time-discretized into a linear combination of the current position and the velocity.

$$x_k = x_{k-1} + v_x dt \tag{18}$$

$$y_k = y_{k-1} + v_y dt \tag{19}$$

With Eq. (17), (18), and (19), the transition matrix, A can be formed as

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & (a_1)dt & (b_1)dt \\ 0 & 1 & 0 & 0 & 0 & \vdots & \vdots \\ 0 & 0 & 1 & 0 & 0 & (a_3)dt & (b_3)dt \\ 0 & 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{20}$$

The processing noise matrix, Q is set to a diagonal matrix, where the values on the diagonal are relatively low because we are confident in the prediction. j is simply a scaling factor to adjust Q.

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .1 \end{pmatrix} j \tag{21}$$

In the measurement update step, the measured distances are incorporated into the current estimate of the system state to produce the *a posteriori*. At this step, we also measure the velocity and calculate an estimate of the current position based

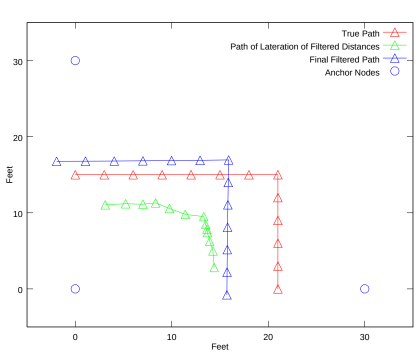

Fig. 5. The figure displays the local coordinate system used in the second experiment. An alternate path is implemented to test the viability of the Kalman Filter algorithm on a more complex path.

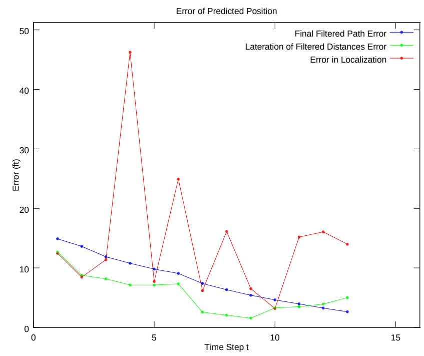

Fig. 6. The figure displays the errors of the steps of second experiment. The red path represents the error in multilateration of the crude distances received without any filtering. The green path displays the error in the multilateration of the filtered distances without knowledge of the true trajectory model. The blue path displays the error in the final filtered path which uses both the filtered distances and a model of the trajectory.

on the *a priori* distances using our localization algorithm, Eq. (4). This results in the measurement vector, h.

$$h = \begin{pmatrix} r_{1m} \\ r_{2m} \\ r_{3m} \\ x = l(r_1, r_2, r_3) \\ y = l(r_1, r_2, r_3) \\ v_x \\ v_y \end{pmatrix} \tag{22}$$

The measurement noise, R, is dynamically set to contain the variance of the distance samples for each anchor node. The noise of the measured position is assumed to be constant across all time steps, and the noise of the velocity is set to zero because the velocity is observed directly and assumed to be correct.

$$R = \begin{pmatrix} Var[r_{1m}] & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Var[r_{2m}] & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & Var[r_{3m}] & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{23}$$

V. Analysis and Results

Fig. 3 and 4 display the results of our application of the Kalman Filter algorithm in our first experiment. Fig. 3 is a graphical display of a position coordinate system while Fig. 4 presents the error of our Kalman Filter implementation. From both figures, it is evident that the implementation is compellingly accurate in the presence of noisy data on our initial experiment. The unfiltered estimated path from the multilateration of the distance estimates, i.e. the green path, does not encompass knowledge about the true mobile node's path; the green path is the lateration of the filtered distances, and the system model of the filtered distances, Eq. (17), does not directly relate to the mobile node's path. Consequently, the shape of the resulting path is not reflective of the true path. In comparison, the final filtered path is a more accurate representation of the true path's shape. The final filtered path incorporates both the refined distances and the position model to improve the accuracy of the estimated path. In all of our experiments, the error in the final time step is within 10 ft. which is significantly accurate in comparison to the error in the multilateration on the raw distances as represented by the red line in the Fig. 4, 6, and 8. Moreover, the complex paths of Fig. 5 and 7 display accuracy within 5 ft. of the true path by the final timestep.

From Fig. 3, 5, and 7, we can understand how the Kalman Filter is working. In each figure, the lateration of the filtered distances results in more accurate distance estimates that are within reasonable distance of the true path. The final filtered path understands the shape of the trajectory through the position model used in our transition matrix, (20). As a result, the green path's locality to the true path is further rarefied by the filtered positions to shapen the estimated path into a trajectory that is closely similar to the true path. According to our results, this conceptual two-step procedure of the refinement of the

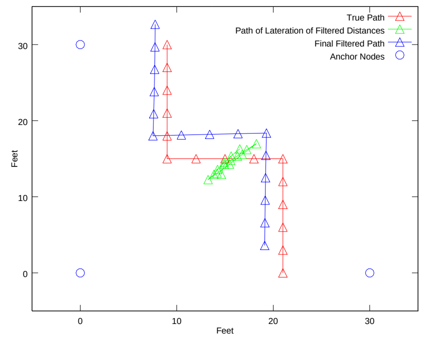

Fig. 7. The figure displays the local coordinate system used in our third experiment. We again test on a complex path.

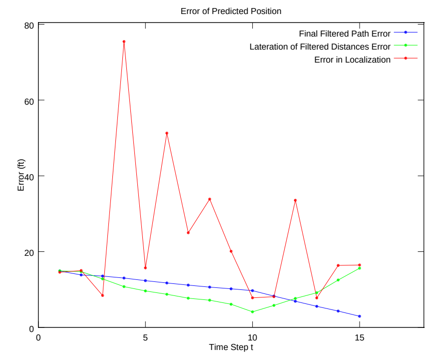

Fig. 8. The figure displays the errors of the steps of second experiment. The red path represents the error in multilateration of the crude distances received without any filtering. The green path displays the error in the multilateration of the filtered distances without knowledge of the true trajectory model. The blue path displays the error in the final filtered path which uses both the filtered distances and a model of the trajectory.

distances before positions is evidently an effective algorithm to combat the noisiness of the RSSI measurements.

VI. Conclusion

In conclusion, we have implemented a compelling tracking algorithm for vehicular or mobile tracking using the Kalman Filter. We utilized RSSI measurements as our medium for estimating the distance between a mobile node and an anchor node. With a set of three or more distance measurements, the position of the mobile node can be determined through various multilateration algorithms. In our experiment, we solved the linear least squares regression equation to choose the position that minimized the summation of the squared differences between the distance measurements and the estimated distances. Our initial Kalman filter implementation filtered solely on the multilaterated positions, determined from the raw distance measurements of the anchor nodes. As a result, the implementation proved to be fairly inaccurate. Therefore, we decided to filter on the measured distances before passing them to our multilateration algorithm. By implementing this change, the error in the path was diminished down to below 10 ft. which is considerably accurate compared to multilateration on the raw distance measurements alone. Ultimately, while our experiments are not representative of true tracking (because of the lack of true mobility and the unrealistic number of RSSI samples at each time step), the accuracy of our implementation of the Kalman Filter is compelling enough to warrant more research on this approach. Further testing is needed to truly test the accuracy of our implementation in a vehicular setting. It is important to note that our implementation and research are very extensible. The implementation of our Kalman Filter algorithm can be improved in several areas. First, we utilized RSSI in our approach because it was an indicator of distance that was very easy to use. However, the implementation in this report is by no means limited to the RSSI medium; in fact, the algorithm's accuracy can drastically be improved with a more accurate indicator of distance. Also, the extended Kalman Filter or the unscented Kalman Filter both permit the use of non-linear prediction models. Therefore, integrating more complex and accurate prediction models can further improve the accuracy of tracking. Furthermore, this research is extensible in a multitude of ways that may reinforce or reject its viability. Next steps include testing in a real vehicular environment, testing the correlation between the number of anchor nodes used and the accuracy of tracking, and, finally, researching a dynamic implementation of our algorithm to dynamically handle the addition and loss of anchor nodes in the mobile node's proximity.

Appendix
Proof of the Distance Prediction Model

The distance prediction model is the model we use to predict the subsequent distance using the current estimated distance and the current position. The current position represents the most accurate estimate that minimizes the mean squared error of the Kalman Filter. Let r_i denote the current distance estimate, let d_i denote the distance between the current position estimate and anchor node, i, and let x_i and y_i denote the coordinates of the anchor node. If you conceptualize a circle with radius r_i around anchor node i, the closest point on that circle to the current position estimate can be related to the current distance, r_i. The following formula defines the relationship between the closest point and r_i.

$$r_i = \sqrt{(\frac{r_i}{d_i}(x - x_i))^2 + (\frac{r_i}{d_i}(y - y_i))^2} \tag{24}$$

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{25}$$

If the position of the closest point is kept constant in relation to the current position, then any translation of the current position will also affect the closest point, and consequently, the current distance estimatation, r_i. Therefore, by time-discretizing Eq. (25), we can linearly approximate the next distance.

$$\frac{dr_i}{dt} = \frac{(x - x_i)}{d_i} v_x dt + \frac{(y - y_i)}{d_i} v_y dt \tag{26}$$

$$r_i(k) = r_i(k-1) + \frac{dr_i}{dt} \tag{27}$$

In this form, the distance model can be arranged within the transition matrix of the Kalman Filter as a linear combination of the current state. Eq. (26) is not truly a linear combination, but instead, in our implementation, we abstract the multiplying terms of v_x and v_y as coefficients. This provides a simple solution for our prediction model. However, an issue with this simplification of the distance model is that the uncertainties in x and y are not accurately integrated into the resulting covariance matrix, P_k. Although Eq. (27) is a function of the current position, x and y, it is simplified into a linear combination that does not reflect the *a priori*'s dependency on the current position. Therefore, a more correct solution would be to implement an extended Kalman Filter that is capable of handling non-linear equations.

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] C. Papamanthou, F. P. Preparata, and R. Tamassia, "Algorithms for location estimation based on RSSI sampling" in *Algorithmic aspects of wireless sensor networks*, S. P. Fekete, Verlag, Germany: Springer, 2008, pp. 72-86.

[3] J. Yin, G. Holland, T. Ebatt, F. Bai, and H. Krishnan, "DRSC channel fading analysis from empirical measurement," Proc. Ist IEEE Int'l Workshop Vehicle Comm &. Apps, Oct. 2006.

[4] X. Zhou, M. Li, and F. He, "Research on the optimization of wireless sensor network localization based on real-time estimation," in *Proc. of the 2012 Int'l Conference on Information Technology and Software Engineering*, 2012, pp. 23-31.

[5] G. Blumrosen, B. Hod, T. Anker, D. Dolev, B. Rubinsky, "Enhanced calibration technique for RSSI-based ranging in body area networks," in *Ad hoc networks*, vol. 11, (1), pp. 555-569, 2013.

[6] M. Rossi, "RSSI based tracking algorithms for wireless sensor networks: theoretical aspects and performance evaluation," M.S. Thesis, University of Padua, Padua, Italy, 2007.

[7] G. Welch, and G. Bishop, "An introduction to the kalman filter," 1995.

[8] J. Hartikainen, A. Solin, S. Srkk, "Optimal filtering with kalman filters and smoothers: a manual for the matlab toolbox ekf/ukf," 2011.