Performance Comparison of Nonlinear Filters for Indoor WLAN Positioning

Hui Wang, Andrei Szabo, Joachim Bamberger
Learning Systems
Information and Communications, Corporate Technology
Siemens AG
Munich, Germany
Email: {hui.wang.ext,andrei.szabo,joachim.bamberger}@siemens.com

Dietrich Brunn, Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute of Computer Science and Engineering
Universität Karlsruhe (TH)
Germany
Email: brunn@ira.uka.de, uwe.hanebeck@ieee.org

***Abstract*—Indoor WLAN positioning should be modeled as a nonlinear and non-Gaussian dynamic system due to the complex indoor environment, radio propagation and motion behaviour. The aim of this paper is to analyze different filtering strategies for real life indoor WLAN positioning systems. The performance criteria for the comparison are the mean of localization errors and computational complexity. Three nonlinear filters are analyzed: Fourier density approximation (FF), particle filter (PF) and grid-based filter (GF), which are representatives for deterministic and random density approximation approaches. Our experimental results help to choose the appropriate filtering techniques under different resource limitations.**

Keywords: Indoor Positioning, Nonlinear Filtering.

I. Introduction

Indoor WLAN positioning system has attracted a lot of research interests from both academia and industry in recent years [1] [2] [3]. Mathematically, it can be formulated as a state estimation problem. A system model describes the state evolution with time, i.e., the motion of mobile devices. A measurement model describes the noisy received signal strength (RSS) observations. Due to the complexity of indoor environment, indoor radio propagation is highly nonlinear, which results in a nonlinear measurement model. In addition, the motion of a mobile device cannot be modeled linearly in general. Therefore, we regard the indoor WLAN positioning system as a nonlinear and non-Gaussian system.

Bayesian filtering provides a general recursive framework to estimate the posterior probability density function of the state given all the available observations. But its optimal analytic solution with respect to an optimality criterion is tractable only in some special cases. For instance, if the system has a linear structure and the posterior density is kept as Gaussian, the optimal solution with respect to minimal posterior mean square error can be derived using the well-known Kalman filter [4]. If the state space is discrete and has a limited size, the grid-based method is optimal [5]. For most nonlinear and non-Gaussian systems like indoor WLAN positioning system, it is impossible to get the exact analytic solution due to the fact that the complexity of the posterior probability increases after each step of the recursive process. To solve this problem, various suboptimal nonlinear filters have been proposed. These filters approximate either the system and measurement model or the probability density function. The first family of filters including the extended Kalman filter (EKF) [5] and its variants approximate the nonlinear model by its local linearization. The other family consists of numerous nonlinear filters that aim to approximate the density function by a number of parameters. For instance, grid-based methods use uniformly sampled grid points to approximate the density functions. And Unscented Kalman filter (UKF) [6] matches the first two moments of the posterior distribution density of the state vector conditioned on the observations by using sample points of the Gaussian-Hermite Quadrature approximation to the integrals. Besides, particle filters [5] randomly generate a collection of particles to do the same work. Recently, researchers also represent the density function by the linear combination of kernel functions, e.g., Gaussian mixtures [7], Fourier densities [8], Dirac mixtures [9] and so on.

The performance of various nonlinear filters are closely tied to the properties of specific applications. The system and measurement model, the noise characteristics, and the system requirements for speed or memory consumption are the key factors to decide which filter is most suitable. In this paper, we compare the performance of three nonlinear filters, namely the filter based on Fourier density approximation (FF), particle filter (PF) and grid-based filter (GF) for our indoor WLAN positioning system. We choose these three filters as candidates because they all can handle nonlinear and non-Gaussian systems with arbitrary complexities by approximating density functions with any required accuracy. The key performance criteria we use are the mean localization error and computation time, which reflect the accuracy and computational efficiency, respectively. We evaluate the filters with both simulated and real measurements, aiming to find the most appropriate one for practical indoor WLAN positioning systems. Our experiments indicate that particle filters are the most computationally efficient. Fourier-based filters are superior with respect to using the fewest components to approximate the density functions accurately.

The remainder of this paper is organized as follows: In Section II, three nonlinear filters to be evaluated are briefly described. In Section III, we introduce the usage of nonlinear filter in indoor WLAN positioning systems. The performance of different filters in our testbed are given and discussed in Section IV. Finally, Section V concludes the paper.

II. Description of Nonlinear Filters

The discrete-time stochastic dynamic system with additive noises is usually modeled by

$$\boldsymbol{x}_{k+1} = a_k\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) + \boldsymbol{w}_k, \tag{1}$$

$$\boldsymbol{y}_{y+1} = h_{k+1}\left(\boldsymbol{x}_{k+1}\right) + \boldsymbol{v}_k. \tag{2}$$

Formula (1) models the system process which updates the current state vector $\boldsymbol{x}_k$ given an input $\boldsymbol{u}_k$ and system noise $\boldsymbol{w}_k$. Formula (2) describes the measurement process which relates the state $\boldsymbol{x}_{k+1}$ to the observation $\boldsymbol{y}_{k+1}$, corrupted also by noise $\boldsymbol{v}_k$.

The Bayesian approach provides a recursive way to estimate the hidden state of dynamic systems with the above form. It has also two steps: prediction step and update step as

$$f_{k+1}^{p}\left(\boldsymbol{x}_{k+1}\right) = \int f_k^T\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{u}_k\right) f_k^e\left(\boldsymbol{x}_k\right) \mathrm{d}\boldsymbol{x}_k, \tag{3}$$

$$f_{k+1}^{e}\left(\boldsymbol{x}_{k+1}\right) = \frac{1}{c_k} f_{k+1}^{L}\left(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}\right) f_{k+1}^{p}\left(\boldsymbol{x}_{k+1}\right), \tag{4}$$

where $f_{k+1}^{p}\left(\boldsymbol{x}_{k+1}\right)$ is the predicted density at time k. $f_{k+1}^{T}\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{u}_k\right)$ is the transition density, which is given by

$$f_{k+1}^{T}\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k, \boldsymbol{u}_k\right) = f_k^w\left(\boldsymbol{x}_{k+1} - a_k\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\right), \tag{5}$$

where $f_k^w\left(\cdot\right)$ is the density of the system noise at time k. $f_k^e\left(\boldsymbol{x}_k\right)$ is the posterior density function at time k. $f_{k+1}^{L}\left(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}\right)$ is the conditional likelihood density given by

$$f_{k+1}^{L}\left(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}\right) = f_{k+1}^{v}\left(\boldsymbol{y}_{k+1} - h_{k+1}\left(\boldsymbol{x}_{k+1}\right)\right), \tag{6}$$

where $f_k^v\left(\cdot\right)$ is the density of of the measurement noise at time k.

The Kalman filter can solve the above equation analytically if the system is linear and the posterior density is kept as Gaussian. For most of the other nonlinear and non-Gaussian systems, the analytic solution to formula (3) and (4) is intractable because the complexity of density functions increases over time. One way to get the suboptimal solution is to approximate the density function by a sum of base functions as

$$f\left(\boldsymbol{x}\right) \approx \sum_i w_i \psi_i\left(\boldsymbol{x}, \boldsymbol{\gamma}_i\right), \tag{7}$$

where w_i is the weight of component i and $\boldsymbol{\gamma}_i$ is the parameter vector that controls the shape or location of components. With different forms of components and ways to generate them, various nonlinear filters are utilized. In this paper we choose three representative nonlinear filters for evaluation. In the following, the major characteristics of these three filters are briefly introduced. The detailed algorithms are beyond the scope of this paper and can be found in the references.

A. Fourier-based filter

Fourier series were first employed to estimate probability densities in [14]. Recently, [8] ensured the non-negativity of Fourier series by approximating the square root of the density instead of the density itself. The density function $f\left(\boldsymbol{x}\right)$ is approximated by its truncated Fourier series expansion as

$$f\left(\boldsymbol{x}\right) = \sum_{\boldsymbol{i}} w_{\boldsymbol{i}} e^{j\boldsymbol{i}^T\boldsymbol{x}}. \tag{8}$$

To ensure nonnegativity, the absolute square root of Fourier series is exploited. In addition, by Fourier series, the expressions in formulas (3) and (4) can be calculated analytically while the type of density function is preserved [8]. Because of the orthogonality of Fourier expansions, the coefficients of components can be calculated analytically. In practical implementation, it is done by the Fast Fourier Transform (FFT) algorithm. Additionally, the number of components, i.e., the complexity of calculation, can be adjusted optimally. Because the importances of components are sorted directly by the absolute value of coefficients with respect to a density distance measure, component reduction is very efficient. However, the Fourier transformation by FFT usually requires a fine sampling resolution in the whole state space. This results in high computational effort and memory consumption especially for high-dimensional density functions.

B. Particle filter

Particle filters represent the density by a series of randomly selected particles. The well-known Sequential Importance Sampling (SIR) particle filter consists of the following steps.

1. Generate a new particle for each old particle randomly using the motion model.
2. Weight the new particle using the measurement model, i.e., likelihood densities.
3. Resample the particles, delete the particles with small weight, and split the particles with large weights. Finally, each particle has the same weight.

The advantage of the particle filter is that it is easy to be implemented and it simplifies the density integrals. With the increasing number of particles, the filter can also handle arbitrary density functions. However, although particle filters can beat the curse of dimensionality in terms of convergence [11], they do not beat the curse with respect to computational complexity [12]. Furthermore, the rate of convergence is not uniform for dynamic systems [13]. This implies that it is difficult to evaluate the quality, i.e., the number of needed samples, of such estimators. Additionally, since particle filters are random approaches, the final result varies especially when the number of particles is small.

C. Grid-based filter

In grid-based method, the density is sampled at uniformly distributed grid points. The integral in Bayesian estimation equation is replaced by a discrete summation. Like the other two filters, grid-based filter can also approximate arbitrary density functions with increasing resolution of grid points.

But like the Fourier filter, the grid resolution is the same in the whole state space. This leads to a high computational effort and memory consumption if a fine sampling or high dimensionality is required.

Figure (1) illustrates the approximation of a Gaussian mixture density with three components by a Fourier-based filter, a particle filter, and a grid-based filter respectively. Theoretically, all these three filters converge to the optimal Bayesian solution when the number of components increase towards infinity. But in practical systems it is very important to choose a proper filter that has a fast converging rate, to get an accurate result with a short computational time.

Two factors affect the computational efficiency. One is the number of components that can accurately approximate the density function. This factor depends highly on the characteristics of specific system. The other one is the computation time given different number of components. Table I lists how the computation time of each step increases with the number of components for the above three filters. We see obviously that particle filter tends to be most efficient when the number of components is large. Furthermore, we can notice that the efficiency of Fourier-based filter is both affected by the number of components of FFT sampling and the reduced number of components.

III. Nonlinear Filtering in indoor WLAN positioning

The indoor positioning system we are working on is based on the received signal strength (RSS) of WLAN system. In a typical scenario, a number of stationary access points (APs) are distributed in an indoor environment. Mobile devices measure the RSS signals, which are used to estimate the position of mobile devices. In mathematic language, the RSS signals are noted as $\boldsymbol{p} = \left[p^1, p^2, \ldots, p^N\right]$ where p^i is the RSS measurement from AP i and the position of mobile device is noted as $\boldsymbol{x} = [x, y, z]$. The function $h_k(\cdot)$ is used to describe the relationship between the position and the RSS at time k as

$$\boldsymbol{p}_k = h_k(\boldsymbol{x}_k) + \boldsymbol{v}_k, \tag{9}$$

where $\boldsymbol{v}_k$ is the additive noise caused by the complicated indoor environment.

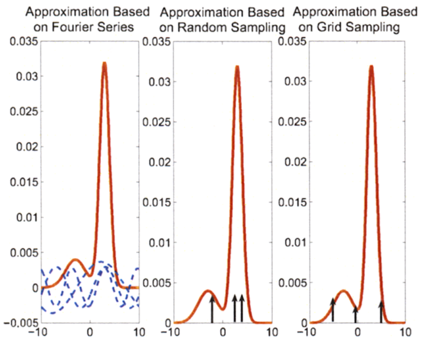

Figure 1. Illustration of density estimation with 3 components.

Table I
Computational Complexity Analysis

Fourier-based Filter	
Transition density approximation by m components (FFT)	$O(m \log m)$
Reduce to n components	$O(m \log m)$
Prediction	$O(n)$
Likelihood density approximation by m components (FFT)	$O(m \log m)$
Reduce to n components	$O(m \log m)$
Update	$O(n \log n)$
Particle Filter	
Prediction	$O(n)$
Update	$O(n)$
Resampling	$O(n)$
Grid-based Filter	
Prediction	$O(n^2)$
Update	$O(n)$

The first problem is how to model the mapping function $h_k(\cdot)$. This function may be determined using a parametric radio propagation model, e.g., multi-wall model or dominant path model [3]. But the performance is not always satisfactory, due to the fact that the radio propagation is highly affected by many field-specific parameters such as walls, doors and so on. These parameters are sometimes difficult to retrieve within the models or even not known to the user. Fig. 2 shows the building map of our test environment and a radio distribution map in this environment. We can see that the building structure has a clear influence to the radio distribution. Additionally, RSS noise has also a complex form. As Fig. 3 shows, RSS noises at reference points have various and complicated forms which can hardly be modeled by a simple Gaussian. In practical systems, people usually model the mapping function and noise in a non-parametric way, i.e., measuring the RSS distribution offline or online at reference points to build a non-parametric radio map or mapping function.

Another problem is how to model the motion of mobile devices. Unlike the car navigation system, the indoor motion is usually hard to predict because the motion direction and distance vary largely. With the assistance of extra sensors like accelerometer or gyroscope, the motion behaviour can be

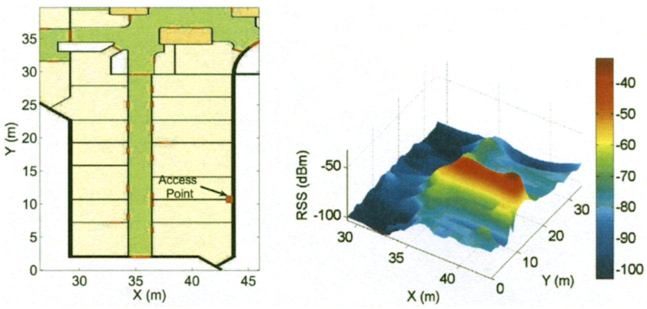

Figure 2. A radio map example in an office building.

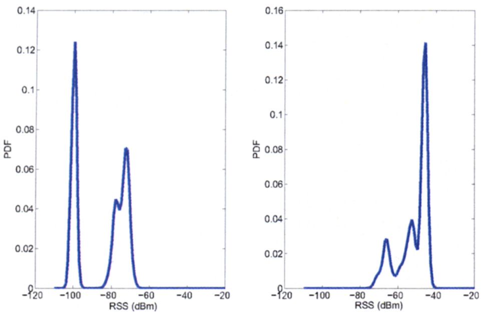

Figure 3. Examples of RSS distribution at reference points.

modeled parameterized. In this paper, we assume the distance is measurable by either accelerometer or odometer [10]. But the angle is unknown. Hence, the following formulas are derived to model the system:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} (d_k + \delta d_k)\cos(\theta_k) \\ (d_k + \delta d_k)\sin(\theta_k) \end{bmatrix}, \qquad (10)$$

$$\boldsymbol{p}_k = h_k\left(\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix}\right) + \boldsymbol{v}_{k+1}, \qquad (11)$$

where distance noise δd_k follows Gaussian distribution $\mathcal{N}\left(0, \sigma_d^2\right)$. The unknown moving direction is modeled by an uniform distributed variable θ_k. The observations $\boldsymbol{p}_k$ is the RSS vector. The observation noise $\boldsymbol{v}_{k+1}$ is non-Gaussian.

IV. Experimental Results

The test environment in this paper is our office building which has several WLAN access points installed. To better evaluate and understand the performance of different nonlinear filters, we make both simulations and field tests. In the simulation a big area and dense radio map is generated and the model parameters are perfectly known. But in the field test not too many reference points are used to construct radio maps and the motion angle does not follow the presumed uniform distribution. In what follows, the setups of simulation and field test are described and results are then discussed.

A. *Description of the Simulation*

We started with simulated data because the walking trace in simulation is generated from the known model parameters, which eliminates influence of the model error in the comparison. We simulate the radio distribution in our office building by a site-specific multi-wall radio propagation model as

$$\boldsymbol{p} = \boldsymbol{p}_0 - 10n log(d) - \sum WAF + v, \qquad (12)$$

where $\boldsymbol{p}_0$ stands for the AP transmission power in dBm, which is set to -20 dBm. n is the radio attenuation coefficient which equals to 1.25. WAF represents wall attenuation factor that is the partition value of walls between transmitter and receiver. Here we set WAF to 8.7 dB for concrete walls, 5.5 dB for glass windows and 4.3 dB for doors. 6 access points are assumed to

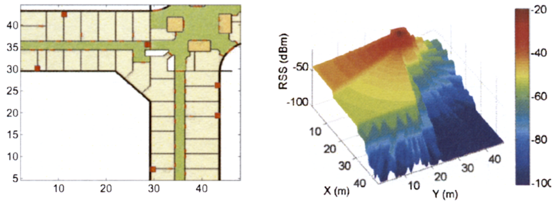

Figure 4. Left: simulated environment with 6 APs. Right: radio map example by multi-wall model.

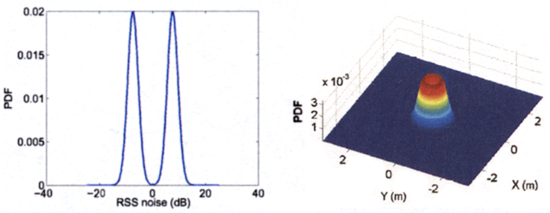

Figure 5. Left: Gaussian-mixture noise model. Right: simulated transition density.

be available in an area of 45m x 45m as the left figure of Fig. 4. The RSS noise v is a two-components Gaussian mixture density $0.5\mathcal{N}(-7.5, 22) + 0.5\mathcal{N}(7.5, 22)$ as shown in the left figure of Fig. 5. Note that although we use a parametric model to generate radio maps, it is still stored in a non-parametric way, i.g. the RSS vectors on 33660 grid points are stored in computer for the calculation of likelihood functions. The simulated walking trace follows the formula (10). All the model parameters are known to the estimator. The distance between two samples is assumed to be a Gaussian variable $\mathcal{N}\left(2.5m, 0.3m^2\right)$. The motion angle is uniformly distributed between $-\pi$ and π. So the transition density for each position is actually a ring as illustrated by the right figure of Fig. 5. Since we assume that the distance is the same for each step, the transition density is also constant. At the update step, a discrete likelihood probability is made by calculating the likelihood given the measured RSS at each point of the radio map.

B. *Description of Field Test*

We also evaluate the filters using the real data. In our building we installed 14 access points as shown in Fig. 6. We measured the radio distribution at 250 uniformly distributed grid points in an area of 15m x 35m before the system runs. The RSS distribution at each point is different because of the complex indoor environment. Fig. 3 shows two examples of RSS distributions. These distribution functions are stored in computer as a vector. In the online step, RSS and accelerometer data are collected while people move in a path as Fig. 6. We also use formula (10) and (11) to model the system. But the distance d_k is estimated by accelerometer and varies, i.e., we must calculate the transition density at each recursive step. The likelihood density is calculated in the same way as in the simulation. At each grid point the RSS measurement is compared with the stored RSS probability density function to

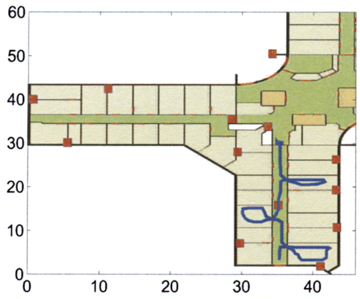

Figure 6. Environment of field test.

get the likelihood value. Finally a discrete likelihood density function is constructed for updating.

C. Results and Discussions

Fig. 7 and Fig. 8 show true posterior density functions at four successive steps in both simulation and field test. These densities are generated by the grid-based numerical method. We can observe that the posterior density shows a strong nonlinearity which favors the usage of nonlinear filters.

To compare the performance of different filters, we take the mean of location errors and computation time per step as evaluation criteria. Fig. 9 and Fig. 10 indicate how the mean of location errors is like given different number of components. Note that for Fourier filter, we take 10000 points in simulation and 750 points in the field test for FFT sampling. It is obvious that all the above nonlinear filters converge almost to the same value given enough components. Fourier filter converges with fewest components. Grid-based filter is worst. This indicates that the samples of Fourier-based filter is more efficient to represent the density functions. In addition, the results of the particle filter are the averaged one over 25 runs. Since particle filter is a random approach, its result also varies. Fig. 11 shows how the result of particle filer varies in the field test given different number of components. It is clearly observed that the fewer particles lead to a larger variation. Fig. 12 and Fig. 13 compare the computation time given different number of components. All the computations are done in a laptop with a single-core 2 GHz CPU. As shown, for grid-based filter, the computation time increases quadratically while for particle filter, the computation time increases linearly. In simulation we use much more grid points than the field test to construct radio map, which means calculating the particle weight takes more time in simulation than in the field test because every particle must find the closest grid from all of them. That is why it takes more time in simulation than in the field test given the same number of particles. Fourier filter has a almost constant computational time. This can be explained by the following facts. As shown in Table I, Fourier filter consists of two parts of calculations. The complexity of Fourier transform part depends on FFT sampling, following the complexity of $O\left(m \log m\right)$. The other part depends on the reduced number of components, following the complexity of $O\left(n^2\right)$. However, if the number of Fourier component is already sufficient to represent the densities, adding more components is just adding

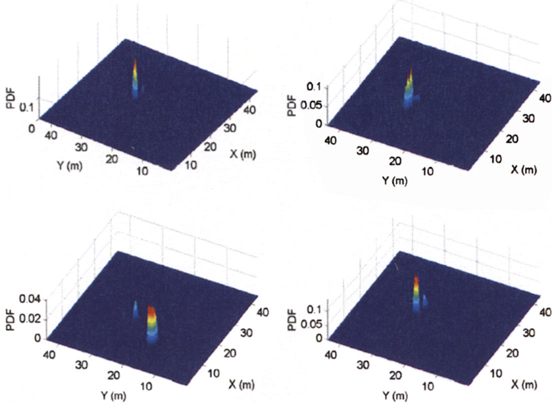

Figure 7. Examples of posterior density functions in simulation.

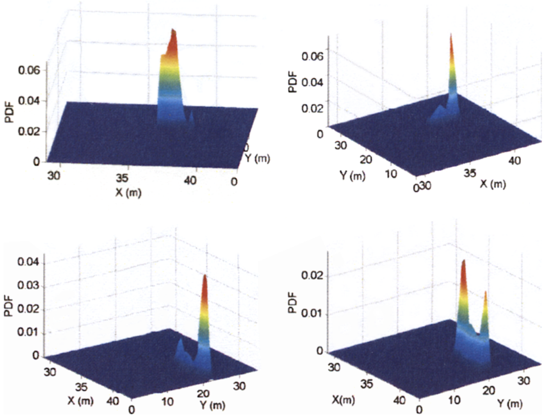

Figure 8. Examples of posterior density functions in the field test.

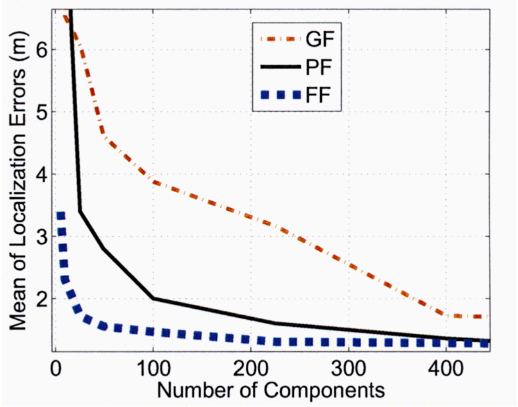

Figure 9. Mean of localization errors given different number of components for simulated data

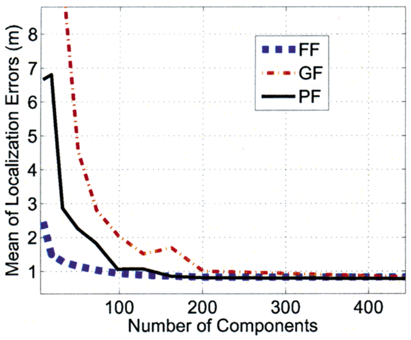

Figure 10. Mean of localization errors given different number of components for real data.

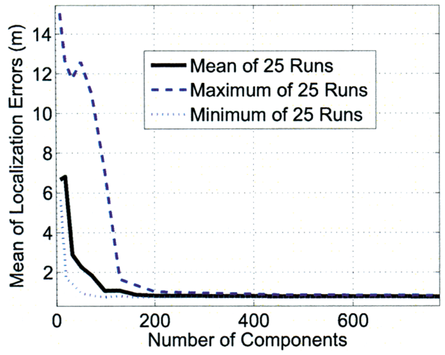

Figure 11. Variation of mean localization errors given different number of components for real data by particle filter.

some zero Fourier series and not increasing the running time. So as in Fig. 12 and Fig. 13 the FFT sampling m is fixed meaning a constant running time, which also dominates the calculation when the number of reduced component n is small. Since the Fourier filter can approximate the densities accurately with very few components (around 200 as shown in Fig. 9 and Fig. 10), more components will not increase the computation time neither. That is why the running time of Fourier filter in Fig. 12 and Fig. 13 looks like constant. We also remark the importance of FFT sampling in Fourier filter by Fig. 14 and Fig. 15 which show how the accuracy and running time change given different FFT sampling resolutions and a fixed reduced number of components. We can notice that more accurate FFT sampling gives better result but takes more time.

Fig. 16 and Fig. 17 plot the running time versus the mean of localization errors, which helps us to identify which filter makes the best trade-off between efficiency and accuracy. In our simulation, Fourier-based filter and particle filter have a compatible performance. But in real test, particle filter performs best. This is because the density function in simulation is more complicated and in simulation the transition density is the same for all steps so some computation time can be saved by reusing the previous transition density for Fourier filter.

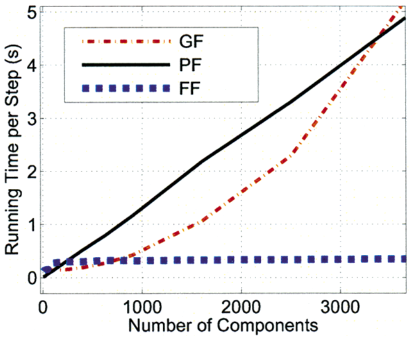

Figure 12. Computation time per step given different number of components for simulated data.

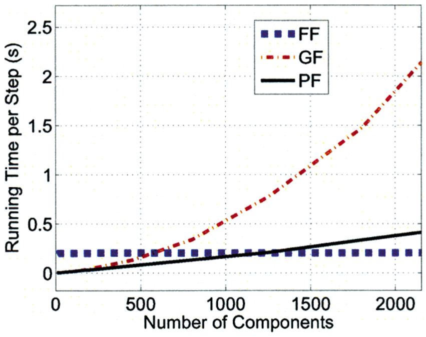

Figure 13. Computation time per step given different number of components for real data.

V. Conclusions

In this paper, we compare the performance of three non-linear filters for indoor WLAN positioning problems. We use both simulated and real measured data for evaluation. From the results, we see that given enough components, all three filters converge to the same result. Given very few components, Fourier-based filter performs best in the sense of accuracy, e.g., less than 200 components can already give very accurate results in our cases. The trade-off number depends on the complexity of system. Particle filter is faster than others for its simple algorithm structure with linear complexity. Fourier is slower with few components because the Fourier transform and component reductions take some extra time. This time is close to be constant, i.e., not increasing with more components, which can be regarded as fixed overhead cost. With more components, time consumption of grid-based method is close to $O\left(n^2\right)$ while that of particle filter is close to $O\left(n\right)$. For Fourier filter, the computation time increases with the density complexity instead of the number of components, i.e., if the current components are enough to represent the density, the more added components are zeros, taking very little time in calculation. Therefore, it is almost constant even using large number of components. The sampling resolution of Fourier filter is important. The larger resolution is used, the

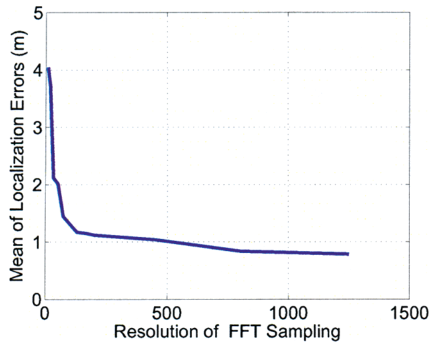

Figure 14. Mean of localization errors given different FFT sampling resolution and a reduced 200 components.

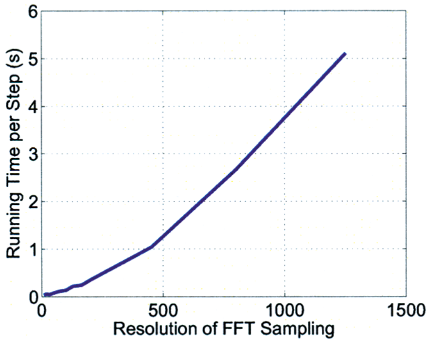

Figure 15. Running time per step given different FFT sampling resolution and a reduced 200 components.

more accurate result is achieved. But also more time is spent, which slows the Fourier filter in high-dimensional problems.

For practical positioning systems with limited computational power, e.g., slow processors, particle filter is the best choice. If the density functions are transfered between different nodes, Fourier-based filter should be used because fewer components require smaller bandwidth and communication time.

References

[1] Bahl P and Padmanabhan VN, "RADAR: An in-building RF-based user location and tracking system," in *IEEE INFOCOM 2000*, pp.775–784.

[2] Roos T, Myllymaki P, Tirri H, Misikangas P and Sievanen J, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vlo. 9, no. 3, July 2002.

[3] Henning Lenz, Bruno Betoni Parodi, Hui Wang, Andrei Szabo, Joachim Bamberger, Joachim Horn and Uwe D. Hanebeck, "Adaptive Localization in Adaptive Networks", *Chapter of Signal Processing Techniques for Knowledge Extraction and Information Fusion*. Springer, 2008.

[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, vol. 82 (Series D), pp. 35–45, 1960.

[5] Branko Ristic, Sanjeev Arulamplalm, Neil Gordon, *Beyond the Kalman filter*. Artech House, Boston, 2004.

[6] Julier, Simon J. and Jeffery K. Uhlmann. "A New Extension of the Kalman Filter to nonlinear Systems." in *The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II, SPIE*, 1997.

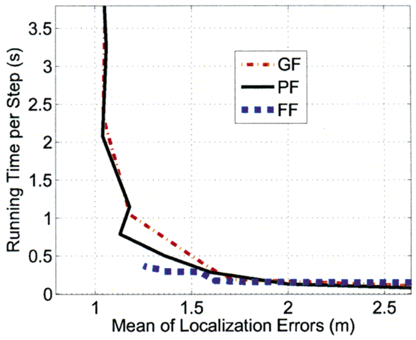

Figure 16. Running time per step given mean of localization errors for simulated data.

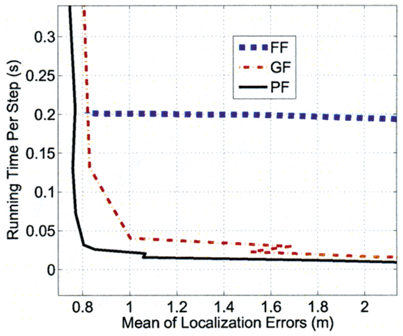

Figure 17. Running time per step given mean of localization errors for real data.

[7] Marco Huber, Dietrich Brunn and Uwe D. Hanebeck, "Closed-Form Prediction of Nonlinear Dynamic Systems by Means of Gaussian Mixture Approximation of the Transition Density," in *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, pp.98–103.

[8] Dietrich Brunn, Felix Sawo and Uwe D. Hanebeck, "Efficient nonlinear Bayesian estimation based on Fourier densities," in *2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, pp.312–322.

[9] Oliver C. Schrempf, Dietrich Brunn and Uwe D. Hanebeck, "Density Approximation Based on Dirac Mixtures with Regard to Nonlinear Estimation and Filtering," in *2006 IEEE Conference on Decision and Control (CDC 2006)*, Dec. 2006.

[10] Hui Wang, Henning Lenz, Andrei Szabo, Joachim Bamberger and Uwe D. Hanebeck, "WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors," in *Workshop on Positioning, Navigation and Communication, (WPNC 2007)*, Mar. 2007.

[11] Arnaud Doucet, Simon Godsill, and Christophe Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing* vol. 10 no. 3 pp.197–208,2000

[12] Fred Daum and Jim Huang, "Curse of Dimensionality and Particle Filters," *Aerospace Conference, 2003. Proceedings. 2003 IEEE*, March 2003.

[13] Miroslav Simandl and Ondej Straka, "Sampling densities of particle filter: a survey and comparison," *American Control Conference (ACC '07)*, pp. 4437–4442 2007

[14] R. Kronmal and M. Tarter, "The Estimation of Probability Densities and Cumulatives by Fourier Series Methods," *Journal of the American Statistical Association*, vol. 63, no. 323, pp. 925–952, September 1968.