Particle Filter for Positioning Accuracy Improvement in GNSS Receiver

Xiaowen Sun
College of Information Science and Technology
Dalian Maritime University
Dalian, China
e-mail: sunny@dlmu.edu.cn

Shufang Zhang, Qing Hu
College of Information Science and Technology
Dalian Maritime University
Dalian, China
e-mail: sfzhang@dlmu.edu.cn

***Abstract*—Traditional GNSS receiver usually uses Least Squares (LS) method or Kalman filtering to calculate the position. This paper proposes a new method that uses particle filtering for positioning calculation. The proposed particle filtering bases on LS method, but resolves the low positioning accuracy problem of LS. It also overcomes the Kalman filtering's drawback that need to know noise properties and receiver's dynamic model previously. The paper gives detailed implementation of the particle filter that is applicable to positioning calculation in GNSS receiver. Based on a suitably built model, the filter uses a modified resampling algorithm, and selects reasonable particle number and proposal distribution according to simulation test. These effectively reduce the calculation and improve the performance of the filter. Simulation results with GSS6700 simulator, based on the digital Intermediate Frequency (IF) signals, show that compared with traditional LS method and Kalman filtering, the proposed particle filtering achieves much higher positioning accuracy.**

Keywords-GNSS receiver; Positioning accuracy; Kalman filtering; Particle filtering; Resampling algorithm

I. INTRODUCTION

GNSS refers to Global Navigation Satellite System. So far, fully operational GNSSs are GPS and GLONASS. Chinese Compass navigation system and European Union's Galileo navigation system are scheduled to be operational by 2020 [1]. GNSS receiver calculates and outputs the precise position, velocity and time, which can be widely used in many areas. And the most commonly used algorithms in the calculation are LS method and Kalman filtering [2]. Solving the set of pseudorange observation equations by LS method, the overall solution minimizes the sum of the squares of the errors made in solving every single equation [3]. The advantages of this method are fast convergence rate, and little affected by initial rough coordinates of the receiver. But it uses current observations without considering other observations related to them. Therefore, the current observation errors greatly influence the positioning solution, resulting in low position accuracy [2][4]. Kalman filtering algorithm uses receiver's dynamic model and noise properties to produce estimates that is closer to the true unknown position. Positions at adjoining times are associated with one another, resulting in more smooth and accurate positions [4]. But Kalman filtering requires prior knowledge of receiver's dynamic model and noise properties that are not easy to get. And without the prior knowledge, it can easily lead to filter divergence[5][6]. Many researches try to overcome the drawback of the classic Kalman filter, but all of them need complex calculations which lead to difficult implementations [7][8]. In addition, initialization of the Kalman filter requires estimated position of the receiver. And large deviations of the initial estimates can lead to slow convergence rate or even divergence of the filter.

This paper proposes a new method that uses particle filtering for the calculation. It resolves the low positioning accuracy problem of LS without prior knowledge of receiver's dynamic model or noise properties. Particle filtering is a sequential Monte Carlo algorithm, which represents the probability by particle set, and it can be used for any state-space model [9][10]. This paper first builds a particle filtering model for positioning calculation in GNSS receiver, and then studies the implementation method of the filter based on that. In the implementation, it uses a modified resampling algorithm that discards low weighted particles and creates new particles by linear combination of high weighted particles. It also gives reasonable advice about selection of particles number and proposal distribution by simulation. These effectively reduce the calculation, improve the filter's performance, and make the particle filter more applicable to positioning in GNSS receiver. Simulation results in both static and dynamic states show that compared with traditional positioning methods, the proposed particle filtering achieves much higher positioning accuracy.

The rest of the paper is organized as follows. Section 2 provides a detailed description of particle filter and its implementation in GNSS receiver. Section 3 presents simulation results in both static and dynamic states. Section 4 concludes the paper.

II. PARTICLE FILTER AND ITS IMPLEMENTATION

A. Particle Filter

Particle filter is a statistical filter that based on recursive Bayesian estimation, approximating the Bayesian solution. It does not require any change to the system model, but gets optimal estimate of the solution by statistic [4].

A dynamic system with input vector can be expressed as follows:

$$x_{t+1} = f(x_t) + g(u_t) + w_t \tag{1a}$$

$$y_t = h(x_t) + e_t \tag{1b}$$

© 2013. The authors - Published by Atlantis Press

Here, x_t is state vector, u_t is measured input vector, w_t is process noise vector, y_t is observation vector and e_t is measurement noise vector. And (1a) is state equation, (1b) is measurement equation.

A set of available observations at time t can be expressed as: $Y_t = \{y_0, y_1, ..., y_t\}$. The key point with recursive Bayesian estimation is to calculate the posterior probability density function (pdf) $p(x_t | Y_t)$ of the state vector. The Bayesian solution is given by[11]:

$$p(x_{t+1} | Y_t) = \int p(x_{t+1} | x_t) p(x_t | Y_t) dx_t \quad (2a)$$

$$p(x_t | Y_t) = \frac{\int p(y_t | x_t) p(x_t | Y_{t-1})}{p(y_t | Y_{t-1})} \quad (2b)$$

Based on the Bayesian solution of $p(x_t | Y_t)$, statistical characteristics such as minimum mean square (MMS) estimate of state vector can be calculated by:

$$\hat{x}_t^{MMS} = \int x_t p(x_t | Y_t) dx_t \quad (3)$$

However, without restrictive linear Gaussian assumptions about the system models, no Bayesian solution of $p(x_t | Y_t)$ can be figured out by (2). Instead, particle filter is used to approximate the Bayesian solution. The filter uses N random samples called particles to represent the $p(x_t | Y_t)$, and each particle is given a weight, $\{x_t^i, \pi_t^i\}_{i=1}^N$, where π_t^i is the weight of particle x_t^i. Particle representation of $p(x_t | Y_t)$ is as follow:

$$p(x_t | Y_t) \approx \sum_{i=1}^{N} \pi_t^i \delta(x_t - x_t^i) \quad (4)$$

The particle filter usually uses importance sampling to create a set of particles for representing $p(x_t | Y_t)$, and the process can use any distribution $q(\cdot)$ which is known as a proposal distribution. Concretely, the particle filter consists of the following basic parts:

1）Create N samples from the proposal distribution: Generate $x_t^i \sim q(x_t)$, $i = 1, 2, ..., N$. Each sample is a particle.

2）Update and normalize the weights:

$$\pi_t^i = \pi_{t-1}^i p(y_t | x_t^i), \ i = 1, 2, ..., N \quad (5)$$

$$\pi_t^i = \pi_t^i / \sum_i \pi_t^i \quad (6)$$

3）Resampling: If the effective number of samples N_{eff} is less than a threshold N_{th}, that is

$$N_{eff} = 1 / \sum_i (\pi_t^i)^2 < N_{th} \quad (7)$$

then the particles are resampled to prevent high concentration of probability at a few particles that resulting in particle degeneration. And the particle degeneration is a key issue of particle filter.

Particle filter can handle complex multimodal posterior distributions, so it can be used in many systems. However, it also faces the problems of particle degeneration and large calculation. This research uses a modified resampling algorithm and a simplified filtering model to resolve the problems.

B. System Model

Based on above information, the filtering system model can be expressed as:

$$x_{t+1} = Ax_t + Bu_t + w_t \quad (8a)$$

$$y_t = Cx_t + e_t \quad (8b)$$

The model is linear in both state equation and measurement equation. The method proposed in this paper is particle filtering based on the position calculated by LS method. Therefore, the positioning results of LS are not only state vector, but also observation vector. And that means C in (8b) is a unit matrix.

For static or low dynamic carriers such as pedestrians, cars and ships, state vector x_t is $[x \ y \ z]^T$, and input vector u_t is $[v_x \ v_y \ v_z]^T$. The state equation can be modeled as:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} T_s & 0 & 0 \\ 0 & T_s & 0 \\ 0 & 0 & T_s \end{bmatrix} \begin{bmatrix} v_{xt} \\ v_{yt} \\ v_{zt} \end{bmatrix} + w_t \quad (9)$$

Here, x, y, z separately represent location components in three directions, v_x, v_y, v_z represent three velocity components, and T_s represents the sample period.

This model is applicable to the particle filter implementation, and can effectively reduce the calculation. As we know, the particle number grows exponentially with the dimensionality of state vector. The model considers the velocity v_t as input vector but not state vector to reduce the dimensionality, thus reduce calculation.

For high dynamic carriers such as aircraft and missiles, the model needs to consider additional acceleration components. The high dynamic condition is not discussed in this paper for the special applications.

C. Implementation of the Particle Filter in GNSS Receiver

The particle filtering for positioning calculation is actually the recursive estimation of the receiver position posterior distribution based on the built model. The implementation of particle filter for positioning in GNSS receiver is as shown in Fig.1.

In the initialization phase, first calculate initial position vector y_0 and velocity vector u_0 by LS method. Then, according to (8b), set the initial state vector x_0 equal to y_0. Finally, generate N initial particles x_0^i by a selected proposal distribution, and record the corresponding initial weights π_0^i, where $i = 1, 2, ..., N$.

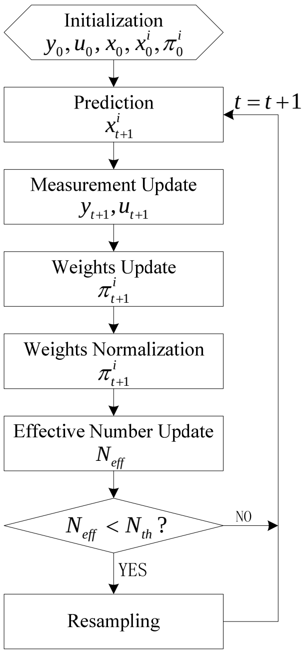

Fig. 1. Implementation of particle filter in GNSS receiver.

In the prediction phase, new state vector can be estimated by (9). Then, in the measurement update phase, calculate new observation vector y_{t+1} and input vector u_{t+1} by LS method. In the following two phases, update and normalize the weights separately according to (5) and (6). Next, calculate the effective particle number N_{eff} , and compare it with the threshold N_{th}. If N_{eff} is larger, make $t = t+1$, and iterate to prediction phase directly. Otherwise, go to the resampling phase before the iteration.

There are three key points with the particle filter implementation: particle number N, proposal distribution and resampling algorithm.

1)Particle number. The calculation increases with the particle number which grows exponentially with the dimensionality of the state vector. And the dimensionality is decreased to three in the filtering model. The specific value of the particle number N will be determined by simulation in the next section.

2)Proposal distribution. As we mentioned before, any proposal distribution can be used to generate the particles. The most common distributions are uniform and Gaussian. The next section will compare simulation results of the two distributions, and give advice on a better choice.

3)Resampling algorithm. Resampling algorithm can decrease the particle degeneration. Traditional algorithm copies high weighted particles and discards low weighted ones to avoid wasting calculation in less important particles. This paper uses a modified resampling algorithm based on the traditional one. It discards low weighted particles like the traditional algorithm, but creates new particles by linear combination of high weighted ones instead of simply coping them.

Concretely, when N_{eff} is less than N_{th} , calculates the linear combination of two highest weighted particles, that is:

$$x_t^{i-new} = \frac{1}{2} \times (x_t^{i-\max} + x_t^{i-\max 2}) \qquad (10)$$

and then replace the lowest weighted particle $x_t^{i-\min}$ with the new created one x_t^{i-new} .

The modified resampling algorithm creates high weighted new particles, and no one is repeated. This effectively increases the importance and diversity of the particles, thus improving the filter's performance.

III. SIMULATION AND ANALYSIS

Simulation experiments use GSS6700 simulator to generate signals in different scenes. GSS6700 is a powerful, flexible platform of Spirent Company for Multi-GNSS development, integration and verification test. It can be configured with up to 12 channels of one constellation or multiple constellations. A specific device is used to receive the simulation signals and then convert them to digital IF signals, based on which the simulations are implemented. To simplify the simulation, GPS is the only system considered, however, the results can generally extend to other GNSS systems.

A. Simulation in Static State

The simulator generates a static scene where the user receiver is at a fixed position. Three methods are separately used for the positioning calculation: LS method, Modified Kalman Filtering (MKF) and the proposed particle filtering. The proposed filter's particle number is 100, and its proposal distribution is Gaussian in this simulation. Positioning errors in three directions with LS method are separately in the range of [-24m, 24m], so set $4\sigma = 24$. Then the variance of Gaussian distribution σ^2 in particle filter is 36.

Fig.2 shows positioning results in Universal Transverse Mercartor (UTM) system of receivers using three methods for 45 seconds. The red "+" in the figure mean the real position. Each receiver gives a positioning result every 500 ms.

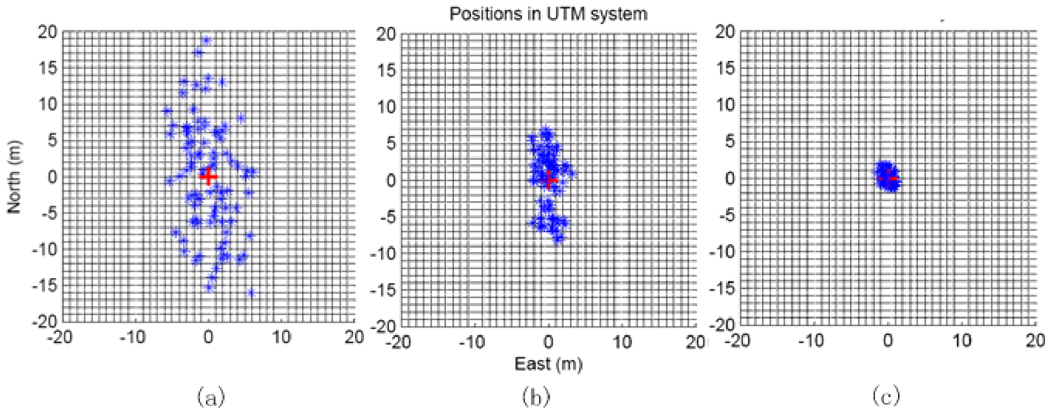

Fig. 2. Positioning results in UTM system. (a) LS method. (b) MKF method. (c) Proposed particle filtering.

Table.I gives the statistical analysis of the three receivers' root mean square error (RMSE). It is clear that the one using proposed particle filtering shows much better performance.

TABLE I. STATISTICAL ANALYSIS IN STATIC STATE

Methods	RMSE
LS method	13.1 m
MKF	7.2 m
Proposed PF	1.4 m

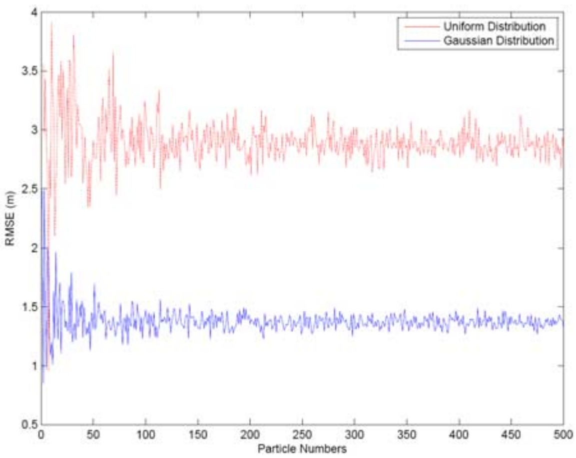

Fig.3. Performance comparison of particle filters with uniform and Gaussian as proposal distribution. The particle numbers are separately in the range of 1 to 500.

More simulation experiments are implemented to further determine the impact of particle number and proposal distribution in the particle filter. The variance of Gaussian distribution is 36, and the range of uniform distribution is [-24m, 24m]. The results are shown in Fig.3. The RMSE of both distributions are more stable with the increase of particle number, and Gaussian distribution shows better performance than the uniform one. Considering calculation and performance, a proposed particle filter with 100 particles and Gaussian distribution is recommended. The recommended filter is also used in the following dynamic simulation.

B. Simulation in Dynamic State

The simulator generates a dynamic scene where the user receiver moves in non-uniform linear motion for 10 minutes. Its velocity range is 0 m/s to 100 m/s, and its acceleration range is 0 m/s^2 to $10m/s^2$.

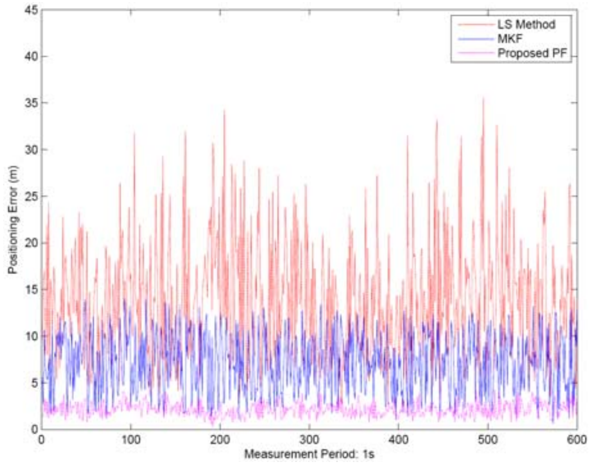

Fig.4. Performance comparison of three different methods in dynamic state for 10 minutes.

Receivers using LS method, MKF and proposed particle filtering are separately tested, and their positioning errors are shown in Fig.4. Each receiver gives a positioning result every second.

TABLE II. STATISTICAL ANALYSIS IN DYNAMIC STATE

Methods	RMSE
LS method	15.3 m
MKF	7.9 m
Proposed PF	2.3 m

According to statistical analysis, the RMSE of receivers using three methods are given in Table.II. The proposed particle filtering algorithm also shows much better performance in dynamic state.

IV. Conclusion

This paper proposes a new method that uses particle filtering combined with LS method for positioning in GNSS receiver. Based on a suitably built system model, the proposed particle filter uses a modified resampling algorithm and selects reasonable particle number and proposal distribution to improve the filter's performance. Simulation results in both static and dynamic states with GSS6700 simulator show effectiveness and superior performance of the method.

Acknowledgment

This research was supported by the Fundamental Research Funds for the Central Universities (No. 3132013307) and the National Key Technologies R&D Program of China (No. 2012BAH36B02). All support is gratefully acknowledged.

References

[1] B. H. Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS - Global Navigation Satellite Systems: GPS, GLONASS, Galileo & more*, New York: Springer-Verlag, 2008, pp. 365-430.

[2] M. Lashley, "Kalman Filter Based Tracking Algorithms for Software GPS Receivers," M.S. thesis, Eng., Auburn Univ., Auburn, Alabama, 2006.

[3] G. Strang, *Linear Algebra and its Applications*, Massachusetts: Wellesley Cambridge Press, 2003, pp. 206-218.

[4] G. Xie, Principles of GPS and Receiver Design. Beijing: Publishing House of Electronics Industry, 2009, pp.127-153.

[5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME-Journal of Basic Engineering*, vol. 82, pp. 35-45, Apr. 1960.

[6] E. D. Kaplan, *Understanding GPS Principles and Applications*, Norwood: Artech House, 1996, pp. 256-260.

[7] J. I. Canelon, R. S. Provence, N. Mehta, and L. S. Shieh, "An alternative Kalman innovation filter approach for receiver position estimation based on GPS measurements," *International Journal of Systems Science*, vol.38, pp.977-990, Dec. 2007.

[8] D. J. Jwo, and C. N. Lai, "Unscented Kalman filter with nonlinear dynamic process modeling for GPS navigation," *GPS Solut.*, vol.12, pp. 249-260, Apr. 2008.

[9] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.

[10] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, "Particle Filters for Positioning, Navigation, and Tracking," *IEEE Trans. Signal Processing*, vol. 50, pp. 425-437, Feb. 2002.

[11] N. Bergman, "Recursive Bayesian estimation: Navigation and tracking applications," Dissertation 579, Linköping Univ., Linköping, Sweden, 1999.