Chapter 13
Sensor Fusion Using Improvement of Resampling Algorithm Particle Filtering for Accurate Location of Mobile Robot

Xiang Gao and YaPing Fu

Abstract Presented chapter deals with the information fusion through used a Divisional Resampling of Particle Filter algorithm. The Divisional Resampling is based on the Multinomial Resampling and the Stratified Resampling, which divides the random number any arrangement into several sub-interval arranged by ascending. The chapter combined Divisional Resampling algorithm and the feedback control algorithm and then integrated the measurement information from odometer and sonar sensor, so that estimating the state vector of a mobile robot to achieve the aim of accurate location.

Keywords Mobile robot · Particle filter · Sensor fusion

13.1 Introduction

More recently, the Extended Kalman Filter (EKF) is an estimation algorithm that performs optimization in the least mean squares sense [1]. So EKF has been successfully applied to neural networks training and sensor fusion problems. However, EKF algorithm can result to unsatisfactory representations of the non-linear functions.

X. Gao (✉) · Y. Fu
Department of Automation Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China
e-mail: gaoxnj@gmail.com

Y. Fu
e-mail: fuyaping1986@126.com

X. He et al. (eds.), *Computer, Informatics, Cybernetics and Applications*, Lecture Notes in Electrical Engineering 107, DOI: 10.1007/978-94-007-1839-5_13,

To overcome these shortcomings [2], Particle Filter (PF) is mentioned. The particle filter is an alternative nonparametric implementation of the Bayes filter. In particle filters, the samples of a posterior distribution are called particles and are denoted. $X_t := \{x_t^1, x_t^2, \ldots, x_t^M\}$, each of particle x_t^m (with $1 < \mathrm{m} < \mathrm{M}$) is concrete instantiation of the state at time t. The input of this algorithm is the particle set x_{t-1}, then the most recent control u_t and the most recent measurement z_t.

The algorithm of Particle Filtering can provide optimal estimation in nonlinear non-Gaussian state models and nonlinear models which has improved its performance over the established nonlinear filtering approaches [3]. The basic idea of Particle Filter is based on state space experience conditional distribution of random sample to produce a set of these samples, which is called Particle and then on the basis of measurements constantly to adjust to the weight and position of Particle. In this chapter the Particle Filter has been employed for the localization of mobile robot by fusing data coming from odometer and sonar sensors which are collecting environmental information to realize the position of mobile robot's velocity and angle information, so as to achieve target tracking.

13.2 The Nonlinear Measurement Model Particle Filter

13.2.1 The Divisional Resampling of Particle Filter

The algorithm of Particle Filter has a drawback which is that after some numbers of iteration k, the particles' weight w_k^i will become 0 [4]. In the ideal case all the weights should converge to the value $1/N$. So particles are degenerated, they become unnecessary to modify the algorithm.

The basic idea of Resampling is removing small weight particle, so focused on right big particle particles. At present a wide range of present application of heavy sampling algorithms are Multinomial Resampling Residual Resampling, Stratified Resampling, and Systematic Resampling 7.

This chapter uses a kind of compromise resampling algorithm—the Divisional Resampling.

The steps are as follows:

Step1: $d = \lfloor \sqrt{N} \rfloor, R = N\%d$, N is the size of the particle collection
Step2: (1) if $R = 0$, then $d = \lfloor \sqrt{N} \rfloor, N^d = N/d, x = 1/N^d$
For $i = 1, \ldots, N^d A_i = \left\{ S_j^i \right\}_{1 \le j \le d}$, $S_j^i \sim U(((i-1)x, ix])$,
END
(2) else
$N^d = \lfloor N/d \rfloor + 1, x = (1 - R/N)/(N^d - 1)$
For $i = 1, \ldots, (N^d - 1)$
$A_i = \left\{ S_j^i \right\}_{1 \le j \le d}, S_j^i \sim U(((i-1)x, ix])$

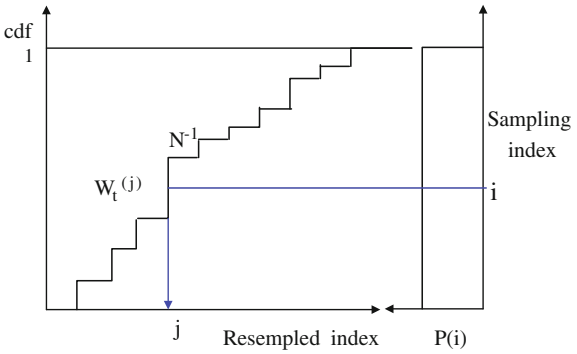

Fig. 13.1 Resempled index

END

$A_{N^d} = \left\{S_j^{N^d}\right\}_{1 \le j \le R}, S_j^{N^d} \sim U(((i-1)x, 1])$

Step3:

$$U = \{U_i\}_{1 \le i \le N} = \bigcup_{i=1}^{N^d} A_i, I^i = D_w^{inv}(U^i), \overset{\sim}{\xi}^i = \xi^{I^i}, \tilde{w}^i = 1/n, i = 1, \ldots, n$$

The process of Resampling as follows: after calculating the accumulative of particle probability, i is random sampling index, then we use i onto the probability of accumulative and regional, projection position and indicate the position of accumulative and corresponding domain, j is a new particles index 6. As shown in figure one shows.

The purpose of resampling is to reduce the number of particles less weight to focus on the larger particle weight. Basic idea is posteriori probability density function of sampling N times again [5]. The posteriori probability density function is $p(x_k|z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i)$, Where x_k^i is the state of a sample of robot position in the K time. w_k^i is the robot state x_k^i probability in the K time. Resampling N times, resulting in a new set of support points $\left\{x_k^{i^*}\right\}_{i=1}^{N}$ of particles, with $p(x_k^{i^*}) = p(xi_k) = w_k^i$. Resampling is an independent distribution, so the weight is resetting to $w_k^i = 1/N$ (Fig. 13.1).

13.2.2 The Prediction Stage

In the prediction stage, we mainly calculate $p(x(k)|Z^-)$ where $x(k)$ is time sequence of states in the k time, $Z^- = \{z(1), \ldots, z(n-1)\}$is observation sequence from 1 to t time. Through calculation of the recursive estimate edge distribution $p(x_k|z_{1:k})$, then there is no need to save the system state historical value.

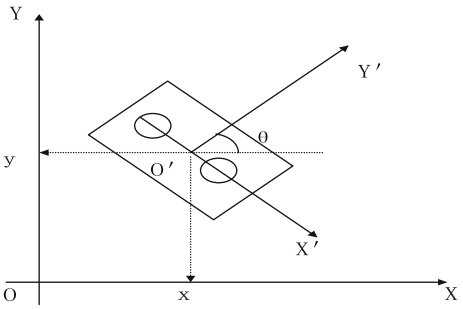

Fig. 13.2 Mobile robot motion model

13.2.3 Update Stage

Measurement update, given [6, 7] $z(k)$, and compute the new value of the state vector

$$p(x(k)|z) = \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i) \tag{13.1}$$

$$w_k^i = \frac{(w_{k^-}^i p(z(k)|x(k) = \xi_{k^-}^i))}{\sum_{j=1}^{N} w_{k^-}^j p(z(k)|x(k) = \xi_{k^-}^j)}$$

$$\xi_k^i = \xi_{k^-}^i$$

In the process of practical operation of weights, we need to normalize weights. After for normalization $w_k^{i^*} = w_k^i / \sum_{i=1}^{N} w_k^i$. A posteriori probability density in the system in k−1 time is $p(x_k|z_{k-1})$. Time measurement information obtained in the k time, in accordance with this article resampling algorithm selected a random sample of N points, after time and status update, N particles of the posterior density can be approximately considered $p(x_k|z_k)$.

13.3 Simulation Results

13.3.1 Improved Particle

Filter will be applied to mobile robot localization

(1) Mobile robot motion model

The object of the chapter studied is commonly used in two rounds of indoor robot, shown as follows (Fig. 13.2).

There is an encoder on the driving wheels and it provides a measurement of the incremental angle. The odometer sensor is used to obtain an estimation of the robot's angular velocity $w(t_k)$ and linear velocity $v(t_k)$. Given a sampling period is $\Delta t_k = t_{k+1} - t_k$.

Mobile robot's in continuous-time kinematic equation is:

$$\dot{x}(t) = v(t)\cos(\theta(t))\ \dot{y}(t) = v(t)\sin(\theta(t))\ \dot{\theta}(t) = w(t) \tag{13.2}$$

Where $(x, y)^T$ point is mobile robot's position and θ is the orientation of robot body rotation movement.

Establish its probability for motion model: $p(l_t|l_{t-1}, u_{t-1})$. The robot in t−1 time relative to the global coordinates the pose is $l_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1})^T$, After time t, reached the position l_t, then the movement model is:

$$l_t = l_{t-1} + \begin{bmatrix} \cos(\theta_{t-1}) & 0 \\ \sin(\theta_{t-1}) & 0 \\ 0 & 1 \end{bmatrix} \mathrm{u}_{t-1} + v_{t-1} \tag{13.3}$$

With: $\mathrm{u}_{t-1} = (\delta_{t-1}, \Delta\theta_{t-1})^T$ is the odometer model input, during the time of $(t-1, t)$, δ_{t-1} is robot's center displacement, $\Delta\theta_{t-1}$ is the angle of the robot rotated, v_{t-1} obeyed Gaussian measure noise. Defined the control law 11:

$$\begin{aligned} v &= v_d\cos(\theta_d - \theta) + k_1((x_d - x) \\ &\cos\theta + (y_d - y)\sin\theta) \\ w &= w_d + k_2 v_d \frac{\sin(\theta_d - \theta)}{\theta_d - \theta} \\ &((y_d - y)\cos\theta - (x_d - x)\sin\theta + k_3(\theta_d - \theta) \end{aligned} \tag{13.4}$$

k_1, k_2, k_3 is Gain.

Dynamics of the tracking error:

$$\begin{aligned} e_x &= x_d - x, e_y = y_d - y \\ \ddot{e}_x &+ K_{d1}\dot{e}_x + K_{p1}e_x = 0 \\ \ddot{e}_y &+ K_{d1}\dot{e}_y + K_{p1}e_y = 0 \end{aligned} \tag{13.5}$$

(2) Coordinate system and the sensor modeling

XOY is the plane the global coordinate system [8], $X'O'Y'$ is *XOY* rotated θ. This chapter uses odometer and ranging sonar sensors to collect mobile robot information. Mobile robot collects surrounding environment information through the sonar sensors. Sonar through the transmitter, meet obstacles issued pulse waves reflected back, then be receiver receive, the reflection of the Angle can be sonar sensors detected. Robot sonar sensors through the collection of information in the

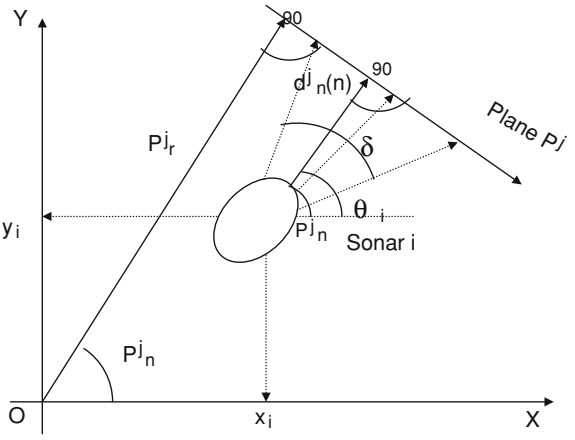

Fig. 13.3 Sonar position in the coordinate system

environment, Sonar pulsing sound through a transmitter, an obstacle to be reflected back, and then received by the receiver, the angle of reflection can be detected by sonar sensors. Sonar sensors in the coordinate system in the position XOY (x_i, y_i), Fig. 13.3 shows. Its direction angle θ_i, δ is the opening angle with sonar. Sonar feature information extraction environment, it is to get the raw data corresponding to the distance and angle information.

Sonar sensor model shown in Fig. 13.3:

Suppose K sampling time, the mobile robot pose $X(k) = (x(k), y(k), \theta(k))$ [9], after a rotation of the coordinate transformation, to the only sonar sensors i-coordinate system in the coordinate OXY (x_i, y_i) conversion to $O'X'Y'$ coordinate system.

System (x_i', y_i'), the sonar sensor model is:

$$\begin{aligned} x_i(k) &= x(k) + x_i' \sin(\theta(k)) + y_i'(\cos\theta(k)) \\ y_i(k) &= y(k) - x_i'(\cos\theta(k)) + y_i' \sin(\theta(k)) \\ \theta_i(k) &= \theta(k) + \theta_i \end{aligned} \tag{13.6}$$

Mobile robot wall as reflected in your environment P^j, sonar sensors are available with distance from the planes P_r^j, P_n^j, P_r^j is the distance between the origin O to the plane.

Sonar i to plane P^j distance of obstacles in k time for $d_i^j(k)$: i is the number of sonar sensors, j is the number of planes, Simplified model used in this article, as shown in Fig. 13.3. $i = 1, j = 1$, then $d_i^j(k)$ is: $d_i^j(k) = P_r^j - x_i(k)\cos(P_n^j) - y_i(k)\sin(P_n^j)$ (14)

Where $z(k\Delta t_k)$ is the odometer and sonar sensor measurements [10], $v(k\Delta t_k)$ is a Gaussian white noise sequence $\sim N(0, R(k\Delta t_k))$, $z((k+1)\Delta t_k) = G(X(k+1)\Delta t_k) + v(k\Delta t_k)$, $z(k\Delta t_k)$ can be decomposed into two sub-vectors of the form:

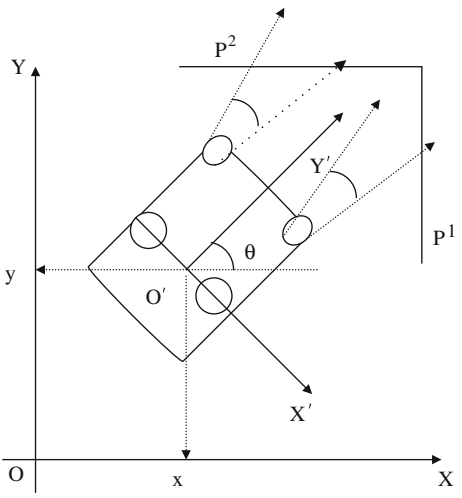

Fig. 13.4 Mobile robot odometer and sonar sensor model

$$\begin{aligned} z_1(k+1) &= [x(k) + v_1(k), \\ & y(k) + v_2(k), \theta(k) + v_3(k)] \\ z_2(k+1) &= [d_1^j(k) + v_4(k)] \end{aligned} \tag{13.7}$$

(Fig. 13.4).

13.3.2 Based on Particle Filter Multi-Sensor

Information fusion for mobile robot localization [11]:

Suppose the robot's initial position in OXY: $x(0) = 1m$, $y(0) = 0m$, $\theta(0) = 45°$

In the $O'X'Y'$ coordinate system, location of sonar sensors: $x_1' = 0.5m, y_1' = 0.5m, \theta_1' = 0°$

Plane position is $P^1 : p_r^1 = 15.5, p_n^1 = 45.00$.

State noise: $w(k) = 0, p(0) = diag[0.1, 0.1, 0.1]$, $R(0) = diag[0.1, 0.1, 0.1]$.

Kalman filter gain $K(k) \in R^{3x4}$

13.4 Simulation Results are as Follows

Figure 13.5, Fig. 13.6

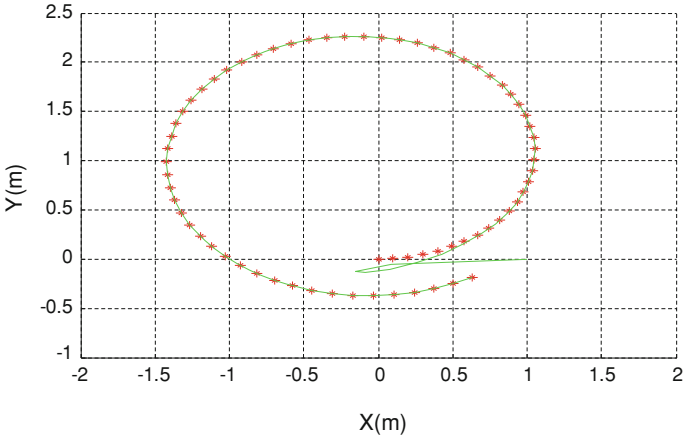

Fig. 13.5 Mobile robot trajectory tracking using particle filter (asterisk * denotes expectations of mobile robot, the green solid line represents the actual robot trajectory tracking)

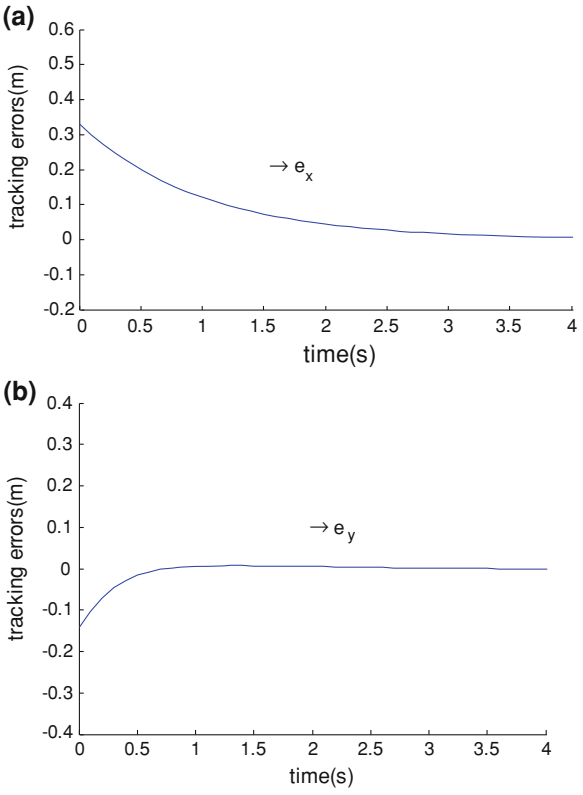

Fig. 13.6 **a** Robot distance error e_x. **b** Robot distance error e_y

13.5 Conclusion

The chapter introduces a Divisional Resampling based on Particle Filter algorithm which combines to robot feedback control theory through the fusion of odometer and sonar sensors from collecting environmental information. And applied the chapter introduces a Divisional Resampling based on Particle Filter algorithms to mobile robot to localization and tracking. Simulation results show that based on the particle filter in the Divisional Resampling algorithm, application control theory, will be able to realize the mobile robot for dynamic target effective and accurate tracking. While the future work primarily concentrates in sensor information fusion simplified algorithm, improving the ability of anti-interference and the operation precision, as well as the particle filter algorithm efficiency, robustness and real-time and develop more important particle filter resampling algorithm.

References

1. Ahrens JH, Khalil HK (2007) Closed-loop behavior of a class of nonlinear systems under EKF-based control. IEEE Trans Autom Control 52:536–540
2. Rigatos GG (2010) Extended Kalman and Particle Filtering for sensor fusion in motion control of mobile robots. Math Comput Simul 81:590–607
3. Von Thrun S, Burgard W, Fox D (2004) Probabilistic Robotics
4. Arulampalam S, Maskell SR, Gordon NJ, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50:174–188
5. Hua Z (2009) Multi-sensor fusion in mobile robot localization of application. Master's thesis, Wuhan University of Technology
6. Xuan Z (2010) Based on improved particle filter for mobile robot localization. doi:10.1089/ind.2010.6.041
7. Hong Z (2002) Shapes in different environments, real-time tracking. Ph.D. thesis, Institute of Automation Chinese Academy of Sciences
8. Cirncione C, Gurrieri GA (1997) A good overview of resampling and related methods [J]. Soc Sci Comput Rev 15(1):83–87
9. Wu BC (2006) Particle filter resampling algorithm and its application. Master thesis, Harbin Institute of Technology
10. Fang Z, Tong GF, Xu XH (2007) A robust and efficient mobile robot localization [J]. Automatica Sinica 33(1):48–53
11. Rigatos GG (2009) Particle Filtering for state estimation in nonlinear industrial systems. IEEE Trans Instrum Meas 58(11):3885–3900