Applied Intelligence
https://doi.org/10.1007/s10489-019-01534-0

A novel parallel accelerated CRPF algorithm

Jinhua Wang[1,2,3] · Jie Cao[1,2,3] · Wei Li[1] · Ping Yu[1,2,3] · Kaijie Huang[1]

Abstract

Particle filtering is one of the most important algorithms for solving state estimation of nonlinear systems and has been widely studied in many fields. However, due to the unknown complex noise in the actual system, its estimation performance is degraded. Moreover, when the number of particles increase, the real-time performance of the algorithm is poor. For these two problems above, this paper proposed a parallel acceleration CRPF (cost-reference particle filter) algorithm based on CUDA (Compute Unified Device Architecture). CRPF does not need known noise statistics in nonlinear system state estimation, which can reduce the influence of unknown noise on state estimation accuracy. Combined with GPU's (Graphics Processing Unit) multi-thread parallel computing capability, CRPF parallel acceleration can be realized. Since the data association can't be parallel resampled, all the particles are evenly distributed to multiple blocks, and resampling process can be parallelized by block parallel computing, so as to improve the speed of the algorithm. At the same time, in order to reduce the global particle performance degradation caused by block resampling, the particles with low probability mass in each block are optimized by using a portion of global high-quality particles. Through two sets of simulation experiments, it is proved that the proposed method has improved in estimation accuracy and the real-time performance has been improved significantly, which can provide a new idea for the practical application of nonlinear filtering method.

Keywords Particle filter · CRPF · GPU · Accelerated parallel processing · CUDA

1 Introduction

Particle filtering is based on Bayesian estimation and Monte Carlo sampling to solve the state estimation problem of non-resolvable nonlinear and non-Gaussian systems. The method is theoretically suitable for any nonlinear, non-Gaussian system, and effectively extends the application range of filtering techniques in parameter and state estimation. It has been widely used in many fields, and many improved algorithms for particle filtering have been proposed [1–4]. For example, UPF (Unscented Particle Filter), RPF (Regularized Particle Filter), APF (Auxiliary Particle Filter),GPF (Gaussian Particle Filter), MPF (Marginalized Particle Filter) etc. But the current PF algorithm has two core problems.

On the one hand, it is assumed that the statistical properties of process noise and measurement noise are known. In fact, the true statistical properties of noise depend not only on the system modeling error and the sensor measurement accuracy, but also on the environmental and human interference factors. So the priori information of system noise and measurement noise are unknown. When using the assumed noise for state estimation, it introduces a random error, which will cause the accumulation of errors in the process of multiple iterations, resulting in greatly reduced the filtering accuracy. To address this problem, the literature [5] first proposed a cost-reference particle filter (CRPF). CRPF does not need known statistical characteristics of process noise and measurement noise and achieves state estimation in the particle filter framework. It is an efficient method to solve the problem of high nonlinearity and unknown statistical characteristics of noise, and has received good results in research. The literature [6–8] introduced the principle and characteristics of CRPF algorithm in detail and proved the superiority of the algorithm in nonlinear, non-Gaussian systems and unknown statistical characteristics of noise through

✉ Jinhua Wang
wjh0615@lut.edu.cn

1 College of Electrical & Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China

2 Key Laboratory of Gansu Advanced Control for Industrial Processes, Lanzhou University of Technology, Lanzhou 730050, China

3 National Experimental Teaching Center of Electrical and Control Engineering, Lanzhou University of Technology, Lanzhou 730050, China

Published online: 17 October 2019

experiments in communication systems and target tracking. In [9], against the characteristics of large intensity and unknown statistical characteristics of sky-wave radar background noise, CRPF was used to estimate the target state of sky-wave over-the-horizon radar. Literature [10] proposed a filtering algorithm combining H∞ and CRPF, which was used in the state estimation problem of nonlinear dynamic systems with unknown non-Gaussian noise and achieved good results.

On the other hand, in order to achieve good estimation accuracy, a large number of particles are required, but a huge computational burden is added. When the real-time requirement is not met, the filter is meaningless in practical applications. In order to improve real-time performance, the literature [11] applied a matrix decomposition method to extract a small subset from all particles, and only calculated the weight of the subset in the weight calculation step, which greatly reduced the calculation time while keeping the filtering precision constant. The literature [12] combined correlative filtering and particle filtering to reduce computation time by using a minimum number of particles in the sampling step. Literature [11, 12] started from the basic principle of the PF algorithm and reduced the calculation time of the algorithm by optimizing and improving the algorithm steps. The most effective solution to this problem is to speed up the particle filtering through the hardware structure. Literature [13–15] implemented particle filters with FPGAs and embedded systems, but with high cost and low flexibility. In recent years, GPUs have been widely used in high-performance computing with their powerful floating-point computing capabilities [16–19]. The GPU can realize the large-scale parallel processing of data by independently assigning tasks in many computing units, which can significantly improve the computing speed. The CPU-GPU heterogeneous parallel structure has also become a new direction for the development of high-performance computer systems. The literature [20, 21] introduced the parallel implementation of the particle filter in CUDA framework and simulated the filtering precision and acceleration ratio, which proved the good effect of the method. The literature [18] proposed a GPU-based distributed computational particle filter algorithm, which was applied to the control of the robot arm and achieved tens of times the acceleration ratio in the case of more than one million particles. The literature [19] introduced in detail the implementation of the GPU-accelerated particle filter and auxiliary particle filter algorithm in CUDA framework, and proved that the method can obtain significant computational acceleration under the condition of ensuring accuracy by simulation of remelting process.

In summary, this paper considers the above two problems at the same time. Under the CUDA framework, a GPU-based parallel accelerated CRPF algorithm is proposed to achieve fast nonlinear state estimation under unknown noise conditions. In order to obtain the ideal acceleration ratio, it is necessary to maximize the parallelization of the particle filter algorithm. The bottleneck that restricts particle filter parallelization lies in the resampling step. Since it is necessary to calculate the weights of all the particles and normalize the weights for resampling, and then resampling all the particles according to the normalized weights, the processing of each particle in the resampling phase is not independent. The literature [21] studied parallel resampling and proposed a dual distribution dependent (D3) resampling method based on system resampling. The method increases the particle diversity after each iteration by utilizing the correlation of the prior distribution, and finally accelerates the particle filtering in the CUDA environment, but does not improve the parallel implementation of the resampling part, resulting the acceleration effect of the overall program operation is not ideal. The literature [20] proposed a parallel resampling method combining FRIM (finite redraw importance-maximizing) prior editing and localized resampling, which reduced the global operation time. This method was 5.73 times faster than the classic parallel algorithm. However, this method did not consider the effect of particle performance on the weight globally, so it would bring about particle degradation and reduction of filtering accuracy. Based on the above discussion, this paper distributes all the particles evenly to multiple blocks, realizes block parallel resampling, and improves the running speed of the algorithm. In order to reduce the global particle performance degradation caused by block resampling, the particles with low probability mass in each block are optimized by using a portion of global high-quality particles, so as to reduce particle degradation and maintain particle diversity while reducing global operating time consumption.

2 Cost-reference particle filter

Considering a nonlinear, non-Gaussian stochastic state-space model as follows:

$$\begin{cases} x_k = g(x_{k-1}) + u_k \\ y_k = h(x_k) + w_k \end{cases} \tag{1}$$

where k is the sampling instant and x_k is the system state vector at time k; y_k is the measurement vector of the system state at time k; $g(g)$ and $h(g)$ are the system state transfer function and the measurement function, respectively, and they can all be nonlinear functions; u_k and w_k are system state noise and measurement noise, respectively. Based on this state space model, the time-varying state sequence $\{x_1, x_2, \cdots, x_k\}$ of the target is estimated from the measurement sequence $\{y_1, y_2, \cdots, y_k\}$ obtained at each moment.

Define cost function and risk function to represent the performance quality of the particle, and the forgetting factor is introduced. Based on the principle of cost minimization, a particle weight evaluation method with unknown noise statistical characteristics is given. The cost function is defined as:

$$c\left(x_{0:k}^{i_p}|y_{1:k},\lambda\right)=\lambda c\left(x_{0:k-1}^{i_p}|y_{1:k-1},\lambda\right)+\Delta c\left(x_k^{i_p}|y_k\right) \tag{2}$$

Simplify the above formula as:

$$c_k^{i_p}=\lambda c_{k-1}^{i_p}+\Delta c_k^{i_p} \tag{3}$$

Where i_p is the particle index and $\lambda(0\leq\lambda\leq 1)$ is the forgetting factor, $\Delta c_k^{i_p}$ is the cost increment, which represents the accuracy of $x_k^{i_p}$ for a given measurement y_k, calculated by $\left(y_k-h\left(x_k^{i_p}\right)\right)^q$ $(q\geq 1)$. Cost-based stochastic measurements are represented by a set of "particle and cost" sets.

$$\Xi=\left\{x_k^{i_p},c_k^{i_p}\right\}_{i_p=1}^{N} \tag{4}$$

Where N is the number of particles. The risk function is defined as:

$$R\left(x_{k-1}^{i_p}|y_k\right)=\Delta c\left(E\left[x_k^{i_p}\right]|y_k\right)=\Delta c\left(g\left(x_{k-1}^{i_p}\right)|y_k\right) \tag{5}$$

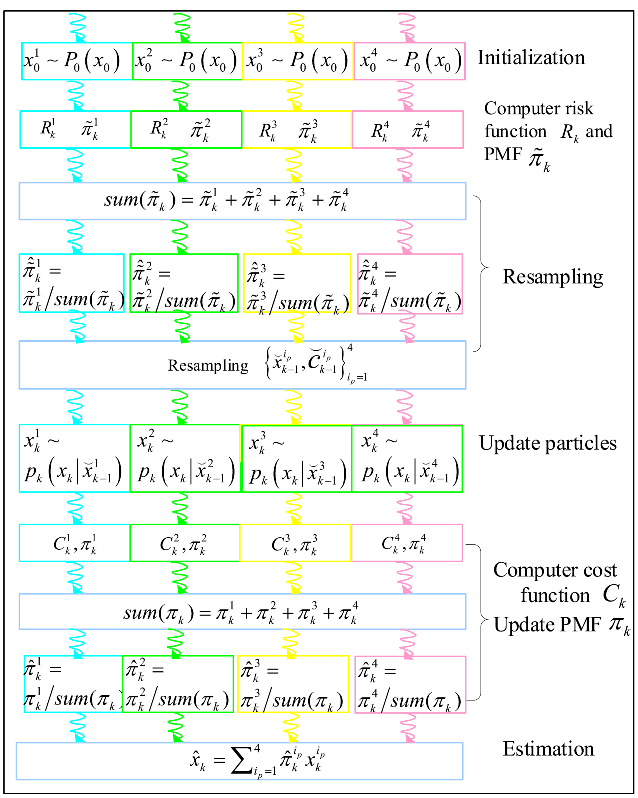


Fig. 1 Calculation flow of parallel particle CRPF with 4 particles

The risk function is the prediction of the cost increment $\Delta c_k^{i_p}$. Then, the prediction of the cost function is defined as:

$$C_k^{i_p}=\lambda C_{k-1}^{i_p}+R\left(x_{k-1}^{i_p}|y_k\right) \tag{6}$$

The probability mass function (PMF) is calculated according to eq. (7).

$$\pi_k^{i_p}\propto\mu_1\left(R_k^{i_p}\right)=\frac{1}{\left(R_k^{i_p}-\min\left\{R_k^{i_p}\right\}_{i_p=1}^{N}+\delta\right)^{\beta}} \tag{7}$$

δ, $\beta>0$, δ is to ensure that the denominator is not zero. According to the above parameter definition, the CRPF algorithm obtains state estimation by recursive calculation through risk estimation, selection, particle delivery, cost update steps, and the specific steps are shown in Algorithm CRPF.

It can be seen from the above algorithm steps that the CRPF algorithm is similar to the PF algorithm, and is also implemented based on three main steps of importance sampling, particle weight evaluation, and resampling. The calculation of process noise and measurement noise is not involved in the recursive estimation of CRPF algorithm. Therefore, no known statistical characteristics of noise are needed in the calculation, which improves the adverse effects of external random interference on PF based on the likelihood assessment. However, computational time consumption is an important factor affecting the

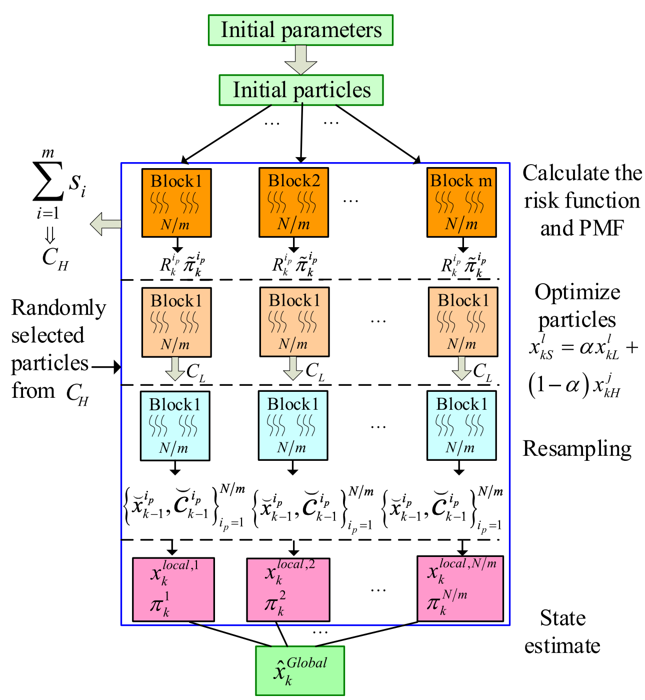


Fig. 2 Block diagram of the proposed CRPF

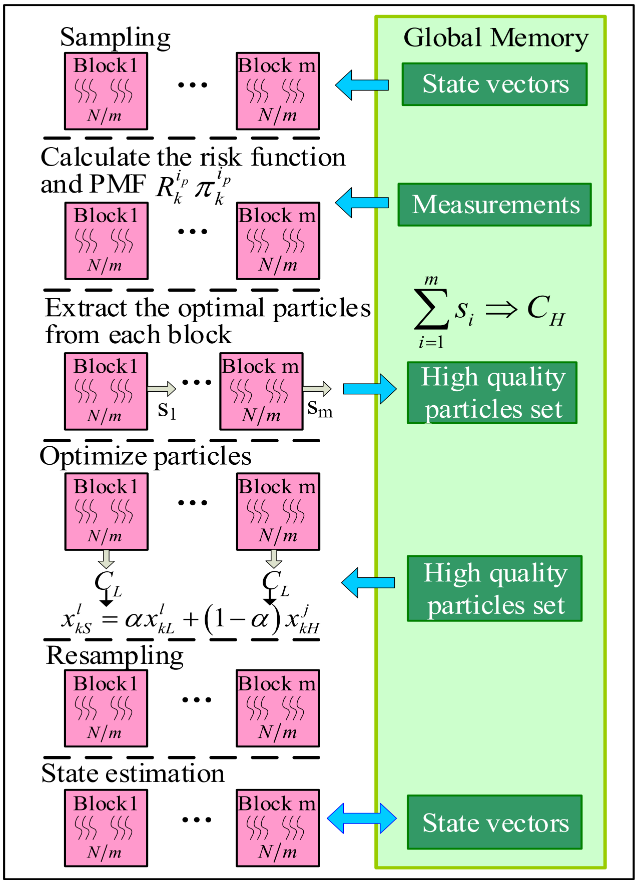


Fig. 3 Flow diagram of the proposed CRPF on CUDA

practical application of the algorithm. Using the powerful parallel data processing capability of GPU, this paper designs a parallel implementation method of CRPF algorithm on GPU based on CUDA framework.

3 CUDA-based parallel acceleration CRPF

3.1 Parallel computing flow of CRPF algorithm

The sampling and weight update in particle filtering are the easiest part of parallelization, and the resampling part is not easy to parallel due to the correlation between particles. So the usual practice is to parallelize parts that are easy to parallel, and serialize parts that are not easily parallel. Figure 1 shows the parallel computing flow of CRPF with four particles as an example. Each thread processes one particle, and in the PMF summation, resampling, and state estimation steps, parallel processing cannot be achieved due to the correlation between the data. Both the PMF summation and the state estimation have a summation operation, which is usually completed by the reduction summation, and the resampling step needs to generate a pseudo-random number by the CPU, and then transmits it to the GPU to complete the resampling of the global particle. The resampling step occupies most of the entire particle filtering running time [20].

3.2 Block parallel CRPF

In order to improve the parallelization of the program, this paper uses the local estimation idea to complete the filtering process. Set the number of particles to N, and use m blocks for local estimation, each block processes N/m particles, PMF summation, state summation, and resampling are performed independently by each block, and m local estimation results are obtained. Finally, the local estimation results are weighted by the formula (8) ~ (10) to obtain the global estimate.

$$\hat{x}_k^{Global} = \sum_{i=1}^{m} \hat{x}_k^{local,i} \hat{\pi}_k^{i} \tag{8}$$

$$\hat{\pi}_k^{i} = \frac{\pi_k^{i}}{\sum_{j=1}^{m} \pi_k^{j}} \tag{9}$$

$$\pi_k^{i} = \sum_{j=1}^{N/m} \pi_k^{i,j} \tag{10}$$

In the eqs. (8)~(10), $\hat{x}_k^{local,i}$ represents the local estimate of the i-th block, π_k^i represents the sum of the PMF of the i-th block, $\hat{\pi}_k^i$ is the corresponding normalized PMF value, and $\hat{x}_k^{Global}$ is the global estimated value at time k. The specific steps are shown in Algorithm of the Block parallel CRPF.

3.3 Improved parallel optimization resampling

According to the above method, using block parallel computation for parts that are not easy to calculate in parallel by m blocks can improve the parallel operation efficiency of the entire program. However, the descendant particles are obtained by locally resampling N/m particles in each block. The lack of consideration of the global optimal particle's greater

Table 1 Hardware information for evaluation

GPU	
Model	GeForce GTX 950
CUDA capability	5.2
Number of SMs	6
Number of cores	768
Bus bandwidth	105.76Gb/s
Clock frequency	1.24GHz

contribution to the state estimation results will result in local optimization and particle dispersion. This causes the global estimation error to increase or even diverge. Based on this, in order to obtain better filtering performance, the resampling method is improved in this paper. The particles with the largest PMF value are extracted from each block to form the optimal particle set C_H. Optimizing the partial particles with small PMF values in each block according to the given strategy by the optimal particle set, and finally resampling the optimized particles in each block. The block diagram of the proposed algorithm implemented on CUDA is shown in Fig. 2. The specific improved resampling algorithm is described below:

Step 1: According to the size of $\pi_k^{i_p}$, the particles whose PMF is less than the set threshold are selected to form a small weight particle set C_L;

$$x_k^{i_p} \in C_L, if\, \pi_k^{i_p} \leq \pi_T \tag{11}$$

Let the threshold $\pi_T = {}^1/_{3N}$ (N is the total number of particles), let $x_{kL}^l \in C_L (l = 1, ..., N_L)$, N_L represents the number of small weight particles. Extract the particles whose PMF value is greater than 1.5/N, and let the number of those large weight particles from each block is s_i, and totally extract $\sum_{i=1}^{m} s_i$ particles from m thread blocks to form a large weight particle set $\mathrm{C_H}$, $x_{kH}^j \in C_H (j = 1, ..., N_H)$, where $N_H = \sum_{i=1}^{m} s_i$ represents the number of large weight particles.

Step 2: Optimize small weight particles, and x_{kS}^l denotes optimized particles.

$$x_{kS}^l = \alpha x_{kL}^l + (1-\alpha) x_{kH}^j \tag{12}$$

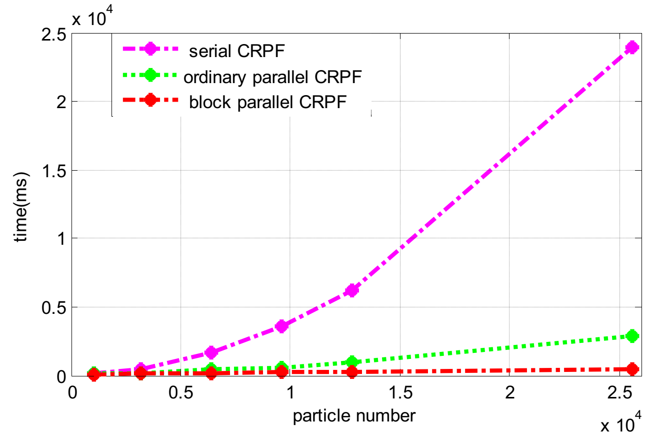


Fig. 4 Running time comparison of the serial CRPF algorithm, the ordinary parallel CRPF algorithm and the block parallel CRPF algorithm

The larger parameter $\alpha \in [0, 1]$ is, means that more information is transferred from x_{kL}^l to the descendant particle x_{kS}^l. If $\alpha = 1$, no optimization is performed, and x_{kH}^j is randomly extracted from C_H.

Step 3: The PMF $\pi_k^{i_p}$ is updated and normalized for the optimized particles, and the optimized particles are resampled with $\left\{\pi_k^{i_p}\right\}_{i_p=1}^{N}$ as the particle credibility to generate a particle cost set $\left\{x_{k-1}^{i_p}, c_{k-1}^{i_p}\right\}_{i_p=1}^{N}$.

Improved resampling algorithm has two advantages. Firstly, optimizing the particles of each thread block through global high-quality particles can improve the particle degradation of each thread block, improve the accuracy of local estimation, and improve the performance of global estimation. Secondly, only partial high-quality particles are extracted from each block, and those particles with small PMF values in each block are optimized to move to the high likelihood region, thereby improving the performance of the global particles and saving the time consumed by global optimization.

Table 2 The speedup ratios of the three parallel algorithms

Number of particles	speedup ratios		
	ordinary parallel CRPF	block parallel CRPF	optimized block parallel CRPF
1024	0.81	1.12	0.99
3200	2.48	3.95	3.00
6400	3.95	10.19	7.47
9600	5.92	16.51	9.35
12,800	6.24	24.45	10.79
25,600	8.37	55.27	13.21

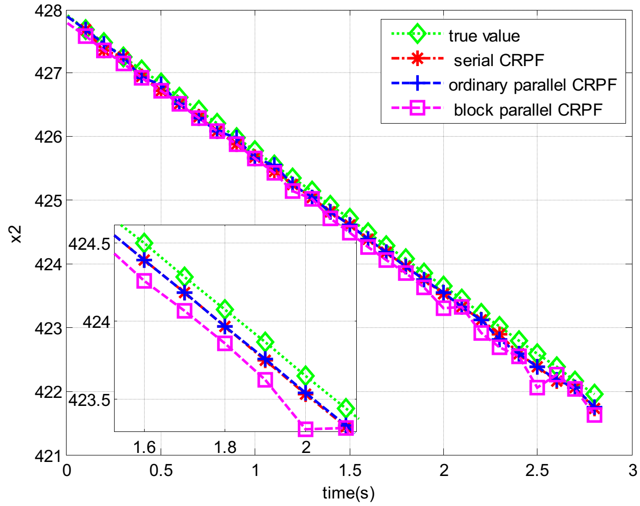


Fig. 5 Comparisons of x_2 estimation results using the serial CRPF algorithm, the ordinary parallel CRPF algorithm and the block parallel CRPF algorithm respectively

3.4 Parallel acceleration CRPF algorithm flow

The block diagram of the block parallel acceleration CRPF algorithm in the CUDA framework proposed in this paper is shown in Fig. 3. See Algorithm of the block parallel CRPF with optimization resampling for the specific steps.

4 Simulation and analysis

We take the dynamic model of 160 MW unit [22] as the object of study, take the opening of fuel control valve, steam turbine control valve and feed water control valve as input, take drum pressure and drum liquid density as state variables, and water level as observation variables. The state variables are tracked by serial CRPF algorithm, parallel CRPF algorithm and block parallel acceleration CRPF algorithm proposed in this paper, and compare and analyze the performance of these algorithms. Equation (13) is the discrete equation for the fuel unit model.

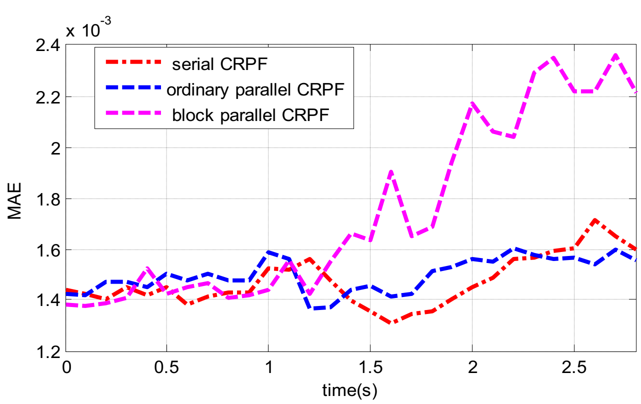


Fig. 6 Comparisons of estimation errors of the serial CRPF algorithm, the ordinary parallel CRPF algorithm and the block parallel CRPF algorithm

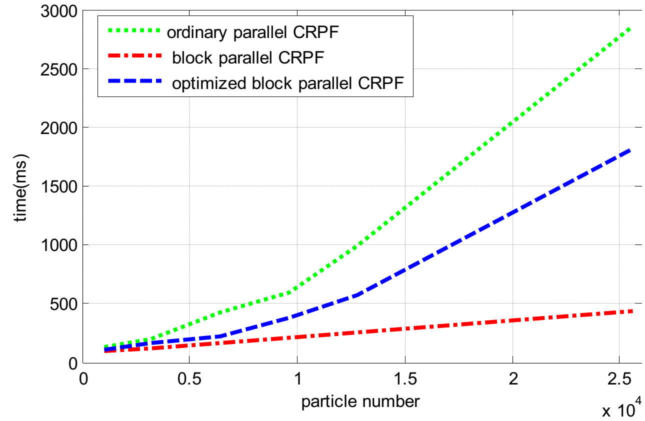


Fig. 7 Running time comparison of the ordinary parallel CRPF algorithm, the block parallel CRPF algorithm and the optimized block parallel CRPF algorithm

$$\begin{cases} x_{1,k} = x_{1,k-1} - A_1 \Delta t + v_{1,k} \\ x_{2,k} = x_{2,k-1} + A_2 \Delta t + v_{2,k} \\ y_k = 0.05B + w_k \end{cases} \tag{13}$$

Among them, the relevant variables are defined as:

$$A_1 = a_{11} u_{2,k} x_{1,k-1}^{9/8} - a_{12} u_{1,k} + a_{13} u_{3,k}$$
$$A_2 = \left(a_{21} u_{3,k} - \left(a_{22} u_{2,k} - a_{22}\right) x_{1,k-1}\right)/85$$
$$B = b_1 x_{2,k} + 100 a_{cs} + q_e/9 - b_2$$

$$a_{cs} = \frac{b_3 x_{2,k}\left(b_4 x_{1,k} - b_5\right)}{x_{2,k}\left(b_6 - b_7 x_{1,k}\right)}$$

$$q_e = \left(b_8 u_{2,k} - b_9\right) x_{1,k} + b_{10} u_{1,k} - b_{11} u_{3,k} - b_{12}$$

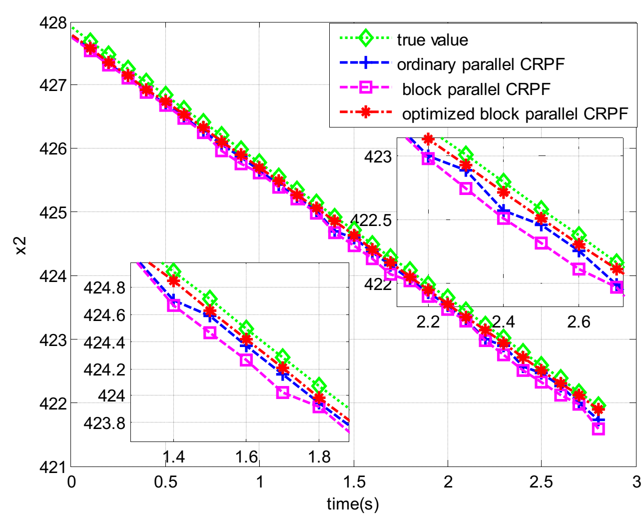


Fig. 8 Comparisons of x_2 estimation results using the ordinary parallel CRPF algorithm, the block parallel CRPF algorithm and the optimized block parallel CRPF algorithm

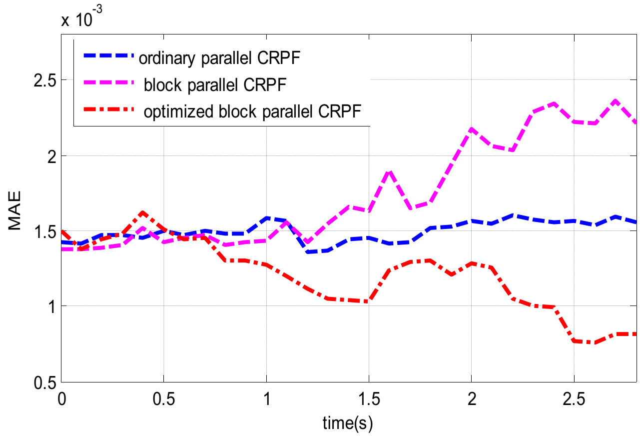


Fig. 9 Comparisons of estimation errors of the ordinary parallel CRPF algorithm, the block parallel CRPF algorithm and the optimized block parallel CRPF algorithm

Where x_1 is the drum pressure; x_2 is the drum liquid density; u_1 is the fuel regulating valve opening; u_2 is the turbine regulating valve opening; u_2 is the feed water regulating valve opening; y is the drum water level; v_k is the process noise, w_k is the measurement noise, they are non-Gaussian noise with unknown statistical properties. In this paper, the gamma noise of eq. (14) is used to simulate the noise in engineering practice.

$$\begin{cases} v_k \sim 0.09 \times 10^{-2} \times \Gamma(0.25, 0.5) \\ w_k \sim 0.16 \times 10^{-2} \times \Gamma(0.25, 0.5) \end{cases} \tag{14}$$

Set the initial state $x_0 = [108 \quad 428]^T$, discrete step size $\triangle t = 0.1s$, and the sensor sampling frequency is 1 Hz. Model parameter: $a_{11} = 0.0018$, $a_{12} = 0.9$, $a_{13} = 0.15$, $a_{21} = 141$, $a_{22} = 1.1$, $a_{23} = 0.19$, $b_1 = 0.131$, $b_2 = 0.068$, $b_3 = 0.00154$, $b_4 = 0.8$, $b_5 = 25.6$, $b_6 = 1.0394$, $b_7 = 0.00123$, $b_8 = 0.854$, $b_9 = 0.147$, $b_{10} = 45.59$, $b_{11} = 2.514$, $b_{12} = 2.096$, $u_t = [0.3 \quad 0.4 \quad 0.5]$.

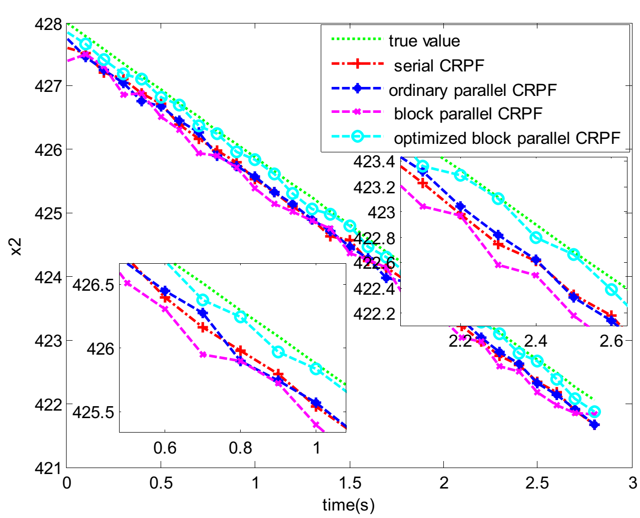


Fig. 10 Comparisons of x_2 estimation results using the serial CRPF algorithm, the ordinary parallel CRPF algorithm, the block parallel CRPF algorithm and the optimized block parallel CRPF algorithm with particle number 6400

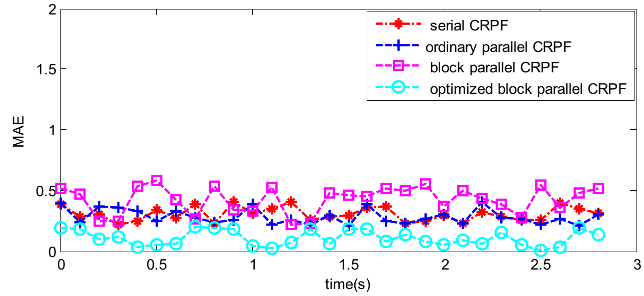


Fig. 11 Comparisons of estimation errors of the serial CRPF algorithm, the ordinary parallel CRPF algorithm, the block parallel CRPF algorithm and the optimized block parallel CRPF algorithm with particle number 6400

State initial prior distribution $x_0^{i_p} \sim N(x_0, \Sigma_0)$, $\Sigma_0 = diag(0.01, 0.01)$, $C_0^{i_p} = 0$, $\sigma_0^{2,i_p} = [0.01 \quad 0.01]^T$. Other parameters: $\delta = 0.1$, $q = 2$, $\beta = 2$, $\alpha = 0.5$, $\lambda = 0.85$.

Select the average absolute error as the evaluation index of the algorithm, which is defined as eq. (15).

$$MAE = \frac{1}{N_s T} \sum_{s=1}^{N_s} \sum_{k=1}^{T} |x_{sk} - \hat{x}_{sk}| \tag{15}$$

x_{sk} and $\hat{x}_{sk}$ are the actual and estimated values of the k-th step state of the s-th simulation, respectively, N_S is the total number of simulations, and T is the number of time steps in one simulation.

4.1 Experiment 1

The serial CRPF algorithm, the basic parallel CRPF algorithm and the block-parallel CRPF algorithm proposed in this paper are used to estimate the state of the fuel unit, and the performance of these algorithms are compared and analyzed. Table 1 is the hardware information for evaluation.

Figure 4 shows the running time of the three algorithms when different particle numbers are adopted. It can be seen

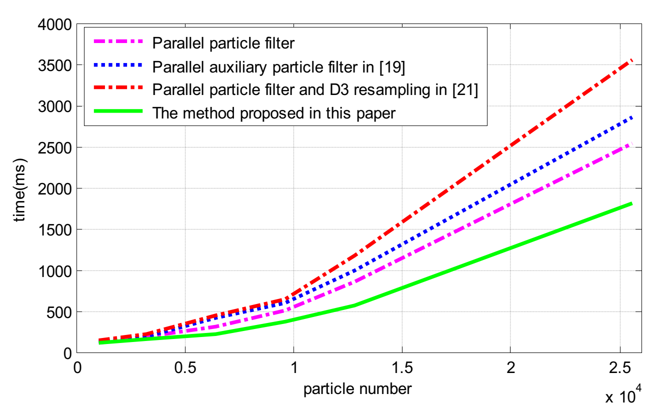


Fig. 12 Running time comparison of the parallel particle filter algorithm, the parallel auxiliary particle filter in [19], the parallel particle filter and D3 resampling in [21] and the optimized block parallel CRPF algorithm proposed in this paper

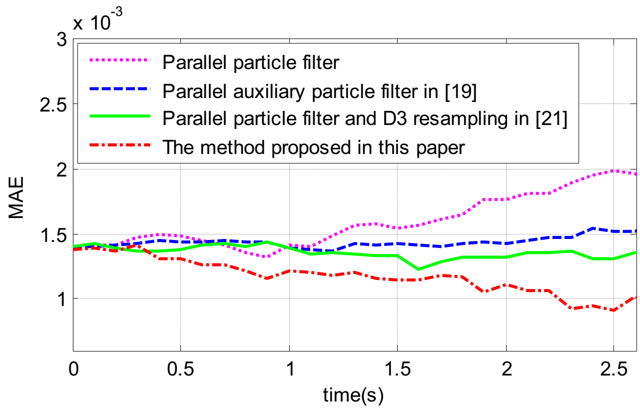


Fig. 13 Comparisons of estimation errors of the parallel particle filter algorithm, the parallel auxiliary particle filter in [19], the parallel particle filter and D3 resampling in [21] and the optimized block parallel CRPF algorithm proposed in this paper

from the figure that the running time of the basic parallel CRPF algorithm is much shorter than that of the serial CRPF algorithm. The running time of the block parallel CRPF algorithm is the least, and as the number of particles increases, the acceleration effect is better. Table 2 lists the acceleration ratio of the parallel algorithm when the number of particles is different. When the number of particles is small, the acceleration is relatively small, and the acceleration ratio increases with the increase of the number of particles. When the number of particles is 25,600, the acceleration ratio of the basic parallel CRPF algorithm is 8.37, while the block-parallel CRPF algorithm has an acceleration ratio of 55.27. Figures 5 and 6 are the state estimation results and estimation error curves of the three algorithms for the steam drum liquid density, respectively, and the number of particles used is 9600. It can be seen from the figure that the state estimation error of the serial CRPF algorithm and the basic parallel CRPF algorithm is very close, and after the multiple block iteration, the block resampling may cause a local optimum, the global estimation performance deteriorates, and the error gradually increase. It can be seen from the experimental results that the block-parallel CRPF algorithm proposed in this paper can obtain a very good acceleration effect compared with the basic parallel CRPF algorithm. However, block resampling distributes all particles evenly into m BLOCKs, and performs local resampling in each iteration. As the number of iterations increases, particle degradation may occur, and some of the BLOCK particles may have poor overall performance. The corresponding local estimation error increases, which ultimately leads to a decrease in the accuracy of the global estimation.

4.2 Experiment 2

The basic parallel CRPF algorithm, the block parallel CRPF algorithm proposed in this paper and the resampling optimized block-parallel CRPF algorithm are used to estimate the state of the fuel unit, and the running time and accuracy of the parallel algorithm before and after optimization are compared and analyzed.

Figure 7 is a comparison of the running time of three parallel algorithms in different particle numbers. The block parallel algorithm has the least running time, the basic parallel algorithm has the longest running time, the parallel algorithm of resampling optimization has less running time than the basic parallel algorithm, but the running time of the block parallel algorithm is longer. It can be seen from Table 2 that when the number of particles is small, the acceleration ratios of the three parallel algorithms are not much different. As the number of particles increases, the acceleration ratio increases. When the number of particles is 3200, the acceleration ratios of the basic parallel CRPF algorithm, the block parallel CRPF algorithm and the resampling optimized block-parallel CRPF algorithm are 2.48, 3.95, 3.00, respectively. When the number of particles is 25,600, the acceleration ratio is 8.37, 55.27, 13.21, respectively. The resampling optimized block-parallel CRPF algorithm has a lower acceleration ratio than the unoptimized block-parallel CRPF algorithm, but the acceleration ratio is significantly improved compared to the basic parallel CRPF. Figures 8 and 9 are the state estimation trajectory and error curve of the steam drum liquid density respectively. The number of particles used is 9600. It can be seen that the error of the block parallel CRPF algorithm gradually increases with the increase of the number of iterations, and the error is greater than the basic parallel CRPF algorithm, the block-parallel CRPF algorithm that introduces resampling optimization has the smallest estimation error and is smaller than the error of the basic parallel CRPF algorithm.

Figures 10 and 11 show the results and errors of the state estimation of the density of the steam drum liquid x_2 using the above four algorithms with particle number 6400. It can be seen from the results that the state estimation error of the optimized block parallel CRPF algorithm is the smallest, and the accuracy of the state estimation is improved compared with the CRPF algorithm.

It can be seen from the experimental results that when compared with the block parallel algorithm, by optimizing the inferior particles in each BLOCK by global optimal particles in resampling, the parallel CRPF algorithm with resampling optimization improves the global performance of the block parallel CRPF, and ultimately improves the accuracy of the algorithm.

4.3 Experiment 3

To further verify the performance of the proposed algorithm, the parallel particle filter algorithm, the parallel auxiliary particle filter in [19], the parallel particle filter and D3 resampling in [21] and the optimized block parallel CRPF algorithm proposed in this paper are used to estimate the state of the fuel unit, and the running time and accuracy of these algorithms are compared and analyzed. Figure 12 is a comparison of the running time of the four algorithms in different particle numbers. As can be seen from the graph, the running time of the optimized block parallel CRPF algorithm proposed in this paper is less than that of the other three algorithms, and the real-time performance has been significantly improved. Figure 13 shows the state estimation errors of the four algorithms. The number of particles used is 9600. As can be seen from the graph, the errors of the proposed algorithm are less than the errors of the other three algorithms, and with the increase of iteration times, the errors gradually decrease. The experimental results show that the proposed algorithm can obtain more accurate results in the case of unknown noise, and has a good acceleration effect in the CPU and GPU heterogeneous systems, which effectively improves the real-time performance.

5 Conclusions

Particle filtering is one of the important methods of state estimation. Real-time and accuracy are the key factors affecting the practical application of particle filter algorithms. In this paper, the performance of the particle filter algorithm is deeply analyzed and studied. The GPU-accelerated block-parallel CRPF algorithm is proposed, and its acceleration effect is remarkable. The acceleration ratio reaches 55.27 when the number of particles is 25,600. In order to improve the accuracy of the algorithm while improving the real-time performance, this paper further proposes a block-parallel CRPF algorithm based on the block-parallel CRPF algorithm. Experiments show that the accuracy of the optimized resampling block-parallel CRPF algorithm is significantly improved, and the acceleration ratio is also significantly improved compared with the basic parallel CRPF algorithm. The main works of this paper are summarized as follows:

- The problems of particle filter in state estimation of nonlinear non-Gaussian systems are analyzed in depth. In view of the influence of unknown noise on the accuracy of state estimation in real systems, the performance of CRPF algorithm is deeply analyzed. And a parallel accelerated CRPF algorithm is proposed.
- Aiming at the real-time problem of CRPF algorithm in state estimation, this paper proposes a block-parallel accelerated CRPF algorithm based on CUDA architecture by using GPU's powerful general parallel computing capability. By distributing particles evenly to m thread blocks, each thread block can perform local parallel resampling, which solves the problem that the resampling step is difficult to implement parallel computing due to data association, greatly improves the parallelization degree of the algorithm, and obtains a good acceleration effect.
- After multiple iterations, the block-parallel CRPF algorithm has uneven distribution of particle weights, which leads to the deterioration of the overall performance of some BLOCK particles, and the corresponding local estimation error increases, which reduces the accuracy of global estimation. Aiming at this problem, this paper proposes a resampling optimized block-parallel CRPF algorithm, which uses global high-quality particles to optimize small-weight particles in each thread block, improving the performance of particles and the accuracy of local estimation in each thread block, which improves the accuracy of the global estimate. At the same time, the real-time performance of the algorithm is significantly improved compared to the basic parallel CRPF algorithm.
- The running time and tracking error of serial CRPF algorithm, basic parallel CRPF algorithm, block parallel CRPF algorithm and resampling optimized block-parallel CRPF algorithm are compared and analyzed by simulation experiments, and the effectiveness of the proposed algorithm is verified.

Acknowledgments The authors are grateful to the anonymous reviewers for their comments, which will help to improve this paper.

Funding This work was supported by National Natural Science Foundation of China [Grant No. 61763028]; Natural science foundation of gansu province [Grant No. 1506RJZA105]; Open class of key laboratory for advanced control of industrial processes in gansu province, 2018.

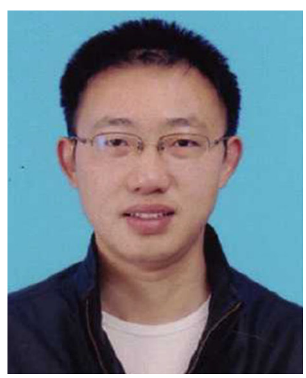
References

1. Chorin AJ, Tu X (2012) A tutorial on particle filters for online nonlinear/nongaussian Bayesia tracking. Esaim Mathematical Modelling & Numerical Analysis 46(3):535–543
2. Zuo JY, Jia YN, Gao QX (2013) Simplified unscented particle filter for nonlinear/non-Gaussian Bayesian estimation. J Syst Eng Electron 24(3):537–544
3. Van Leeuwen PJ (2010) Nonlinear data assimilation in geosciences: an extremely efficient particle filter. Q J R Meteorol Soc 136(653): 1991–1999
4. Read J, Achutegui K, Míguez J (2014) A distributed particle filter for nonlinear tracking in wireless sensor networks. Signal Process 98(5):121–134

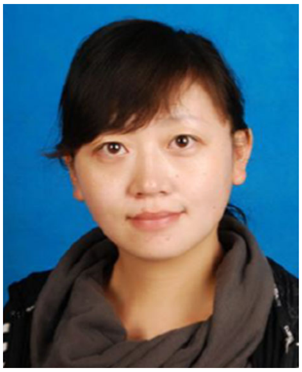
5. Miguez J, Xu S, Bugallo MF, Djuric PM (2004) Particle filtering for systems with unknown noise probability distributions. Statistical Signal Processing, 2003 IEEE Workshop on (pp. 522-525). IEEE
6. Lim J (2014) Particle filtering for nonlinear dynamic state systems with unknown noise statistics. Nonlinear Dyn 78(2):1369–1388
7. Yu YH, Zheng XY (2011) Particle filter with ant colony optimization for frequency offset estimation in OFDM systems with unknown noise distribution. Signal Process 91(5):1339–1342
8. Bugallo MF, Xu S, Djuri'c PM (2007) Performance comparison of EKF and particle filtering methods for maneuvering targets. Digital Signal Process 17(4):774–786
9. Jin LU, Hongtao SU, Shui P (2013) Target state estimation of the over-the-horizon radar using the cost-reference particle filter. Journal of Xidian University 40(5):20–25+59
10. YU YH (2013) Combining H∞ filter and cost-reference particle filter for conditionally linear dynamic systems in unknown non gaussian noises. Signal Process 93(7):1871–1878
11. Shabat G, Shmueli Y, Bermanis A, Averbuch A (2015) Accelerating particle filter using randomized multiscale and fast multipole type methods. IEEE Trans Pattern Anal Mach Intell 37(7):1396–1407
12. Li SJ, Zhao S, Cheng B, Chen JL (2018) Accelerated Particle Filter for Real-Time Visual Tracking With Decision Fusion. IEEE Signal Processing Letters 25(7):1094–1098
13. Zając M (2014) Online fault detection of a mobile robot with a parallelized particle filter. Neurocomputing 126(3):151–165
14. Schwiegelshohn F, Ossovski E, Hübner M (2016) A resampling method for parallel particle filter architectures. Microprocess Microsyst 47:314–320
15. Gan Q, Langlois JMP, Savaria Y (2014) A Parallel Systematic Resampling Algorithm for High-Speed Particle Filters in Embedded Systems. Circuits Syst Signal Process 33(11):3591–3602
16. Carter F, Hitschfeld N, Navarro CA, Soto R (2018) Gpu parallel simulation algorithm of brownian particles with excluded volume using delaunay triangulations. Comput Phys Commun 229(8):148–161
17. Bazow D, Heinz U, Strickland M (2018) Massively parallel simulations of relativistic fluid dynamics on graphics processing units with CUDA. Comput Phys Commun 225:92–113
18. Chitchian M, Simonetto A, van Amesfoort AS, Keviczky T (2013) Distributed Computation Particle Filters on GPU Architectures for Real-Time Control Applications. IEEE Trans Control Syst Technol 21(6):2224–2238
19. Lopez F, Zhang L, Mok A, Beaman J (2015) Particle filtering on gpu architectures for manufacturing applications. Comput Ind 71: 116–127
20. Chao MA, Chu CY, Chao CH, Wu AY (2010) Efficient parallelized particle filter design on CUDA. IEEE Signal Processing Systems (10):299–304
21. Das SK, Mazumdar C, Banerjee K (2014) Gpu accelerated novel particle filtering method. Computing 96(8):749–773
22. WANG J H, CAO J, LI W et al (2018) An adaptive CRPF fault diagnosis method under strong noise condition. Journal of Beijing University of Aeronautics and Astronautics 44(5):923–930

Jinhua Wang was born in Gansu, P. R. China. She received B.S. degree from Xi'an university of science and technology, Xi'an, China, in 2001, and M.S. degree from Lanzhou university of technology, Lanzhou, China, in 2010. She is currently a Ph. D. student in pattern recognition and intelligent system, Lanzhou University of Technology. She is also an associate professor in Electrical and Information Institute, Lanzhou University of Technology. Hers research interests are in the areas of pattern recognition and intelligent information processing.

Jie Cao was born in Gansu,P. R. China. She received B.Tech. degree from Ganshu Institute of Technology in 1987, Lanzhou, China, and the M. Tech. degree from Xi'an Jiaotong University in 1994. She is a professor and doctoral tutor of Lanzhou University of Technology. Hers research interests include information fusion and pattern recognition.



Wei Li was born in Henan, China. He received the B.E. degree in electronic and information engineering from Harbin Engineering University, Harbin, China in 2007, the M.S. degrees in signal and information processing from Lanzhou University of Technology, Lanzhou, China in 2012. He is currently working toward the Ph.D. degree in Control theory and control engineering from Lanzhou University of Technology, Lanzhou, China. His research interests intelligent algorithm analysis and modeling and simulation of intelligent wireless communication system.



Ping Yu was born in Gansu, P. R. China. She received B.S. degree from Lanzhou University of Technology in 2002, and M.S. degree from Lanzhou university, China, in 2008. She is currently a Ph. D. student Lanzhou university of technology. Hers research interests include pattern recognition and intelligent information processing.

Kaijie Huang was born in Hubei, P. R.China. He received B.S. degree from Hubei University of Science and Technology, China, in 2014, and M.S. degree from Lanzhou university of technology, China, in 2017. He is currently a Ph. D. student, Lanzhou University of Technology. His current research interests include pattern recognition and intelligent information.