

WEI ZHENG, Ph.D.

Corresponding author

E-mail: wzheng1@bjtu.edu.cn

JUAN HAN

E-mail: juanhao066@126.com

WEIJIE KONG

E-mail: 11120273@bjtu.edu.cn

LIXIANG WANG

E-mail: 13125105@bjtu.edu.cn

National Engineering Research Center
of Rail Transportation Operation and Control System,
Beijing Jiaotong University
No.3 Shangyuancun, Xizhimenwai, Haidian District, Beijing,
100044, China

Transport Technology

Review

Submitted: Mar. 20, 2014

Approved: Feb. 3, 2015

GROUP-SMA ALGORITHM BASED JOINT ESTIMATION OF TRAIN PARAMETER AND STATE

ABSTRACT

The braking rate and train arresting operation is important in the train braking performance. It is difficult to obtain the states of the train on time because of the measurement noise and a long calculation time. A type of Group Stochastic M-algorithm (GSMA) based on Rao-Blackwellization Particle Filter (RBPF) algorithm and Stochastic M-algorithm (SMA) is proposed in this paper. Compared with RBPF, GSMA based estimation precisions for the train braking rate and the control accelerations were improved by 78% and 62%, respectively. The calculation time of the GSMA was decreased by 70% compared with SMA.

KEY WORDS

parameter estimation; state estimation; particle filter; rail braking system;

1. INTRODUCTION

With the high-speed development of rail transportation, the safety and efficiency requirements during the transport process are increasing. Urban train parking precision affects the alignment of the train door with the platform screen door (PSD). The train braking system is a crucial factor affecting the parking accuracy. The braking rate is affecting the performance of the train braking system. The long-time continuous operation and the variation of the loading conditions and operation system will lead to the inevitable change of the braking rate. The displacement and velocity measurement accuracy is affecting the braking accuracy and it

is affected by the performance of the velocity sensors. The aforementioned precision problems can be converted to the joint estimation of unknown parameter and states by modelling the train braking process. In the train braking model, the velocity and displacement are states in the state space model, and the braking rate is the unknown parameter to be estimated.

The commonly used parameter identification methods are point estimation algorithms and filtering algorithms [1]. The traditional point estimation algorithms contain the least squares [2], maximum likelihood estimation method [3] and EM algorithm (Expectation Maximization) [4]. When the point estimation algorithms are used for the parameter estimation, it requires that the objective function is continuous and derivable. The estimated parameter value is searched by the gradient information, so it easily falls into the local minimum. The parameter identification method based on neural network [5] has the ability to approximate non-linear functions with arbitrary precision, but there are still several problems such as the determination of the network structure, sample data selection and network training algorithms. Genetic algorithm [6] has defects of premature or slow convergence. The Kalman filter and extended Kalman filter (EKF) have been applied successfully in the state estimation [7]. The Interacting Multiple Model (IMM) algorithm [8] is based on Kalman Filter (KF) and it has been used for the state estimation of the hybrid system. Knowledge of the transition probability between different models must be achieved in advance, which is difficult in practical engineering.

The Particle Filter (PF) uses the non-parametric Monte Carlo method to achieve the recursive Bayesian filter. This algorithm is suitable for any non-linear systems that can be modelled by state space equation and its estimation precision is approximate to the optimal estimation. N. Metropolis and S. Ulam in [9] proposed a particle filter for the first time. Development of the particle filter is introduced in [10] and its application in sensor diagnosis and accident detection in [11, 12]. There exist two major defects of the method. Firstly, the particle diversity decreases after several steps and all the particles tend to be of the same value. The Auxiliary particle filter (APF) and Gaussian particle filter (GPF) were discussed to solve this problem in [13]. The second problem is that the particle filter needs a large number of samples to approximate the posterior density of the system state. Rao-Blackwellization technology in [14] reduces the number of particles by reducing the dimension of the state vector. In RBPF algorithm, the state space is divided into linear sub-state space estimated by Kalman filter algorithm and non-linear sub-state space estimated by particle filter algorithm [15-18]. The RBPF has applications in the hybrid Gaussian [19-21], fixed parameters estimation [22], hidden Markov models (HMMs) [20-21], Dirichlet process models and Dynamic Bayesian Networks (DBN) [23]. There is no transition function for the unknown parameter in the state space equations. After several steps of the iterative calculation, all values of the parameter particles will tend to the same if the RBPF algorithm is used for the joint train braking rate estimation and the train states. SMA [24] is more efficient than RBPF for the state estimation of the hybrid system. SMA originally grows from the stochastic M-algorithm [25], based on a random sampling. In the SMA algorithm, the unknown parameter is discrete in several values and each discrete value corresponds to a system mode. When each parameter particle is predicted, the particle is transferred to all the possible discrete values from the unknown parameter, leading to an exponential growth of the particles number.

The GSMA is proposed to reduce the computation complexity, keeping the estimation precision. The second part of this paper introduces the principle of the estimation algorithms for the hybrid system. The GSMA is proposed in the third part. After the train braking model is specified in the fourth part, the RBPF algorithm, the SMA algorithm and the GSMA are employed for the joint estimation of train braking rate and states in the fifth part. Advantages and limitations of the GSMA algorithm are given in section six.

2. PROBLEM FORMULATION

Train braking system main function is to achieve consistent braking performance by a brake control-

ler. The driver completes the vehicle control using the vehicle traction control system and the brake control system, but the driver cannot directly manipulate the train power actuators. The brake model is a vehicle operating model including train braking control system [26-28]. The target acceleration has the fixed linear relationship with the drive input so it is chosen as the input of the braking controller and represented by $a_{Tar}(t)$. The brake controller will track $a_{Tar}(t)$ by feedback regulation. The dynamic regulation process is described as Equation (1):

$$\dot{\hat{a}}(t) = -\frac{1}{\tau}\hat{a}(t) + \frac{f}{\tau}a_{Tar}(t-T) \tag{1}$$

where $\hat{a}(t)$ denotes the control acceleration and it is the output of the controller, $\dot{\hat{a}}(t)$ means the derivation of $\hat{a}(t)$ after value t and f represents the braking rate which is defined as the ratio of the actual braking power with the expected braking power. The nominal value of f is 1, meaning the actual braking power is ideally equal to the expected braking power. Response time of the power actuator τ represents the time needed for the braking power to reach the desired value. Transmission delay time from the generation of the target acceleration to the execution of the power actuator is T , so $a_{Tar}(t-T)$ denotes the execution of the target acceleration with $a_{Tar}(t-T) = 0$ where $t \leq T$.

The accumulated acceleration of the vehicle is denoted by $\dot{v}(t)$ and it can be calculated from $\hat{a}(t)$ and additional acceleration caused by the bends and gradients of the railroad track, denoted by $d(t)$:

$$\dot{v}(t) = \hat{a}(t) + d(t) \tag{2}$$

The displacement of the urban railway is represented by $D(t)$ and it is calculated from the velocity represented by $v(t)$:

$$\dot{D}(t) = v(t) \tag{3}$$

Equations, from (1) to (3), describe the relationships between the target acceleration rate and the train operation status consisting of the acceleration, velocity and displacement. The corresponding block diagram of urban train braking model is presented in Figure 1. Notation \oplus represents the input signals point.

The Laplace linear operator of function $f(t)$ with a real argument $t(t \geq 0)$, transforms $f(t)$ to function

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt$$

with complex argument s . After Laplace transformation, differentiation and integration become multiplica-

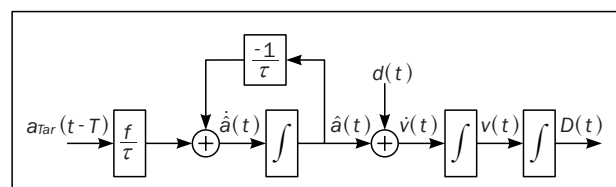


Figure 1 - Block diagram of urban train braking model

tion and division in polynomial equations. The inverse Laplace transformation reverts the solution to the time domain.

The Laplace transformation of Figure 1 is shown as Figure 2.

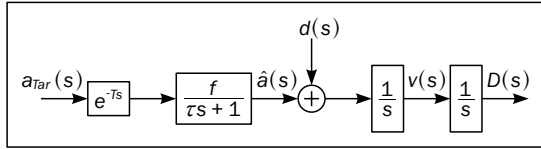


Figure 2 - Laplace transformation blocks of urban train braking model

Here $1/s$ represents the integration element and $f/(\tau s + 1)$ is the inertial transfer function in the Laplace space. Value e^{-Ts} denotes delay of T in the Laplace space, approximated with $(-s + \lambda)/(s + \lambda)$. Figure 1 is now approximated by Figure 3(a) and then by Figure 3(b) [26].

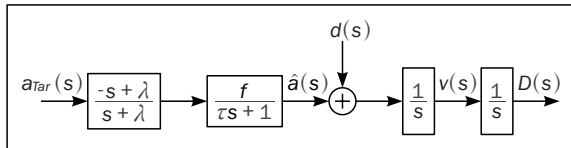


Figure 3(a) - Approximation transfer block of urban train braking model

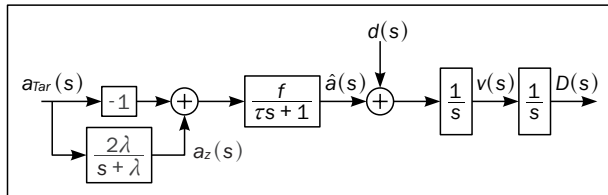


Figure 3(b) - Equivalent transfer block of urban train braking model from Figure 3(a)

Here value λ is an introduced coefficient calculated from T with the *Pade*() function in Matlab Software and $a_z(s)$ is considered as the value derived from $a_{Tar}(s)$ [26]. Displacement $D(t)$ velocity $v(t)$, control acceleration $\hat{a}(t)$ and intermediate variables $a_z(t)$ are selected as the state variables. By inverse Laplace transformation, values from Figure 3(b) are transformed to Figure 4.

Function $d(t)$ is assumed as the process Gaussian noise and it is denoted by scalar $\Omega(t)$. Assum-

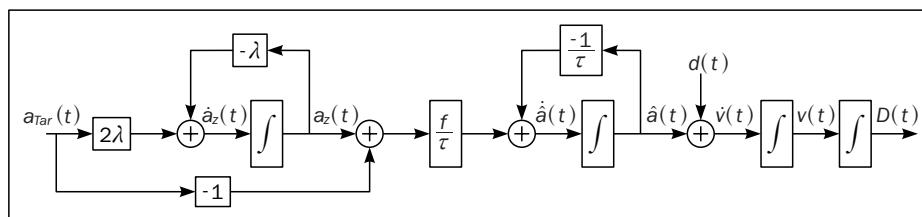


Figure 4 - Inverse Laplace transformation from Figure 3(b)

ing that $x = [D \ v \ \hat{a} \ a_z]^T$ and $y = [D \ v \ a_z]$, the state space equations of the model can be described as:

$$\dot{x} = Ax + ba_t + E\Omega$$

$$y = Cx$$

$$x = [D \ v \ \hat{a} \ a_z]^T$$

$$b = \begin{bmatrix} 0 & 0 & -\frac{f}{\tau} & 2\lambda \end{bmatrix}^T$$

$$E = [0 \ 1 \ 0 \ 0]^T$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(4)¹

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\tau} & \frac{f}{\tau} \\ 0 & 0 & 0 & -\lambda \end{bmatrix}$$

Here x represents the state vector of the system, y denotes the observation vector of the system and λ is a parameter related to the transmission delay time T . The system coefficient matrices are A , b , E and C and Ω represents the process Gaussian noise. The train braking model was demonstrated and verified with the practical operation data of YIZHUANG subway line in Beijing in literature [28]. As can be seen in the state space model, there are parameters λ , τ , f to be estimated. Static parameters λ and τ are estimated through offline methods.

3. PRESENTATION OF GSMA ALGORITHM

The hybrid system, particle filter algorithm, SMA and GSMA algorithms are presented further in the text.

3.1 Hybrid system

The state of the hybrid system (θ_k, x_k) is composed of parameter θ_k and the state x_k . Each system model corresponds to a set of state space equations containing state transition equations and observation equations:

$$\theta_{k+1} \sim p(\theta_{k+1} | \theta_k) \tag{5}$$

$$x_{k+1} = \varphi(\theta_k)x_k + B(\theta_k)u_k + \Gamma(\theta_k)w_k \tag{6}$$

1. In this formula T represents the transposition for the rows with columns.

$$y_k = H(\theta_k)x_k + v_k \quad (7)$$

Here, θ_{k+1} represent the value of the discrete system parameter at discrete moment $k+1$. The input value of the system at moment k is u_k . The probability transition matrix of discrete parameters is $p(\theta_{k+1}|\theta_k)$. The system coefficient matrices are $\varphi(\theta_k)$, $B(\theta_k)$, $\Gamma(\theta_k)$ and $H(\theta_k)$. Process noise and measurement noise at moment k are w_k and v_k and both are the independent white Gaussian noise. Parameter θ_{k+1} is calculated from the probability transition matrix $p(\theta_{k+1}|\theta_k)$ by (5). The continuous state transition equation and observation equation are (6) and (7), respectively. The state estimation is in calculating the parameter and state posterior probability density $p(\theta_k, x_k | Y_{1:k})$, where $Y_{1:k}$ describes the system measurements from moment 1 to moment k in steps.

3.2 Particle Filter algorithm

- (1) Filter is sequential to estimate parameter θ and state x of a system given a set of observation variables. Variable z is used to represent the parameters and states variables set $\{\theta, x\}$. The posterior probability density of z for $i = 0, 1, \dots, k-1, k$ is estimated by the given observation process values y_i , which presents the value of y at moment i , $i = 0, 1, \dots, k-1, k$.
- (2) Particle filters use a large random number of sample points sampled from the sampling function in the state space to approximate the posterior probability distribution of the estimated variables.
- (3) The procedures of particle filters are as follows [20]:
 - 1) For $i = 1, \dots, N$, draw samples from $z_k^{(i)} \sim p(z_k | z_{k-1}^{(i)})$. Here N is the number of the sample particles.
 - 2) For $i = 1, \dots, N$, update the importance weights by $\hat{\omega}_k^{(i)} = \omega_{k-1}^{(i)} p(y_k | z_k^{(i)})$,
 - 3) For $i = 1, \dots, N$, normalize the importance weights by

$$\omega_k^{(i)} = \frac{\hat{\omega}_k^{(i)}}{\sum_{i=1}^N (\hat{\omega}_k^{(i)})^2}$$

- 4) Compute an estimation of the effective number of particles \hat{N}_{eff} by:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (\hat{\omega}_k^{(i)})^2}$$

- 5) If the effective number of particles \hat{N}_{eff} is less than the given threshold N_{thr} , then resample in steps:
 - (a) Draw N particles from the current particle set with probabilities proportional to their weights. Replace the current particle set with the new one.
 - (b) For $i = 1, \dots, N$, set $\omega_k^{(i)} = 1/N$
 - 6) go back to step 2).

3.3 RBPF algorithm

RBPF algorithm is commonly used for parameter and state estimation of hybrid system. From the Bayesian theory, the state posterior probability density function $p(\theta_k, x_k | Y_{1:k})$ can be decomposed:

$$p(\theta_k, x_k | Y_{1:k}) = p(x_k | Y_{1:k}, \theta_k) p(\theta_k | Y_{1:k}) \quad (8)$$

The numerical solution of $p(\theta_k | Y_{1:k})$ can be approximately calculated through the particle filter algorithm. Given the available parameter values and the state coefficient matrices of the system, the analytic solution of $p(x_k | Y_{1:k}, \theta_k)$ can be derived using Kalman filter algorithm. If the particle filter is used to estimate the unknown parameter, $p(\theta_k | \theta_{1:k-1}^{(j)}, Y_{1:k})$ is chosen as the sampling function to approximate the unknown posterior probability density function of the parameter. The parameter particles are obtained based on $p(\theta_k | \theta_{1:k-1}^{(j)}, Y_{1:k})$ and then the probability density function of $p(\theta_k | Y_{1:k})$ in (8) can be estimated. Function $p(\theta_k | \theta_{1:k-1}^{(j)}, Y_{1:k})$ denotes the probability density function of the unknown parameter θ_k given the history values of the parameter particle $\theta_{1:k-1}^{(j)}$ and observations $Y_{1:k}$. According to the theory of probability, $p(\theta_k | \theta_{1:k-1}^{(j)}, Y_{1:k})$ can be simplified as [12]:

$$\begin{aligned} p(\theta_k | \theta_{1:k-1}^{(j)}, Y_{1:k}) &\propto \\ &\propto p(y_k | \theta_k^{(j)}, \theta_{1:k-1}^{(j)}, Y_{1:k-1}) p(\theta_k | \theta_{1:k-1}^{(j)}, Y_{1:k-1}) \\ &= p(\theta_k | \theta_{1:k-1}^{(j)}) \int p(y_k, x_k | \theta_k, \theta_{1:k-1}^{(j)}, Y_{1:k-1}) dx_k \quad (9) \\ &= p(\theta_k | \theta_{1:k-1}^{(j)}) \cdot \\ &\quad \cdot \int p(y_k | x_k, \theta_k) p(x_k | \theta_k, \theta_{1:k-1}^{(j)}, Y_{1:k-1}) dx_k \end{aligned}$$

The border of the integral is the value space of x_k . Values $\theta_k^{(j)}$ denote the j -th particle estimation values at moment k . If the mean of the observation values and variance of measurement noise v_k in (7) are denoted by $H(\theta_k)x_k$ and Q_v , respectively, then $p(y_k | x_k, \theta_k)$ is expressed as:

$$p(y_k | x_k, \theta_k) = \mathcal{N}(H(\theta_k)x_k, Q_v) \quad (10)$$

In formula (9), $p(x_k | \theta_k, \theta_{1:k-1}^{(j)}, Y_{1:k-1})$ is derived through the Kalman filter algorithm as [12]:

$$\begin{aligned} p(x_k^{(j)} | \theta_k, \theta_{1:k-1}^{(j)}, Y_{1:k-1}) &= \mathcal{N}(\hat{x}_{k|k-1}^{(j)}, P_{k|k-1}^{(j)}) \\ \hat{x}_{k|k-1}^{(j)} &= \varphi(\theta_k^{(j)})x_{k-1}^{(j)} + B(\theta_k^{(j)})u_k \quad (11) \end{aligned}$$

$P_{k|k-1}^{(j)} = \varphi(\theta_k^{(j)})P_{k-1}^{(j)}\varphi^T(\theta_k^{(j)}) + \Gamma(\theta_k^{(j)})Q_w\Gamma^T(\theta_k^{(j)})$ where $x_k^{(j)}$ and $\theta_k^{(j)}$ represent the state particles and parameter particles of the system at moment k . Here, $\hat{x}_{k|k-1}^{(j)}$ and $P_{k|k-1}^{(j)}$ represent the predicted state mean and variance of state $x_k^{(j)}$, respectively for the given state $x_{k-1}^{(j)}$. The variance of process noise w_k in (6) is Q_w . From Equations (9) to (11), the sampling function can be parsed as:

$$p(\theta_k^{(j)} | \theta_{1:k-1}^{(j)}, Y_{1:k}) \propto \mathcal{N}(\hat{y}_{k|k-1}^{(j)}, C_{y_{k|k-1}}^{(j)}) \quad (12)$$

where

$$\hat{y}_{k|k-1}^{(j)} = H(\theta_k^{(j)})\hat{x}_{k|k-1}^{(j)},$$

$$C_{y_{k|k-1}}^{(j)} = H(\theta_k^{(j)})P_{k|k-1}^{(j)}H^T(\theta_k^{(j)}) + Q_w$$

and the weights of the particles are calculated from:

$$w_k^{(j)} = p(y_k | \theta_k^{(j)})p(\theta_k^{(j)} | \theta_{k-1}^{(j)})/p(\theta_k^{(j)} | \theta_{k-1}^{(j)}, Y_{1:k-1})$$

$$\propto \sum_{s_k^{(j)}} p(y_k | \theta_k^{(j)}, \theta_{1:k-1}^{(j)}, Y_{1:k-1})p(\theta_k^{(j)} | \theta_{1:k-1}^{(j)}) \quad (13)$$

$$\propto \sum_{s_k^{(j)}} p(y_k | \theta_k^{(j)}, \theta_{1:k-1}^{(j)}, Y_{1:k-1})$$

Here $w_k^{(j)}$ represents the weight of the j -th particle at moment k and the system hybrid state estimation is obtained from the weighted sum of particle weights and particle values.

The mean and variance of continuous state corresponding to each particle can be derived using the Kalman filtering algorithm. The posterior probability density function of x_k can be expressed as a mixture of Gaussian function. The particle will show the degradation property if RBPF is used for the joint estimation of unknown parameter and system states because most of the parameter particles tend to transfer to the same discrete value after sampling step according to the sampling function, as the unknown parameter does not have the transition function. During the re-sampling process, the particles with large weights are replicated many times while the particles with small weights are rejected, and the replicated particles represent the samples. After several re-sampling steps, all the values of the parameter particles tend to be the same.

3.4 SMA algorithm

The SMA algorithm is derived from the M-algorithm to solve the particle degradation problem occurring in the RBPF algorithm. The unknown parameter θ in RBPF is assumed as discrete in $M + 1$ values $\{\theta^{(0)}, \theta^{(1)}, \dots, \theta^{(M)}\} (M \geq 0)$. Discrete values represent the value space and any particle for θ should belong to the value space. In each moment step from 0 to k , the parameter particle needs to be updated based on the sampling function. If the parameter particle $\theta_{k-1}^{(i)}$ is sampled at moment $k - 1$, the transition probability from parameter particle $\theta_{k-1}^{(i)}$ to all possible discrete values in $\{\theta^{(0)}, \theta^{(1)}, \dots, \theta^{(M)}\}$ is calculated first and then the parameter value corresponding to the largest transition probability is sampled as the parameter particle $\theta_k^{(i)}$. In each particle trajectory, although there are M possible values to which parameter particle $\theta_{k-1}^{(i)}$ can be transferred at moment k , only one value of them is finally retained. Since the diversity of the particles is important for obtaining accurate posterior probability density, it is preferable to retain more than one value which has relatively large weight. Compared with RBPF

algorithm, the SMA algorithm retains all the possible values for each particle trajectory and the particle trajectories grow like a tree. After each sampling step, one particle trajectory will be extended into M particle trajectories with the results that the total number of particles at time k become M times to that of particles at precious moment $k - 1$. If the number of particles is initialized as N at moment t_0 , the whole number of the particles will be $M^{k+1} \times N$ at moment t_{k+1} .

3.5 Proposed GSMA algorithm

The unknown parameter is usually static or changes slowly in the industry domain. During the estimation process, when each particle is sampled, it is enough to retain the possible values closer to the parameter value instead of retaining all the possible values.

For instance, assuming that one parameter is discrete as $\{1.0, 0.8, 0.5, 0.3, 0.1\}$. If the current estimated parameter value is 0.8, only the particle value 1.0 or 0.5 will be sampled for the next step, because the parameter is assumed to change slowly. So the estimated parameter will be first discrete and then all discrete values will be divided into different groups in steps as follows.

During each sampling step, parameter particles can transfer only within the group. For example, if discrete space of θ is assumed to contain five values as $\{\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \theta^{(3)}, \theta^{(4)}\}$, then at least two different groups could be generated: $\{\theta^{(0)}, \theta^{(1)}, \theta^{(2)}\}$ and $\{\theta^{(2)}, \theta^{(3)}, \theta^{(4)}\}$. The parameter particle value $\theta^{(1)}$ at moment $k - 1$, can transfer to a value in the first group, not in the second one.

The member of the group should include the similar value in the discrete space.

Each discrete value is allowed to exist in different groups. The particle should have the transferring possibility between different groups, such as $\{\theta^{(0)}, \theta^{(1)}, \theta^{(2)}\}$ and $\{\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\}$. If the value of the parameter is appointed as $\theta^{(1)}$ initially and since $\theta^{(2)}$ exists both in these two groups, the sampling scope for $\theta^{(2)}$ is possibly in the second group.

The group, in which the average value of the member is the closest to the parameter particle, will be selected. The average values of the groups should cover all the discrete values in the state space, except the two boundary values in the group. The number of members in a group is kept as small as possible, but at least three. If the state space of estimated parameter θ is $\{\theta^{(0)}, \theta^{(1)}, \dots, \theta^{(M)}\}$, the largest number of the groups is $M - 2$. The groups could be specified as $\{\theta^{(0)}, \theta^{(1)}, \theta^{(2)}\}$, $\{\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\}$, $\{\theta^{(2)}, \theta^{(3)}, \theta^{(4)}\}$, ..., $\{\theta^{(i-1)}, \theta^{(i)}, \theta^{(i+1)}\}$, ..., $\{\theta^{(M-2)}, \theta^{(M-1)}, \theta^{(M)}\}$.

The particle filter procedure is described in Figure 5. The discrete space of estimated values is assumed as $\{1, 2, 3, 4, 5\}$ and the space is grouped into three

groups shown as $\{1,2,3\}$, $\{2,3,4\}$, $\{3,4,5\}$. The initial value of the parameter is set as 3 and the number of the initial particles is described as N_0 . Figure 5 shows the increased procedure from moment 0 to moment $t+n$ for one particle. For any t , re-sampling scheme will be executed if the number of the particles reaches the boundary maximum number Q and N particles with higher weights will be kept for the next step.

The calculation steps of the GSMA algorithm are shown as Figure 8, where S denotes the particles, N_0 is set as 1 and N represents the number of re-sampling particles; if the space is set as $\{1,2,3,4,5\}$ then $a_i \in \{1,2,3\}$, $b_i \in \{2,3,4\}$, $c_i \in \{3,4,5\}$, $i \in \{1,2,3\}$.

Step 1: Parameter particles initialization:

From $\theta_{k+1} \sim p(\theta_{k+1}|\theta_k)$ in (5), parameter estimation value of θ_{k+1} is calculated from $p(\theta_{k+1}|\theta_k)$. In the particle filter algorithm, $p(\theta_{k+1}|\theta_k)$ is represented by $\theta_{(k+1)|k}^{(i)}$, value of $\theta_{k+1}^{(i)}$, estimated from $\theta_k^{(i)}$. It is assumed that $\theta_{1|0}^{(i)} \sim p(\theta_0)$, $i = 1, \dots, N_0$, particle value $\theta_{1|0}^{(i)}$ is sampled from the initial defined particle values of parameter θ . The weights of the particle are set as $1/N_0$.

Step 2: States particles initialization: $x_0^{(i)} = \hat{x}_0$, $i = 1, \dots, N_0$ and $P_0^{(i)} = p_0$, $i = 1, \dots, N_0$, where the particles for $x_0^{(i)}$ and the particles for $P_0^{(i)}$ are sam-

pled on the normal distribution with the mean value \hat{x}_0 and variance p_0 , at moment 0.

Step 3: For $k = 1, 2, \dots$, repeat the following steps:

For $i = 1, 2, \dots, N$, the state particle $x_{k|k-1}^{(i)}$ of the system and the mean square error matrix $P_{k|k-1}^{(i)}$ are predicted by

$$\begin{aligned} x_{k|k-1}^{(i)} &= \varphi(\theta_{k|k-1}^{(i)})x_{k-1}^{(i)} + B(\theta_{k|k-1}^{(i)})u_k \\ P_{k|k-1}^{(i)} &= \varphi(\theta_{k|k-1}^{(i)})P_{k-1}^{(i)}\varphi^T(\theta_{k|k-1}^{(i)}) + \Gamma(\theta_{k|k-1}^{(i)})Q\omega\Gamma^T(\theta_{k|k-1}^{(i)}) \end{aligned} \quad (14)$$

where $\theta_{k|k-1}^{(i)}$ is estimation mean value of the parameter particle values $\theta_k^{(i)}$ given by (5).

$y_{k|k-1}^{(i)}$ is the estimation mean value of $y_k^{(i)}$ given by (7) and it is calculated by:

$$y_{k|k-1}^{(i)} = H(\theta_{k|k-1}^{(i)})x_{k|k-1}^{(i)} \quad (15)$$

$R_k^{(i)}$ denotes the estimation standard deviation of the parameter and it is derived by:

$$R_k^{(i)} = H(\theta_{k|k-1}^{(i)})P_{k|k-1}^{(i)}H^T(\theta_{k|k-1}^{(i)}) + Q_v \quad (16)$$

The weights for the parameter and state particles are calculated and normalized:

$$\tilde{w}_k^{(i)} = p(y_k | Y_{0:k-1}, \theta_{k|k-1}^{(i)}) \sim \mathcal{N}(y_{k|k-1}^{(i)}, R_k^{(i)}) \quad (17)$$

$$w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^{N_k} \tilde{w}_k^{(j)}} \quad (18)$$

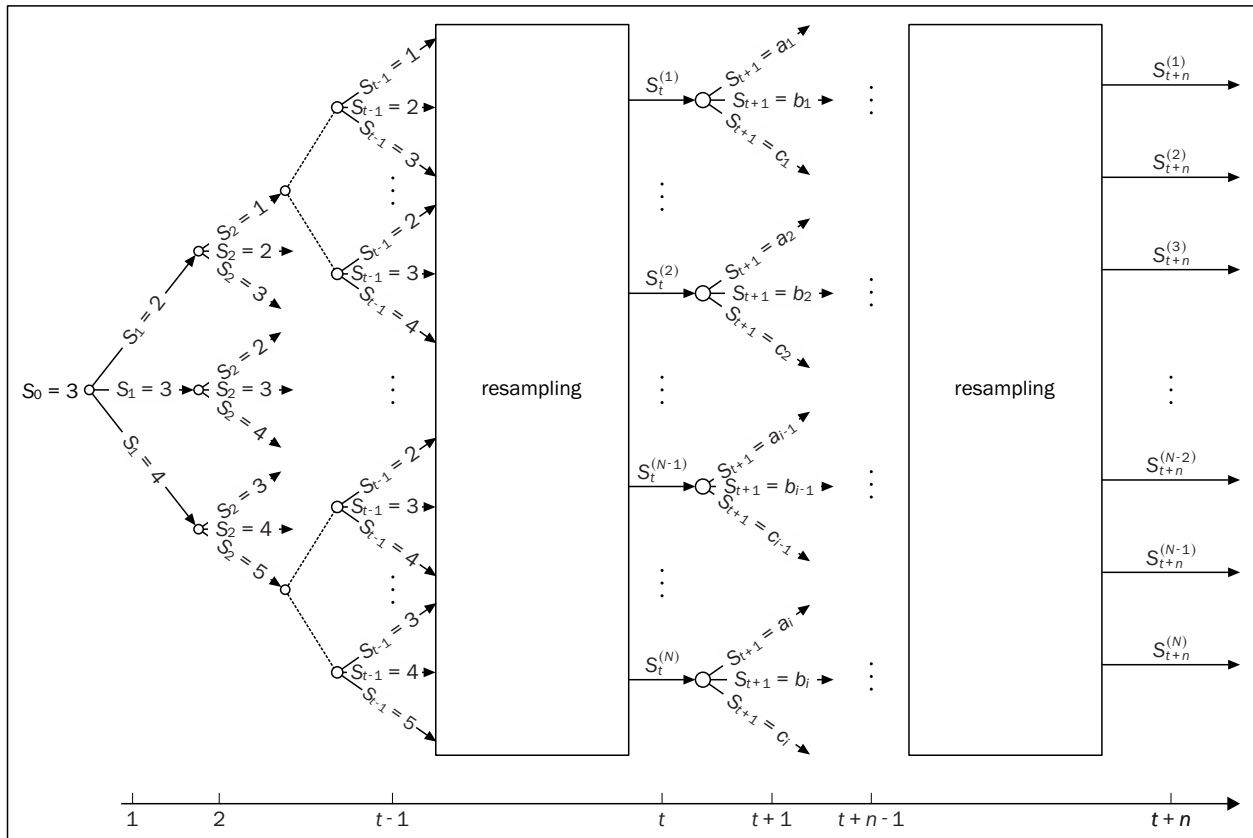


Figure 5 - The filter process of GSMA algorithm from moment 1 to $t+n$

The mean values of the parameters and states are calculated by:

$$\hat{\theta}_k = \sum_{i=1}^{N_k} w_k^{(i)} \theta_{k|k-1}^{(i)} \quad (19)$$

where N_k means the number of the particles at moment k . Kalman gain $K_k^{(i)}$ is calculated by:

$$K_k^{(i)} = P_{k|k-1}^{(i)} H^T (\theta_k^{(i)}) R_k^{-1(i)} \quad (20)$$

The state particle $x_k^{(i)}$ of the system and the mean square error matrix $P_k^{(i)}$ are updated by observation value y_k from (7):

$$\begin{aligned} x_k^{(i)} &= x_{k|k-1}^{(i)} + K_k^{(i)} (y_k - H(\theta_k^{(i)}) x_{k|k-1}^{(i)}) \\ P_k^{(i)} &= P_{k|k-1}^{(i)} - K_k^{(i)} R_k^{(i)} K_k^{T(i)} \end{aligned} \quad (21)$$

The mean values of the states are calculated by:

$$\hat{x}_k = \sum_{i=1}^{N_k} w_k^{(i)} x_k^{(i)} \quad (22)$$

The group, in which the parameter particle exists must be determined before the sampling action and the particle is extended within the group.

4. SIMULATION RESULTS

The RBPF algorithm, SMA algorithm and GSMA algorithm are used for the joint estimation of the train braking rate and the states of the urban train. The state space equation of the simulation system is described in (4). The static parameters τ and λ are set as $\tau = 0.4s$, $\lambda = 1.6667$. The Ω_k is the value of the Gaussian at moment k with expected value $E[\Omega_k] = 0$ and variance $\text{Var}[\Omega_k] = 0.05$.

The braking rate f in (4) represents the system parameter and displacement D , velocity v , control acceleration \hat{a} . The filter value of the target acceleration a_z denotes the estimated status of the system.

The initial condition of the system parameter and states are defined as follows:

(1) It is assumed that the braking rate of the urban railway changes as:

$$\begin{aligned} f_k &= f_0 - 0.005 \times k \\ f_0 &= 1 \end{aligned} \quad (23)$$

where initialization value f_0 is set as 1.0. The scope of the braking rate f is $[0, 1]$ and f , τ and λ are used to calculate the "real" state value $x = [D \ v \ \hat{a} \ a_z]^T$ by (4).

(2) For the displacement D , velocity v , control acceleration \hat{a} and intermediate variable a_z in the state space, the initial values are 0 m, 20 m/s, (-1) m/s² and (-2) m/s².

The braking rate v is discrete as

$$\{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0\}.$$

These values are not calculated from (23). The values of parameter particles will be assigned from them.

For GSMA, SMA and RBPF, the initial states particles $x_0^{(i)}$ for D , v , \hat{a} and a_z are the same. They are sampled on the normal distribution with the mean value 0, 20, -1 and -2, respectively. The initial particles

of $P_0^{(i)}$ are sampled on the normal distribution with variance 0.001;

(1) In the GSMA algorithm, the initial parameter particles $\theta_{1|0}^{(i)}$ for braking rate f are sampled in the discrete space

$$\{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0\}.$$

Discrete values are grouped as

$$\begin{aligned} &\{0.9, 0.8, 0.7\}, \{0.8, 0.7, 0.6\}, \{0.7, 0.6, 0.5\}, \\ &\{0.6, 0.5, 0.4\}, \{0.5, 0.4, 0.3\}, \{0.4, 0.3, 0.2\}, \\ &\{0.3, 0.2, 0.1\}, \{0.2, 0.1, 0.0\}. \end{aligned}$$

The initial numbers of parameter and state particles are both represented by N_0 and they are both set as 50. The boundary maximum upper limit particle number Q and the re-sampling particle number N are set as 500 and 50, respectively.

(2) In SMA, the initial parameter particles $\theta_{1|0}^{(i)}$ for braking rate f are sampled in the discrete space

$$\{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0\}.$$

The initial numbers of parameter and state particles are both represented by N_0 and they are both set as 50; Q and N are set as 500 and 50, respectively.

(3) For RBPF, the initial parameter particles $\theta_{1|0}^{(i)}$ for braking rate f are sampled in the discrete space

$$\{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0\}.$$

The initial numbers of parameter and state particles are both represented by N_0 and they are both set as 250.

During the estimation calculating procedure, the discrete time integration is 1.0 second. The values of the observations y are generated based on the system state space equations by the simulation tool MATLAB. The hardware is PC ThinkPad R400, with Intel T6570, 2.1GHz and 2GB DDR3 memory.

In (19) and (22), $\hat{\theta}_k$ represents braking rate f and $\hat{x}_k = [D \ v \ \hat{a} \ a_z]^T$. Thus f , D , v , \hat{a} and a_z can be updated and calculated by the observations generated in (23) and (4). The estimation errors of the parameters and states, calculated by (23) and (4) are shown from Figure 6 to Figure 10. The corresponding estimated mean errors are specified in Table 1.

Table 1 - Comparison of the parameter and status estimation using RBPF algorithm, SMA and GSMA

The estimation mean error	Different algorithms		
	RBPF algorithm	SMA	GSMA
Train braking rate f	0.105	0.027	0.023
Displacement D (m)	0.305	0.304	0.304
Velocity v (m/s)	0.185	0.181	0.180
Control acceleration \hat{a} (m/s ²)	0.095	0.037	0.036
Derived a_z from the target acceleration (m/s ²)	0.0001	0.0001	0.0001

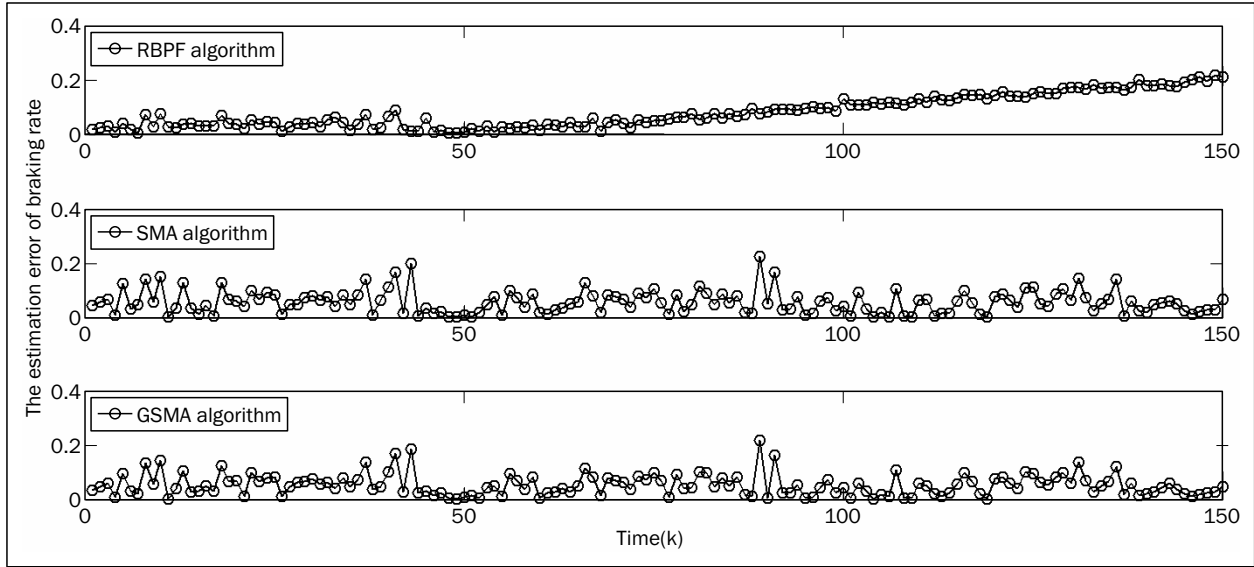


Figure 6 - Comparison of the estimation error of braking rate f with different algorithms

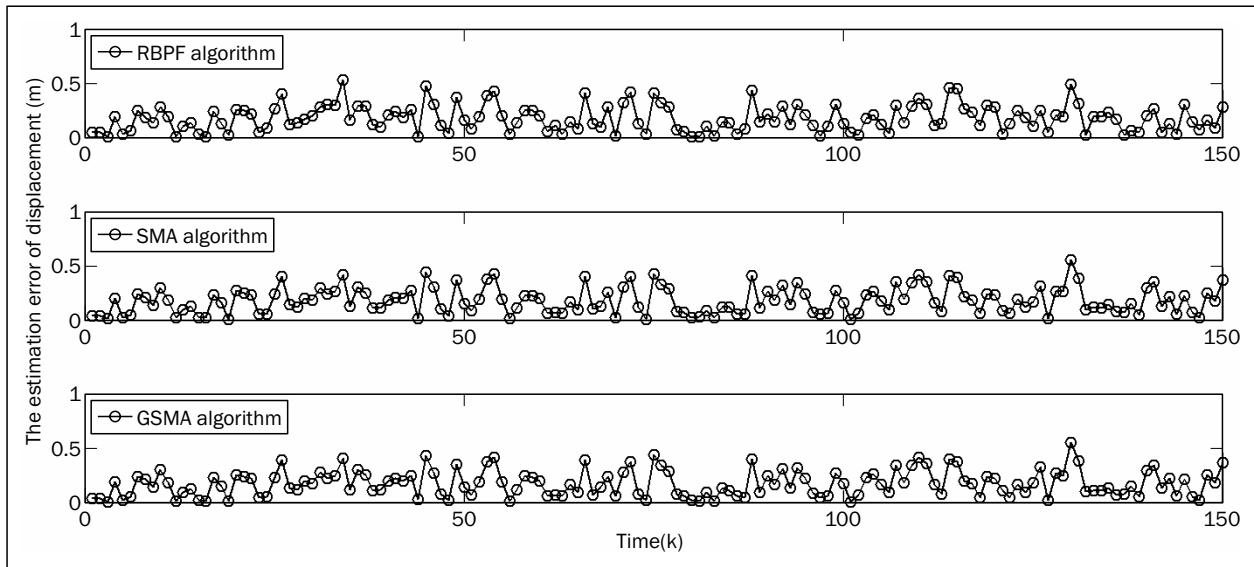


Figure 7 - Comparison of the estimation error of displacement D with different algorithms

Table 2 - Comparison of computing time using RBPf algorithm, SMA and GSMA

Different algorithm	RBPf algorithm	SMA	GSMA
Computation time(s)	6.590	56.505	16.760

In (4), y is observable output and $y = [D \ v \ 0 \ a_z]^T$. Displacement D of the train is measured by displacement sensor and velocity v is calculated from the displacement at the moment. The a_z is derived from the target acceleration and can be calculated from the system input $a_{Tar}(t)$.

From the system parameter and status shown in Figure 7, Figure 8, Figure 10 and in Table 1, the error curves of urban train displacement D , velocity v and a_z are almost equal because these three variables are observable. Figure 6 and Figure 9 show that the estima-

tion precisions of the train braking rate f and the control acceleration \hat{a} using GSMA and SMA are similar. The estimation accuracy of GSMA and SMA are much higher than that of RBPf algorithm. The computation times of the GSMA, SMA and the RBPf algorithm in 150 recursive steps are shown in Table 2. The real-time performance of the GSMA is better than that of SMA and weaker than that of RBPf. Although the RBPf has the best real-time estimation, its estimation error for the braking rate and control acceleration is relatively larger, which will bring about the unexpected hidden danger during the train operation process. Compared with RBPf, GSMA has the better estimation precision and its computation time is only one third of the time consumed by SMA. Taking the real-time performance and the estimation precision into account together,

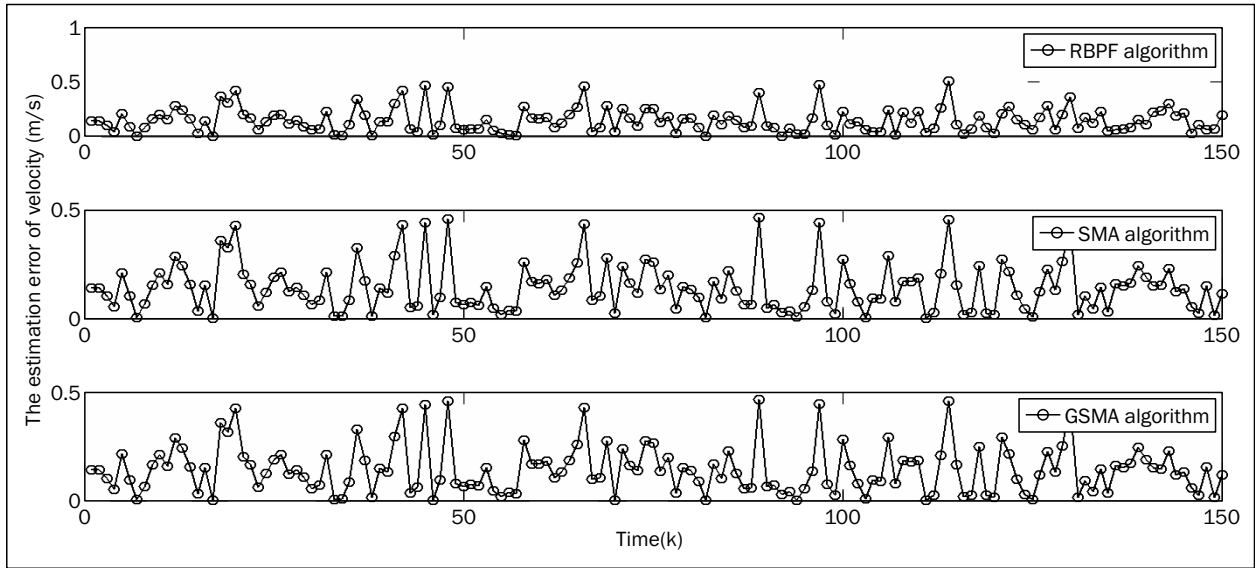


Figure 8 - Comparison of the estimation error of velocity v with different algorithms

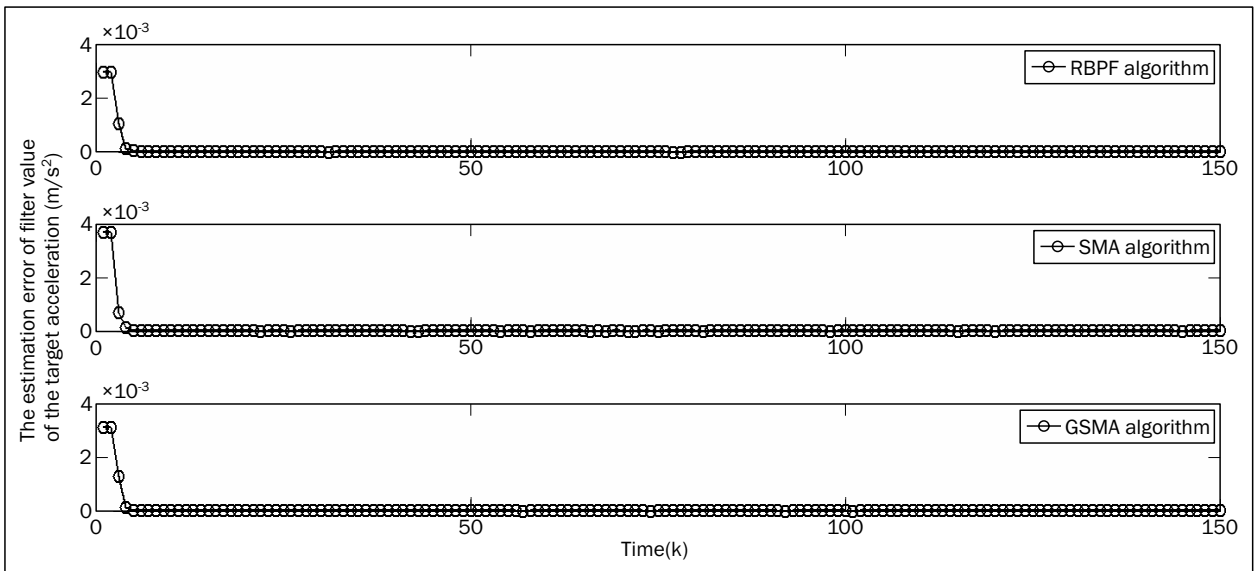


Figure 10 - Comparison of the estimation error of derived z_a from the target acceleration with different algorithms

the GSMA is most suitable for the joint estimation of the train braking rate and the train operation states.

5. DISCUSSION AND CONCLUSION

In order to improve the parking precision of the urban train, the GSMA is proposed, based on the RBPf algorithm and SMA. Adequate estimation for the dynamic braking rate, the displacement and velocity measurement accuracy are the key factors affecting the parking precision. Compared with the RBPf and SMA, GSMA reduces the estimation error of braking rate greatly so that the parking precision of the urban train is improved. In addition, GSMA keeps the real-time performance by decreasing the number of the calculating particles during the estimation process.

In general, the braking rate of urban train changes occasionally and slowly so that the longer parameter estimation time of GSMA, compared with RBPf, is tolerable.

In the urban railway braking domain the benchmark is the accuracy requirement, so that the proposed GSMA takes precedence on the accuracy requirements. If the range of the unknown parameter is wide and the number of the discrete values for the discrete parameter is big, the computation time advantage of the GSMA will become more apparent, because the computation complexity of the proposed algorithm is only associated with the number of discrete values in one group. Whether the unknown parameter is static or dynamic, GSMA can obtain accurate estimation result if the range of the parameter is known in advance.

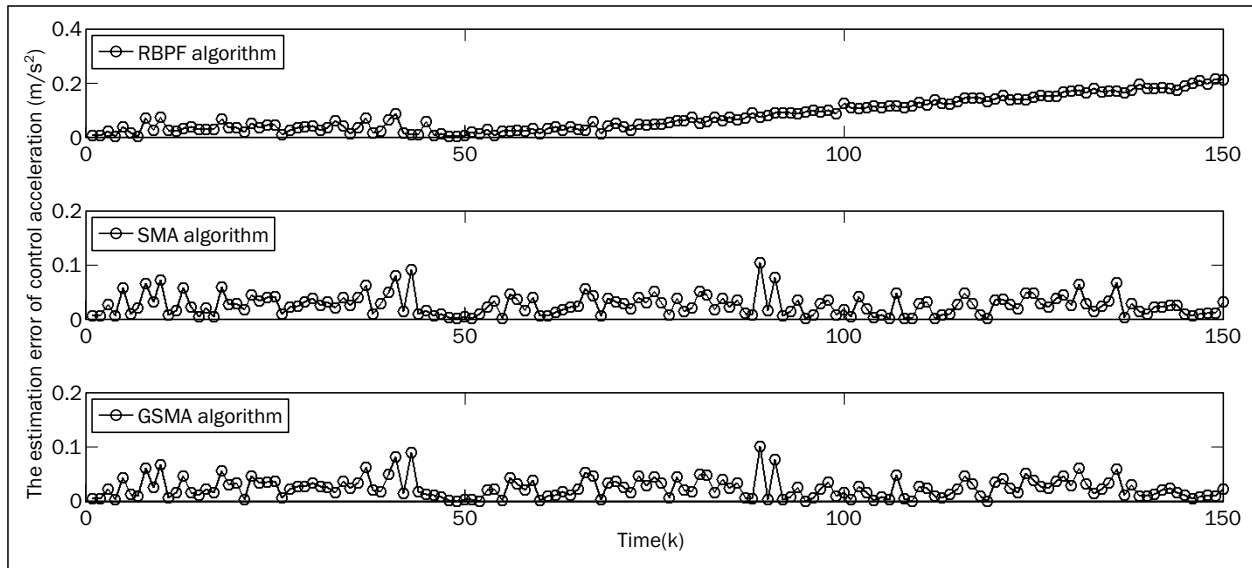


Figure 9 - Comparison of the estimation error of control acceleration \hat{a} with different algorithms

If there is little knowledge of the unknown parameter, it is difficult to use the GSMA.

In the future work, the GSMA will be improved for estimation of the unknown parameter with little a priori knowledge by combining it with an algorithm that can determine the range of the parameter.

ACKNOWLEDGMENT

This paper has been partly supported by the “International Science & Technology Cooperation Program of China” (2012DFG81600) of the Ministry of Science and Technology of the People’s Republic of China.

郑伟，通讯作者，北京交通大学 轨道交通运行控制系统国家工程研究中心。北京市海淀区西直门外上园村3号。邮编：100044。邮箱：wzhengl@bjtu.edu.cn
韩娟，北京交通大学 轨道交通运行控制系统国家工程研究中心。北京市海淀区西直门外上园村3号。邮编：100044。邮箱：juanhao066@126.com
孔维杰，北京交通大学 轨道交通运行控制系统国家工程研究中心。北京市海淀区西直门外上园村3号。邮编：100044。邮箱：11120273@bjtu.edu.cn
王利祥，北京交通大学 轨道交通运行控制系统国家工程研究中心。北京市海淀区西直门外上园村3号。邮编：100044。邮箱：13125105@bjtu.edu.cn

摘要

基于GROUP-SMA 算法的列车参数及状态联合估计

列车制动率及制动运行是影响列车制动性能的关键因素。由于测量噪声的存在及较长的计算时间，列车的若干状态很难实时获得。本文基于Rao-Blackwellization Particle Filter (RBPf)算法及Stochastic M-algorithm (SMA)提出了一种Group Stochastic M-algorithm (GSMA)算法。与RBPf相比，GSMA算法针对列车制动率及控制减速度的估计精度分别提高了78% 和 62%。与SMA相比，GSMA算法的计算时间减少了70%。

关键词

参数估计；状态估计；粒子滤波；列车制动系统；

REFERENCES

- [1] Liu J, West M. Combined parameter and state estimation in simulation-based filtering. New York: Springer-Verlag; 2001.
- [2] Gene HG, Charles FL. An analysis of the total least squares problem. Journal on Numerical Analysis. 1980 Dec;17(6):883-893.
- [3] Felsenstin J. Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of Molecular Evolution. 1981 Nov; 17(6):368-376.
- [4] Shumway RH, Stoffer DS. An approach to time series smoothing and forecasting using the EM algorithm. Journal of Time Series Analysis. 1982 July; 3(4):253-264.
- [5] Morshed J, Kaluarachchi JJ. Parameter estimation using artificial neural network and genetic algorithm for free-product migration and recovery. Water Resources Research. 1998 May; 34(5):1101-1113.
- [6] Yao L, Sethares WA. Nonlinear parameter estimation via the genetic algorithm. IEEE Trans. Signal processing. 1994 Apr; 42(4):927-935.
- [7] Ngoduy D. Low-rank unscented Kalman filter for free-way traffic estimation problems. Transportation Research Record: Journal of the Transportation Research Board. 2012 Jan; 2260(13):113-122.
- [8] T, Bar-Shalom Y, Pattipati KR, Kadar I. Ground target tracking with variable structure IMM estimator. IEEE Trans. Aerospace and Electronic Systems. 2000 Jan; 36(1):26-46.
- [9] Metropolis N, Ulam S. The Monte Carlo method. Journal of the American Statistical Association. 1949 Sep; 44(247):335-341.
- [10] Doucet A, de Freitas N, Gordon NJ, et al. An introduction to sequential Monte Carlo methods. New York: Springer-Verlag; 2001.

- [11] Billot R, Faouzi NE, Sau J, De Vuyst F. Integrating the impact of rain into traffic management. *Transportation Research Record: Journal of the Transportation Research Board*. 2010 Dec; 2169(15):141-149.
- [12] Wei T, Huang YF, Chen CLP. Adaptive sensor fault detection and identification using Particle Filter algorithms. *IEEE Trans. on Systems, Man and Cybernetics, part C: Applications and Reviews*. 2009 Feb; 39(2):201-213.
- [13] Dong HY, Cao B, Yang YP. Application of Particle Filter for target tracking in wireless sensors networks. *International Conference on Communications and Mobile Computing*; 2010 Apr 12-14; Shenzhen, China. Los Alamitos: IEEE Computer Society Press; 2010.
- [14] Casella G, Robert CP. Rao-Blackwellization of sampling schemes. *BIOMETRIKA*. 1996 Jan; 83(1):81-94.
- [15] Akashi H, Kumamoto H. Random sampling approach to state estimation in switching environments, *Automatica*. 1977 July; 13(4):429-434.
- [16] Olsson J, Ryden T. Rao-Blackwellization of particle Markov Chain Monte Carlo methods using forward filtering backward sampling. *IEEE Trans. Signal Processing*. 2011 July; 59(10):4606-4619.
- [17] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics Computing*. 2000 July; 10(3):197-208.
- [18] Kong A, Liu JS, Wong WH. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*. 1994 Aug; 89(425):278-288.
- [19] Doucet A, de Freitas N, Murphy K, Russell S. Rao-Blackwellised particle filtering for dynamic Bayesian networks. *Proceedings of the Sixteenth conference on Uncertainty in Artificial Intelligence*; 2000 Jun 30-July 3; Stanford, U.S.A. San Francisco: Morgan Kaufmann Publishers Inc; 2000.
- [20] Yuvapoositano n, P. Reduced-complexity Rao-Blackwellised Particle Filtering for fault diagnosis. *International Symposium on Intelligent Signal Processing and Communications Systems*; 2011 Dec 7-9; Chiang Mai, Thailand. Red Hook: Curran Associates Inc; 2012
- [21] Dearden R, Clancy D. Particle Filters for real-time fault detection in planetary rovers. *NASA Technical Report Server (NTRS)*. 2001 Jan 1 [cited 2013 June 11]. Available from: <http://ntrs.nasa.gov/search.jsp?R=20020002861>
- [22] de Freitas N, Dearden R, Hutter F, et al. Diagnosis by a waiter and a Mars explorer. *Proceedings of the IEEE*. 2004 Mar; 92(3):455-468.
- [23] Dearden R, Willeke T, Simmons R, Verma V, et al. Real-time fault detection and situational awareness for rovers: report on the Mars technology program task. *IEEE Aerospace Conference Proceedings*; 2004 Mar 6-13; Big Sky, U.S.A. New Jersey: IEEE; 2004.
- [24] Huang YF, Zhang JQ, Djuric PM. Bayesian detection for BLAST. *IEEE Trans. Signal Processing*. 2005 Mar; 53(3):1086-1096.
- [25] Anderson JB, Mohan S. Sequential coding algorithms: a survey and cost analysis. *IEEE Trans. Communications*. 1984 Feb; 32(2):169-177.
- [26] Luo RS. Fault-tolerant control technology research of automatic train operation [M.A. thesis]. Beijing: Beijing Jiaotong University; 2011 [cited 2013 June 11]. Available from: Beijing Jiaotong University Library E-Reserve
- [27] Yu ZY, Chen DW. Modeling and system identification of the braking system of the urban rail vehicles [in Chinese]. *Journal of the China Railway Society*. 2011 May; 33(10):37-40.
- [28] Luo RS, Yu ZY, Tang T. Accurate train stopping by model following sliding mode control. *International Conference on Information Technology and Management Science*; 2012 Oct 20-21; Chongqing, China. Berlin: Springer-Verlag Berlin Heidelberg; 2013.

