

多点测试的多模型机动目标跟踪算法

王伟¹ 余玉揆¹

摘要 针对机动目标跟踪问题,提出了一种多点测试多模型粒子滤波算法 (Independence multi-try method, IMTM). 整个算法分为两个阶段,第一阶段为利用多点测试 (Multi-try method, MTM) 结构从各模型产生的粒子中选取一个最优粒子,实现了模型间的交互;第二阶段为利用 IMH (Independence Metropolis-Hastings) 滤波算法对第一阶段产生的粒子进行取舍,完成整个状态估计. 相对于传统的交互式多模型 (Interacting multiple model, IMM) 算法,该算法无需事先设定模型转移概率矩阵且为整体并行结构,结构简单,能够充分地交互各模型之间的粒子,进而自动有效地调整各模型权值比重,降低了人为干扰. 仿真表明,该算法能够有效地降低滤波峰值误差,整体跟踪精度较高,算法的实时性较好.

关键词 机动目标跟踪, 多点测试, 多模型, 并行粒子滤波

引用格式 王伟, 余玉揆. 多点测试的多模型机动目标跟踪算法. 自动化学报, 2015, 41(6): 1201–1212

DOI 10.16383/j.aas.2015.c140471

Multi-try and Multi-model Particle Filter for Maneuvering Target Tracking

WANG Wei¹ YU Yu-Kui¹

Abstract A novel multiple try and multiple model particle filter named independence multi-try method (IMTM) is presented for maneuvering target tracking. It can be divided into two stages: 1) A multi-try test structure is utilized to choose an optimal particle from a set of particles generated by multiple models; this stage brings about model interaction. 2) Independence Metropolis-Hastings (IMH) will reject or accept the particle produced in stage 1. This method does not require the model conversion matrix so that it can reduce the man-made error introduced by the model conversion matrix as used in traditional methods. It is a parallel and simple structure particle filter. With this method, the particles are exchanged between models automatically and effectively to adjust the model weight. Simulations show that this algorithm can effectively reduce the peak-error, and is a real-time algorithm.

Key words Maneuvering target tracking, multi-try method (MTM), multi-model (MM), parallel particle filter

Citation Wang Wei, Yu Yu-Kui. Multi-try and multi-model particle filter for maneuvering target tracking. *Acta Automatica Sinica*, 2015, 41(6): 1201–1212

在实际的滤波环境中,待滤波目标的状态模型可能随着时间而发生变化,或状态模型过于复杂,无法确切地用数学模型描述其状态模型,此时仅使用一个固定的模型来描述运动轨迹将会使得跟踪性能变差甚至滤波发散,主流算法是利用多个模型来描述其数学模型,即多模型 (Multiple model, MM) 算

法^[1]. 对每一个模型进行独立的滤波输出,根据各个模型的权重融合各滤波器的输出. 其中模型间切换的准确度或者说各模型间的权值分配决定了数据融合的精度. 最具代表性的算法为机动目标跟踪中的多模型算法,其利用多个运动模型来描述机动目标的运动轨迹^[2]. 当目标模型和观测模型都为线性或弱非线性时,基于卡尔曼滤波器 (Kalman filter, KF)、扩展卡尔曼滤波器 (Extended Kalman filter, EKF) 等滤波器的 MM 算法 (MM-KF) 可以很好地工作. 但当存在强非线性方程时,此类方法将不再适用^[3].

粒子滤波 (Particle filter, PF) 对非线性环境具有很好的适应性,基于粒子滤波的 MM 算法 (Multiple model particle filter, MM-PF) 可以较好地解决状态模型和观测模型中存在非线性方程的问题^[4–5]. MM-PF 算法根据每个模型的粒子数目的不同一般可分为两大类:一类为每个模型中的粒子数目为时变的 MM-PF 算法,其根据一个假定的先验已知的马尔科夫转移概率矩阵来调整每个模型中

收稿日期 2014-07-01 录用日期 2014-12-31
Manuscript received July 1, 2014; accepted December 31, 2014
新世纪优秀人才支持计划 (NCET-11-0827), 中央高校基本业务费专项资金 (HEUCFX41308), 中国博士后科学基金 (2014M550182), 黑龙江省博士后特别资助资金 (LBH-TZ0410), 哈尔滨市科技创新人才 (2013RFXXJ016) 资助
Supported by the New Century Excellent Talents Support Program (NCET-11-0827), Fundamental Research Funds for the Central Universities (HEUCFX41308), China Postdoctoral Science Foundation (2014M550182), Heilongjiang Postdoctoral Special Fund (LBH-TZ0410), and Innovation of Science and Technology Talents in Harbin (2013RFXXJ016)

本文责任编辑 文成林
Recommended by Associate Editor WEN Cheng-Lin
1. 哈尔滨工程大学自动化学院无线电与卫星定位应用技术实验室 哈尔滨 150001
1. Application Technique in Radio and Satellite Positioning System Laboratory, College of Automation, Harbin Engineering University, Harbin 150001

的粒子数目,各模型所分配的粒子数目代表了各模型的权重,结构较为简单,当运动模型不发生变化时,大量的粒子服从正确模型,估计精度较高;但当模型发生转变时,由于大量的粒子还处于上一时刻的模型分布中,符合转换后模型分布的粒子较少,而较少的粒子将无法准确描述模型分布,导致模型切换时估计精度的降低,甚至算法发散^[6],且对于硬件实现来说,时变的粒子数目将是致命的^[7].另一类为每个模型中的粒子数目为固定不变的交互式粒子滤波算法(Interacting multiple model particle filter, IMM-PF),各个模型的滤波器独立地输出状态估计和模型似然函数,结合马尔科夫转移概率矩阵对模型权重进行动态调整^[8-10],并对各个滤波器的输出进行加权,虽然各滤波器的粒子数目不变,但其结构复杂、计算量大、实时性差.

对于大部分 MM 算法而言,不同模型间的转移被认为是一个马尔科夫过程,由于无法准确地确定模型间的马尔科夫转移概率,MM 算法假定这个马尔科夫转移概率矩阵为一个先验已知的,这样必然引入人为干扰因素对机动目标跟踪精度的影响.多模型粒子滤波算法中,算法的并行结构仅存在于各个滤波器之间,而单个滤波器内部并非为并行结构,不利于算法的流水线实现,算法的实时性较差.现存的大部分并行结构粒子滤波算法中并行模块之间仅存在一个简单的粒子交换过程^[11-12],而无法根据外界变化而动态的调整各模型中的粒子,并不适合如机动目标跟踪之类的领域中应用.在目标跟踪领域,文献 [7] 给出了一种基于 IMH (Independence Metropolis-Hastings) 的并行结构粒子滤波算法,但在粒子交互阶段,仅仅简单地认为各模型所贡献的粒子数与这个模型的整体权重成正比,在目标机动性高或者模型较多时,算法极易发生发散.

本文提出了一种新的基于多点测试的多模型机动目标跟踪算法 (Independence multi-try method, IMTM),其滤波的主体部分为 Metropolis 化独立抽样粒子滤波 (IMH),为一种基于马尔科夫链蒙特卡罗 (Markov chain Monte Carlo, MCMC) 采样的重采样技术^[13],可以在获得粒子及其权重后立即开始进行重采样,无需等待所有粒子的产生,避免了基本粒子滤波算法中的权值归一化的并行瓶颈问题,可以通过并行流水线操作减少延时,提高算法的实时性.而在模型的切换过程中,本文在粒子滤波重采样阶段利用多点测试的方法 (Multi-try method, MTM) 对各个模型产生的粒子进行选择,自动地调整各模型的粒子数目进而调整模型的权值,无需事先假定一个先验已知的马尔科夫转移矩阵来实现模型交互,有效地降低了人为因素对机动目标跟踪精度的影响,运算量得到了降低.采用 MTM 算法

使得粒子滤波所需存储的粒子数目降低到了原先的 $1/M$ (M 为模型数).

统计学中,MTM 算法在多个建议点中选择一个好的建议点,使得马尔科夫链朝着一些选定好的方向做大步移动,可使得 MCMC 抽样在不降低接受率的情况下做大步跳跃进而提高算法的混合效率,这与多模型算法中从多个模型中选择一些更接近目标运动轨迹的模型并赋予其更大权值的原理相同.传统统计学中,MTM 算法各个试验建议点的产生均为从同一个建议分布中产生,并在同分布的试验点集中选择一个更好的,并不适合多模型算法中的多个建议分布^[14];后有学者提出了从多重相关的建议分布中产生试验点^[15-17];文献 [18] 虽然提出了一种从多重独立的建议分布中产生试验点的 MTM 结构,但其与现存的多重相关的建议分布的 MTM 算法一样,在选择完试验点后需根据此试验点反产生一个“参照集”(Reference set) 进行选择,算法结构较为复杂. Liu 提出了一种基于 IMH 算法的 MTMIS 算法^[13],由于试验样本为独立产生,故无需产生“参照集”.本文将 IMH 算法与 MTM 算法结合提出的算法 IMTM 将同样无需产生“参照集”,算法结构简单,易于实现.需要注意的是,MTMIS 算法试验建议点的产生为从同一个建议分布中产生,而本文是从多重独立建议分布(多模型)中产生.同时为了使得 IMTM 算法中的建议分布更加符合粒子滤波结构,本文从上一时刻的粒子中随机选取粒子作为各个建议分布的输入值,从而与现存的 MTM 算法中各个建议分布的输入都为同一个粒子不同,这样可以使得试验建议点的选择更加有效,滤波效果更好,并给出了其细节平衡方程的证明.

1 问题描述

假设目标状态模型可由 M 个模型动态表述,则多模型的状态转移方程和量测方程可表示为

$$\mathbf{x}_{k+1}^r = F(\mathbf{m}_r)\mathbf{x}_k + G(\mathbf{m}_r)\mathbf{u}_k(\mathbf{m}_r) \quad (1)$$

$$\mathbf{z}_{k+1}^r = H(\mathbf{m}_r)\mathbf{x}_{k+1}^r + \mathbf{v}_{k+1}^r, \quad r = 1, \dots, M \quad (2)$$

其中, \mathbf{x}_k 表示时刻 k 的状态向量, \mathbf{x}_{k+1}^r 表示 $k+1$ 时刻模型 \mathbf{m}_r 的状态向量,状态转移矩阵 F 、过程噪声矩阵 G 以及观测矩阵 H 都与模型 \mathbf{m}_r 有关,过程噪声 \mathbf{u}_k 和量测噪声 \mathbf{v}_k 同样与模型相关.

在贝叶斯框架下,目标跟踪的目的为计算状态变量 \mathbf{x}_k 的后验估计值.设 $\mathbf{y}_k = [\mathbf{x}_k; \mathbf{M}_k]$, 利用贝叶斯公式可得:

$$p(\mathbf{y}_k | \mathbf{z}_{k-1}) = \int p(\mathbf{y}_k | \mathbf{y}_{k-1})p(\mathbf{y}_{k-1} | \mathbf{z}_{k-1})d\mathbf{y}_{k-1} \quad (3)$$

$$p(\mathbf{y}_k|\mathbf{z}_k) = \frac{p(\mathbf{z}_k|\mathbf{y}_k)p(\mathbf{y}_k|\mathbf{z}_{k-1})}{\int p(\mathbf{z}_k|\mathbf{y}_k)p(\mathbf{y}_k|\mathbf{z}_{k-1})d\mathbf{y}_k} \quad (4)$$

其中, $p(\cdot|\cdot)$ 代表条件概率密度函数分布. 式 (4) 中的变量除了积分量 $\int p(\mathbf{z}_k|\mathbf{y}_k)p(\mathbf{y}_k|\mathbf{z}_{k-1})d\mathbf{y}_k$ 为未知, 其余都为已知量. 但只有当状态方程为有限方程或者线性高斯时, 积分量才可利用卡尔曼滤波方法解析地求出. 对于其他无法得到解析解的情况, 一般利用蒙特卡洛仿真的方法近似计算其后验概率密度函数, 即可采用粒子滤波算法.

粒子滤波利用一堆带有权值的粒子 $\{\mathbf{y}_k^i, w_k^i\}_{i=1}^N$ 构建后验概率密度函数 $p(\mathbf{y}_k|\mathbf{z}_k)$. 式 (4) 可写为如下形式:

$$p(\mathbf{y}_k|\mathbf{z}^k) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{y}_k - \mathbf{y}_k^i) \quad (5)$$

其中, w_k^i 代表粒子 \mathbf{y}_k^i 的权重, $\delta(\cdot)$ 为狄拉克函数.

传统的多模型粒子滤波算法 IMM-PF 一般将变量 \mathbf{x}_k 和模型变量 \mathbf{M}_k 分开计算, 即:

$$p(\mathbf{y}_k|\mathbf{y}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{M}_k) \Pr(\mathbf{M}_k|\mathbf{M}_{k-1}) \quad (6)$$

其中, $\Pr(\mathbf{M}_k|\mathbf{M}_{k-1})$ 为事先人为假设的模型转移概率. 而联合分布 $p(\mathbf{y}_k|\mathbf{z}_k)$ 可写为

$$p(\mathbf{x}_k, \mathbf{M}_k = \mathbf{m}|\mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{M}_k = \mathbf{m}, \mathbf{z}_k) \Pr(\mathbf{M}_k = \mathbf{m}|\mathbf{z}_k) \quad (7)$$

传统的多模型粒子滤波算法具体的理论推导过程可参见文献 [19].

2 多点测试多模型粒子滤波算法

多点测试多模型粒子滤波算法 IMTM 为基于多点测试 (MTM) 的 IMH 算法. 与传统多模型粒子滤波算法不同的是, 在计算后验概率密度函数 $p(\mathbf{y}_k|\mathbf{z}_k)$ 时, 并不将变量 \mathbf{x}_k 和模型变量 \mathbf{M}_k 分开计算, 而是对每一个模型输出的值利用 MTM 算法进行选择, 这样就避免了 $\Pr(\mathbf{M}_k|\mathbf{M}_{k-1})$ 项, 即无需人为设定模型转移概率, 避免了人为干扰因素. 最后得到的粒子集合为对整个后验概率密度函数分布的估计, 而非传统多模型粒子滤波算法中每一个模型对应一个粒子集合, 这样使得 IMTM 算法极大地交互了各模型之间的粒子, 实现了一种混合采样过程.

2.1 IMH 算法

IMH 算法是一种特殊的 MH (Metropolis-Hastings) 算法, 其建议转移函数 $T(x, y)$ 被认为是一个独立实验密度 $g(y)$, 从而使得 $g(\cdot)$ 产生的建议移动点 y 独立于前一个状态 x^t .

在信号处理中的 IMH 算法主要为如下两步:

步骤 1. 用贯序的方法从 $g(\cdot)$ 中独立地产生一潜在的转移点 y 并计算其权值:

$$w(y) = p(y_t|x_t) \frac{\sum_{i=1}^N p(x_t|x_{t-1}^i)}{\sum_{i=1}^N g(x_t|x_{t-1}^i, y_t)} \quad (8)$$

步骤 2. 以概率 $\min\{1, w(y)/w(x_k)\}$ 接受 y , 即 $x_{k+1} = y$, 以剩余概率拒绝 y .

IMH 算法中, $p(x_t|x_{t-1})$ 为先验概率密度函数. 若建议分布取为先验概率密度函数, 即 $g(\cdot) = p(x_t|x_{t-1})$, 则可以将权值计算中的累加和项去除, 极大地简化整个计算过程, 式 (8) 变为

$$w(y) = p(y_t|x_t) \quad (9)$$

可以看出, IMH 算法在粒子的权值计算完后即可进行粒子的取舍, 避免了传统的粒子滤波算法需要所有粒子的权值计算完毕后才可进行重采样的并行瓶颈问题^[11].

2.2 IMTM 算法

本文提出的算法为基于 IMH 算法的多点测试多模型粒子滤波算法, 利用多点测试 MTM 实现了各个模型粒子之间的交互, 自动完成了模型间的权值分配, 无需像传统的多模型算法那样需要人为设定一个马尔科夫矩阵. 算法的具体步骤如下:

步骤 1. 初始化 $k = 0$, 利用 SIS 策略从初始建议分布 $q(x_0)$ 中抽取得到 $x_0^{1:N}$, 并计算权 $w_0^{1:N}$. 其中初始建议分布 $q(x_0)$ 由 M 个模型的初始建议分布 $q(x_0^r)$, $r = 1, \dots, M$ 按等比例构成, 抽取粒子时, 从每个建议分布 $q(x_0^r)$ 中抽取 N/M 个粒子 $\{x_{r,0}^{1:N/M}, w_{r,0}^{1:N/M}\}$, 构成的粒子集 $\{x_{r,0}^{1:N/M}, w_{r,0}^{1:N/M}, r = 1, \dots, M\}$ 为 $\{x_0^{1:N}, w_0^{1:N}\}$.

步骤 2. 从 $[1, 2, \dots, N]$ 中等概率产生 $M \times (N + N_b)$ 个标志量 $J(i)$, $i = 1, 2, \dots, (M \times N_s)$, 其中 N_b 为 IMH 链 burn-in 阶段所需的粒子数, $N_s = N + N_b$. 设 $k-1$ 时刻的粒子集合 $S_{k-1} = \{x_{k-1}^i, i = 1, 2, \dots, N\}$ 中根据每个模型的试验建议分布的不同, 利用 $x_k^{(r,i)} \sim q^r(\cdot|x_{k-1}^{J(i+r \times N_s)}, y_k)$, $i = 1, \dots, N_s$, 抽取各个模型的潜在转移粒子, 其中 $q^r(\cdot)$, $r = 1, \dots, M$ 为选取的建议分布, 利用下式计算各个粒子的权值:

$$w(x_k^{(r,i)}) = w(x_{k-1}^i) \frac{\pi(x_k^{(r,i)})}{q^r(x_k^{(r,i)})} \quad (10)$$

并计算权值总和:

$$W_k^i = \sum_{r=1}^M w(x_k^{(r,i)}) \quad (11)$$

抽取的各模型粒子构成了多重独立分布的试验点集 $\{\{x_k^{(1,i)}, w(x_k^{(1,i)})\}, \dots, \{x_k^{(M,i)}, w(x_k^{(M,i)})\}\}$.

步骤 3. 以与 $w(x_k^{(r,i)})$ 成比例的概率从试验点集 $\{\{x_k^{(1,i)}, w(x_k^{(1,i)})\}, \dots, \{x_k^{(M,i)}, w(x_k^{(M,i)})\}\}$ 中抽取 \tilde{x}_k^i , 假设 $\tilde{x}_k^i = x_k^{(r,i)}$.

步骤 4. 以概率 $\min\{1, W_k^i/(W_k^i - w(\tilde{x}_k^i) + w(\hat{x}_k^{i-1}))\}$ 令 $\hat{x}_k^i = \tilde{x}_k^i$, 以剩余概率令 $\hat{x}_k^i = \hat{x}_k^{i-1}$, 其中 \hat{x}_k^{i-1} 为上一个已接受的粒子.

步骤 5. 利用步骤 4 得到的 N_s 个粒子 \hat{x}_k^i 构成粒子集 $\hat{S}_k = \{\hat{x}_k^i, w_k^i, i = 1, 2, \dots, N_s\}$. 由于 IMH 链在 burn-in 阶段并未达到平衡, 即 burn-in 阶段产生的 N_b 个粒子为过渡粒子, 无法描述目标分布, 需剔除以提高精度, 故去除 \hat{S}_k 中的前 N_b 个过渡粒子, 得到 k 时刻的粒子集合 $S_k = \{x_k^i, w_k^i, i = 1, 2, \dots, N\}$, 其估计输出为

$$\hat{I}_k = \frac{1}{N} \times \sum_{i=1}^N x_k^i \quad (12)$$

此算法的马尔科夫链的细节平衡的证明如下:

证明. 为了证明的简洁性, 本文定义如下符号: 后验概率分布 $p(x|z)$, 记为 $\pi(x) = p(x|z)$, 试验点集 $\{y_1, y_2, \dots, y_M\} = \{x_k^{(1,i)}, x_k^{(2,i)}, \dots, x_k^{(M,i)}\}$, 试验分布 $q^r(y_r|x_{k-1}, z)$, 记为 $g(y_r) = q^r(y_r|x_{k-1}, z)$, 权值 $w_r(y_r) = \pi(y_r)/g_r(y_r)$.

假定 $A(x, y)$ 为算法中从当前状态 x 到选中的试验点 $y = y_{r=m}$ 的真实转移概率, 则选中第 m 个试验点, 其满足如下细节平衡方程:

$$\pi(x)A(x, y) = \pi(y)A(y, x) \quad (13)$$

这里我们只考虑 $x \neq y$ 的情况 (若 $x = y$, $A(y, x)$ 为一个狄拉克函数 $\delta(y - x)$, 结论明显成立), 由于 y_r 的可互换性及 $\pi(y) = w(y) * g(y)$, 有式 (14) 成立. 可以看出, 式 (14) 最后一个表达式是关于 x 和 y 对称的, 即平衡条件: $\pi(x)A(x, y) = \pi(y)A(y, x)$ 成立. \square

2.3 算法分析

上一小节给出了整个多点测试多模型粒子滤波算法的步骤, 本节将分别对算法的运行时间和时间复杂度进行分析, 说明 IMTM 算法在实时性方面和计算量方面的优越性.

$$\begin{aligned} \pi(x)A(x, y) &= \pi(x) \times M \times P[(y = y_m)|x] = \\ &M \times \pi(x) \int \dots \int \left[\prod_{r=1}^M g_r(y_r) \right] \times \frac{w(y)}{\sum_{r=1}^M w_r(y_r)} \times \\ &\min \left\{ 1, \frac{\sum_{r=1}^M w(y_r)}{\sum_{r=1}^M w(y_r) - w(y) + w(x)} \right\} dy_1 \dots dy_{m-1} dy_{m+1} \dots dy_M = \\ &M \times \pi(x)g(y)w(y) \int \dots \int \left[\prod_{r=1, r \neq m}^M g_r(y_r) \right] \times \\ &\min \left\{ \frac{1}{\sum_{r=1}^M w_r(y_r)}, \frac{1}{\sum_{r=1}^M w(y_r) - w(y) + w(x)} \right\} dy_1 \dots dy_{m-1} dy_{m+1} \dots dy_M = \\ &M \times \pi(x)\pi(y) \int \dots \int \left[\prod_{r=1, r \neq m}^M g_r(y_r) \right] \times \\ &\min \left\{ \frac{1}{\sum_{r=1, r \neq m}^M w_r(y_r) + w(y)}, \frac{1}{\sum_{r=1, r \neq m}^M w(y_r) + w(x)} \right\} dy_1 \dots dy_{m-1} dy_{m+1} \dots dy_M \quad (14) \end{aligned}$$

2.3.1 算法实时性分析

下面将分析 IMTM 算法的运行时间, 并给出算法的一种并行结构, 以提高算法的实时性. 图 1 为算法的整个流程图.

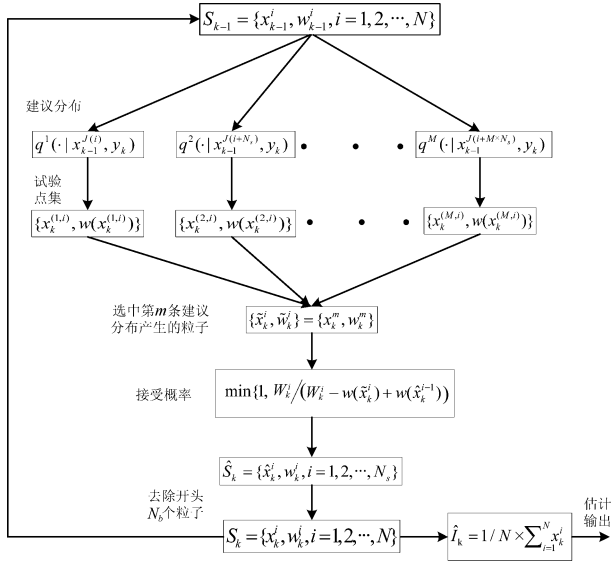


图 1 多点测试的多模型算法框图

Fig. 1 The diagram of multi-try and multi-model particle filter

从图 1 可以看出, 本文提出的基于多点测试的多模型算法 (IMTM) 中试验点集的产生是由多条建议分布并行产生, 若 $T_{P1}, T_{P2}, \dots, T_{PM}$ 分别代表各个建议分布产生粒子的时间, $T_{w1}, T_{w2}, \dots, T_{wM}$ 为权值计算所需时间, 则本算法在试验点集环节运算时间为 $\max\{T_{P1}+T_{w1}, T_{P2}+T_{w2}, \dots, T_{PM}+T_{wM}\}$, 与 IMH 算法中单条建议分布所需时间^[20] 相仿, 整个算法与基本的 IMH 算法相比, 仅仅多出步骤 3 中从试验点集中选取 \tilde{x}_k^i 所需的时间. 步骤 3 的目的为找出满足 $w^{m-1} < W \times rand \leq w^m$ 的 m 的值. 如果采用串行结构进行权值大小寻找的话, 模型或者建议分布数多的情况下将产生较大的延时, 本文提出图 2 结构可以使得模型选择为并行运行, 实时性得到较大提高, 且整个算法的运行时间将不再与模型数 M 相关, 算法具有确定的运行时间, 大概为 5 个时钟周期.

从图 2 可以看出, 本文首先对权值计算环节产生的权值 $\{w^M, \dots, w^2, w^1\}$ 进行并行累加求和得到 $\{\sum_{r=1}^M w^r, \dots, w^1 + w^2, w^1\}$, 其中权值总和 $W = \sum_{r=1}^M w^r$ 乘以一个随机数 $rand$ 后得到 $W \times rand$, 分别与 $\{\sum_{r=1}^M w^r, \dots, w^1 + w^2, w^1\}$ 各项比较, 若 $W \times rand \leq \sum w^r$ 则此项输出 $p_r = 1$, 否则输出 $p_r = 0$, 输出后的 M 个值组成二进制序列 $P = 1p_{M-1} \dots p_2 p_1$ 和 $P' = p_{M-1} \dots p_2 p_1 0$, 将 P

和 P' 异或后得到控制变量 P_{new} , P_{new} 中将只有一个值为 1, 即对应着被选中的试验点. 如 $M = 4$, $P = "1100"$, $P' = "1000"$, 得到控制量 $P_{new} = "0100"$, 即可得 $m = 3$, 选中了第 3 个试验点作为潜在的转移点.

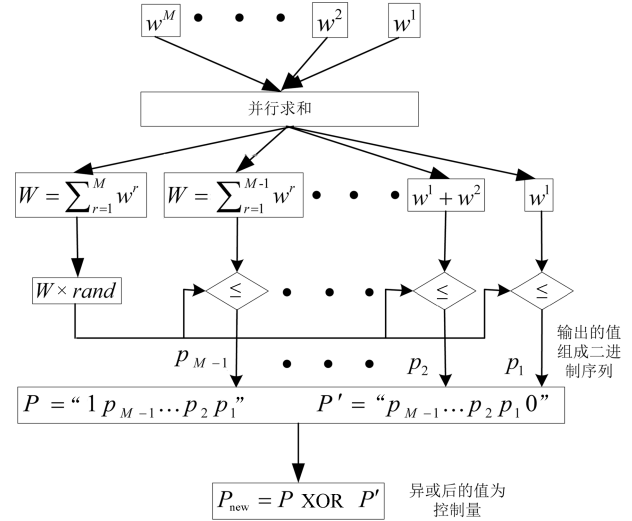


图 2 步骤 3 的工作流程图

Fig. 2 The work flow chart of Step 3

采用上述权值比较结构后, 整个算法为整体并行结构, 整个算法实时性与单 IMH 链实时性相仿, 较基本粒子滤波算法在实时性上有很大的提高. 在机动目标跟踪应用中, 本文选取先验概率密度函数为重要性密度函数, 即 $q(x_k|x_{k-1}, z) = p(x_k|x_{k-1})$, 且在重采样后粒子的权值都赋值为 $1/N$, 这样整个算法无需存储权值.

2.3.2 算法时间复杂度分析

通过与传统多模型粒子滤波算法^[8] 和交互式多模型粒子滤波算法 IMM-PF^[9-10] 的时间复杂度的对比, 本节将说明 IMTM 算法在计算量方面的优越性. 假设 M 为总模型数, N 为每个模型的粒子数, n_x 为状态方程的维数, n_z 为观测方程的维数. 传统的多模型算法 IMM-PF 和 MM-PF 的第一步均为输入交互模块, 以实现各模型粒子之间的交互作用, IMM-PF 输入交互模块的时间复杂度为 $O(M^2 + 2M^2 + M^2 n_x^2 + M^2 n_z^2 + MN(n_x^3 + n_z^2))$, 由于 N 远大于 M 和 n_x , 故可简化为 $O(MNn_x^3)$; MM-PF 算法在输入交互模块的算法是与基本粒子滤波重采样相类似的方法, MM-PF 在输入交互模块的时间复杂度为 $O(MN)$ ^[21]; IMTM 算法避免了输入交互模块, 其模型之间的交互体现在多点测试结构. 在粒子滤波的重要性采样阶段 (即粒子更新阶段), 三种算法的时间复杂度相同, 均为 $O(MNn_x^2)$. 在重采样阶段, 对于采用系统重采样的 IMM-PF 和

MM-PF 算法的时间复杂度为 $O(MN)$, 而采用多点测试结构的 IMTM 算法, 未采用图 2 并行结构时其时间复杂度为 $O((M+1)N)$, 采用并行求和结构后, 其时间复杂度变为 $O((M^2+M+1)N)$, 可简化为 $O(M^2N)$. 可以看出, 算法的实时性的提高是以牺牲复杂度为代价的, 这也是并行结构对硬件实现的更高要求. 在输出阶段, 算法 MM-PF 的时间复杂度为 $O(MNn_x)$, IMM-PF 由于在输出阶段需要输出交互的缘故, 其时间复杂度为 $O(MNn_x^2)$, IMTM 算法输出模块仅需 N 个粒子参与运算, 其复杂度为 $O(Nn_x)$. 相对于 MM-PF 和 IMTM 算法, 算法 IMM-PF 还需要一个模型概率更新模块, 其复杂度为 $O(MNn_x^3)$.

将各模块的时间复杂度相加可得: IMTM 算法的时间复杂度为 $O(MNn_x^2 + M^2N + Nn_x)$, 一般情况下认为 M 和 N 两者处于同一个数量级, 故可简化为 $O(MNn_x^2)$; IMM-PF 算法的时间复杂度为 $O(MNn_x^3 + MNn_x^2 + MN + MNn_x^2 + MNn_x^3)$, 可简化为 $O(Nn_x^3)$; MM-PF 算法的时间复杂度为 $O(MNn_x^2 + MN + MNn_x)$, 可简化为 $O(Nn_x^2)$. 各算法的各模块的时间复杂度总汇于表 1.

从表 1 可以看出, 本文提出的算法 IMTM 较 IMM-PF 算法的复杂度有较大的提高, 使用多点测试结构避免了 IMM-PF 算法中的输入交互模块和概率更新模块而进行模型交互, 算法结构简单. IMTM 算法与 MM-PF 算法复杂度相差不大, 但 MM-PF 算法不具备固定运算时延, 不易于硬件实现.

3 仿真分析

本文对两种机动目标进行仿真验证算法 IMTM 的性能, 对比算法为传统的多模型粒子滤波算法 MM-PF^[8] 和交互式多模型粒子滤波算法 IMM-PF^[9-10]. 第一种机动目标仅含有两种运动模式: 常速运动 CV 和恒加速度 CA 运动^[19], 即 $M=2$, 仿真总时间为 100s. 目标在前 40s 保持零速度运动, 40s 后以恒加速度 σ_a 加速运动. 其可分为四种仿真场景, 场景 1 和场景 2 目标在 40s~60s 保持恒加速度 σ_a , 60s~100s 保持恒速度. 场景 3 和场景 4 目标在 40s~100s 一直保持恒加速度 σ_a . 表 2 为第一种机动目标运动模型的四个仿真场景的参数

设置.

CV 模型和 CA 模型的状态方程和观测方程可写为

$$\begin{bmatrix} x \\ v_x \\ a \end{bmatrix}_k^{CV} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x \\ v_x \\ a \end{bmatrix}_{k-1}^{CV} + \begin{bmatrix} 0 \\ 0 \\ \sigma_a \end{bmatrix}^{CV} \quad (15)$$

$$\begin{bmatrix} x \\ v_x \\ a \end{bmatrix}_k^{CA} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & \alpha \end{bmatrix} \times \begin{bmatrix} x \\ v_x \\ a \end{bmatrix}_{k-1}^{CA} + \begin{bmatrix} 0 \\ 0 \\ \sigma_a \sqrt{1-\alpha^2} \end{bmatrix}^{CA} \quad (16)$$

量测方程均为

$$x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x \\ v_x \\ a \end{bmatrix} \quad (17)$$

IMM-PF 和 MM-PF 算法需事先设定一个模型概率的马尔科夫矩阵和初始模型概率, 马尔科夫矩阵设为

$$P = \begin{bmatrix} 1 - \frac{T}{\tau_1} & \frac{T}{\tau_1} \\ \frac{T}{\tau_2} & 1 - \frac{T}{\tau_2} \end{bmatrix} \quad (18)$$

其中, σ_a 为加速度噪声标准方差, $R = \sigma_m^2$ 为量测噪声方差, $T=1$ 为仿真时间间隔, τ_1 和 τ_2 为事先人为设定的模式 1 和模式 2 相互转换参数, $\alpha \in (0, 1]$.

分别使用粒子数 $N=1000$ 和粒子数 $N=10000$ 对每一个场景进行仿真以观察粒子数目对算法性能的影响, N 为每个模型所用的粒子数, IMTM 中过渡粒子数设为 100. 进行 100 次蒙特卡洛仿真, 得到各场景的位置均方根误差 (Root mean square error, RMSE).

表 1 三种算法的各模块的时间复杂度

Table 1 The time complexity for the three methods

| 算法 | 输入交互阶段 | 采样阶段 | 重采样阶段 | 输出模块 | 模型概率更新阶段 | 整体复杂度 |
|--------|--------------|--------------|-----------|--------------|--------------|--------------|
| IMTM | - | $O(MNn_x^2)$ | $O(M^2N)$ | $O(Nn_x)$ | - | $O(MNn_x^2)$ |
| IMM-PF | $O(MNn_x^3)$ | $O(MNn_x^2)$ | $O(MN)$ | $O(MNn_x^2)$ | $O(MNn_x^3)$ | $O(MNn_x^3)$ |
| MM-PF | $O(MN)$ | $O(MNn_x^2)$ | $O(MN)$ | $O(MNn_x)$ | - | $O(MNn_x^2)$ |

表2 四种仿真场景参数
Table 2 The parameters of four simulations

| 场景 | α | σ_a (m/s ²) | σ_m (m) | τ_1 (s) | τ_2 (s) |
|----|----------|--------------------------------|----------------|--------------|--------------|
| 1 | 0.9 | 50 | 30 | 50 | 5 |
| 2 | 0.9 | 50 | 30 | 5 000 | 5 |
| 3 | 0.9 | 1 | 30 | 50 | 5 |
| 4 | 0.9 | 1 | 30 | 5 000 | 500 |

1) 场景1分析. 场景1中, 目标在40s~60s之间有5g的恒定加速度, 其余时间段为匀速运动, 场景1仿真结果如图3所示. 粒子数 $N = 10000$ 时, MMPF在两段匀速阶段性能最好, IMM-PF次之, 但在运动模型转变时, IMTM保持良好的模型过渡性能, 误差峰值(Peak RMSE)远远低于其余两种算法. 粒子数降低到 $N = 1000$ 时, IMM-PF算法与MM-PF算法在60s~100s阶段的性能相当, IMTM算法误差峰值大约为32m, 而MM-PF为42m, IMM-PF为51m.

2) 场景2分析. 场景2与场景1唯一的不同在于对于MM-PF和IMM-PF的模型转换概率 P 发生了变化, 其余与场景1保持相同, 在40s~60s具有一个5g的恒定加速度, 场景2仿真结果如图4所示. 粒子数 $N = 1000$ 和 $N = 10000$, 在40s发生运动模型转变时, MM-PF失锁, 整体性能IMM-PF最佳, 而IMTM模型过渡最平滑, 误差峰值远远低于IMM-PF.

3) 场景3分析. 目标在40s~1000s之间一直保持0.1g的加速度运动, 场景3仿真结果如图5所示. 当粒子数 $N = 10000$ 时, 算法IMTM性能最好, 误差峰值为25m, 而IMM-PF算法为32m, 当

粒子数降为1000时, 可以看出粒子数对MM-PF算法的影响很明显.

4) 场景4分析. 场景4的目标运动轨迹与场景3相同, 仅仅为模型转换矩阵 P 发生改变, 场景4仿真结果如图6所示. 当粒子数 $N = 10000$ 时, 算法MM-PF与IMTM性能相仿, IMM-PF在模型发生转变时误差较大. 粒子数 $N = 1000$ 时, MM-PF算法在50s后已发生发散现象, 粒子数对其影响较大.

从以上4个仿真场景可以看出, IMTM算法在所有场景中目标运动发生变化时估计性能最好, 其对外界环境的适应性强. 对于场景1和场景2, IMM-PF算法在目标匀速运动阶段估计性能较IMTM算法好, 在场景3和场景4中三种算法在匀速运动阶段估计性能相当. 可以看出, 模型转移矩阵 P 与目标运动模型的匹配程度对算法IMM-PF和MM-PF算法精度影响较大, 而算法IMTM无需事先设定矩阵 P , 避免了人为干扰的因素. 场景1和场景2中IMTM算法在匀速阶段精度较PF算法精度低的一个不可忽视的原因在于, 算法IMTM中主体算法的IMH算法精度本身较多模型算法中基本粒子滤波PF的精度低. 同时还可以看出, 算法MM-PF对粒子数敏感, 模型转移矩阵 P 不匹配时易发生不收敛现象. 其原因在于MM-PF算法虽然整体粒子数保持不变, 但其采用全局滤波使得每个模型的粒子数是时变的, 粒子数表征了模型的权值, 当运动轨迹不发生变化时, 其将大量的粒子用以描述正确模型, 此时其估计精度高, 如图4所示. 当目标运动轨迹发生变化时, 此时正确的模型的粒子数较少, MM-PF将使用较少的粒子来描述后验概率的分布, 而较少粒子将无法准确地描述后验概率的密度函数, 导致算法发散, 这与文献[6]的结论相同. 表3为各算法的每一个循环所用的平均时间.

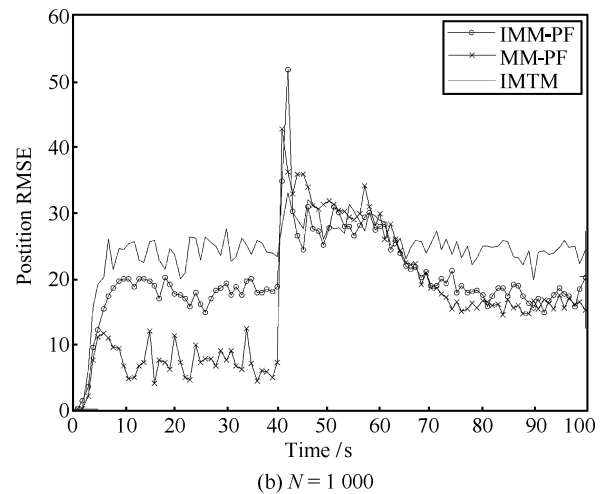
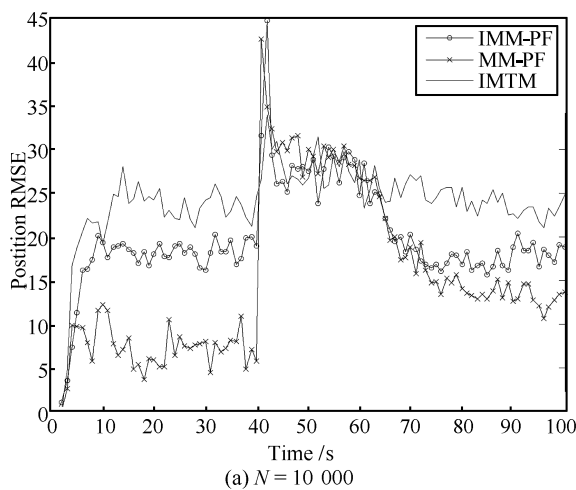


图3 场景1位置均方根误差

Fig. 3 RMSE of position estimate of Scene 1

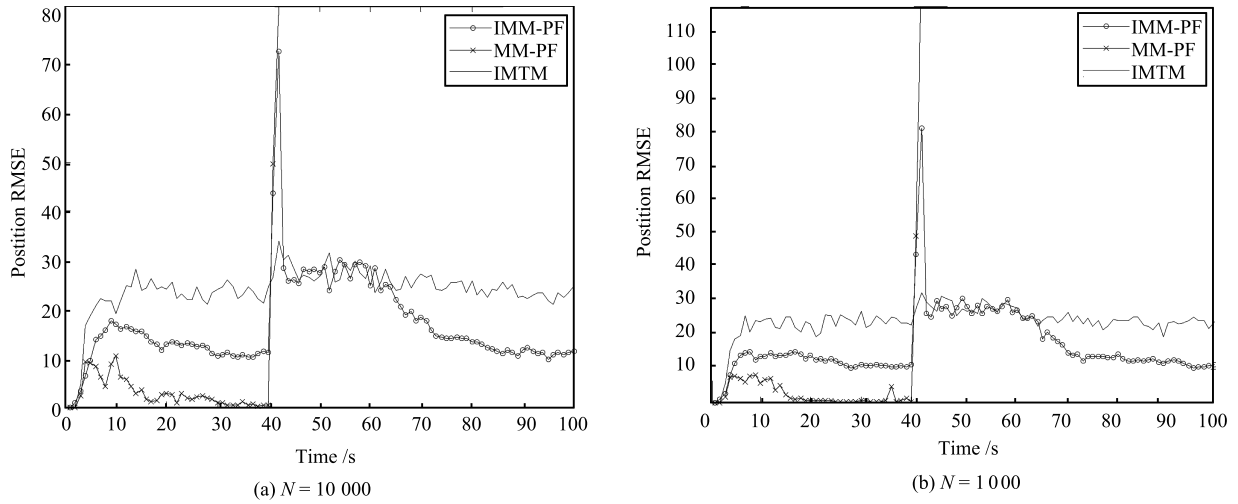


图 4 场景 2 位置均方根误差
Fig. 4 RMSE of position estimate of Scene 2

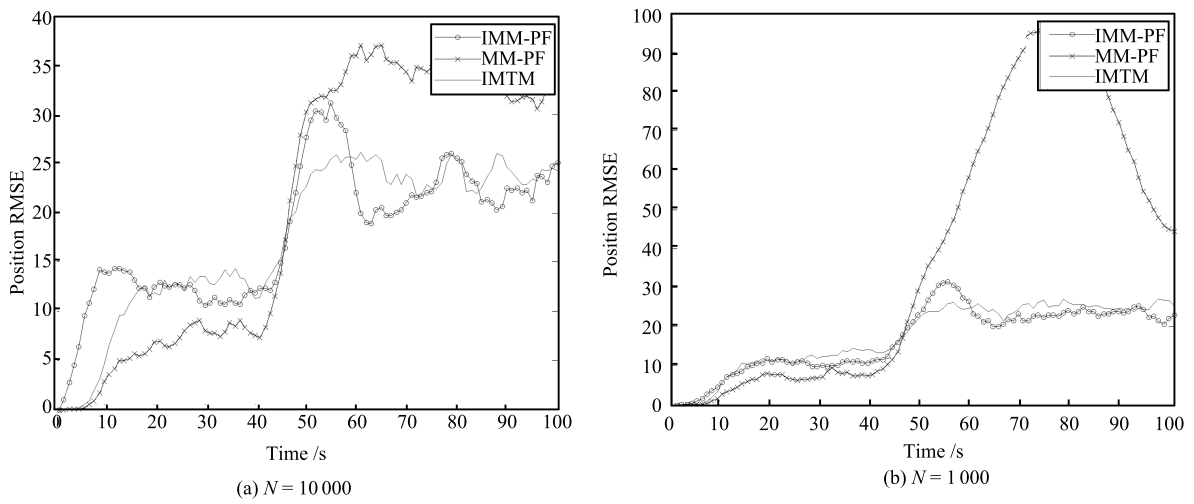


图 5 场景 3 位置均方根误差
Fig. 5 RMSE of position estimate of Scene 3

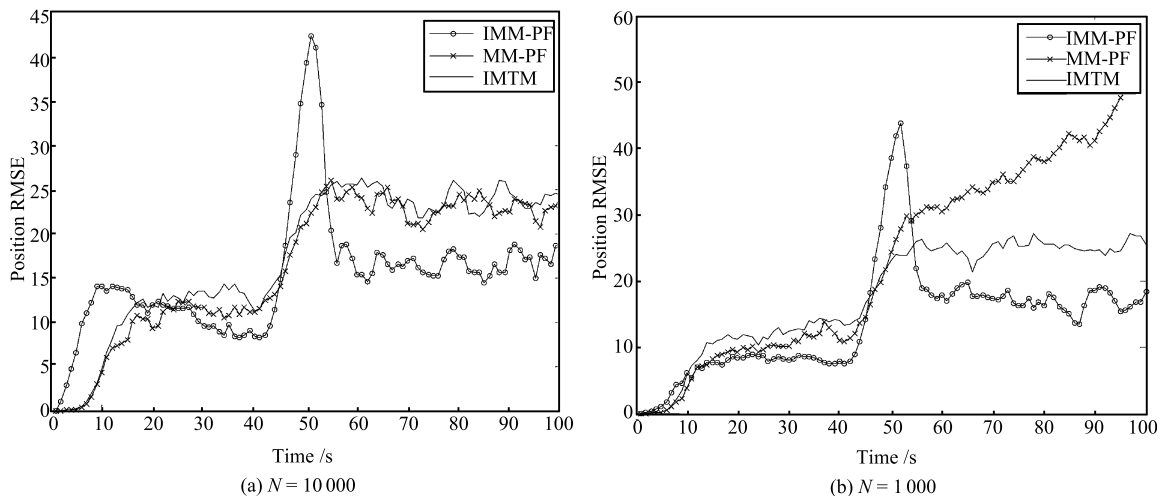


图 6 场景 4 位置均方根误差
Fig. 6 RMSE of position estimate of Scene 4

表3 各算法所需的平均时间 (s)
Table 3 Average time for estimate (s)

| N | IMM-PF | MM-PF | IMTM |
|------|--------|-------|------|
| 1000 | 1.2 | 2.1 | 0.20 |

从表3可以看出, IMTM 算法所需的时间最少, 将近为 MM-PF 算法的 1/10, IMTM 算法无粒子滤波中的并行运行瓶颈问题, MM-PF 算法的时间较长, 原因在于其为一个全局粒子滤波, 而 IMM-PF 算法为两个粒子滤波并行运算, 且需注意的, MM-PF 算法在粒子预测阶段的时间与这一时刻两个模型的权值比有关, 只有当两个模型权值相等时, MM-PF 中粒子预测阶段时间最小, 与 IMM-PF 和 IMTM 算法中粒子预测阶段所需时间相同, 当一个模型权值为 1 而另一个为 0 时, 粒子预测所需的时间为 IMM-PF 和 IMTM 的两倍, 即 MM-PF 算法不具有固定的算法延时时间。

第二种仿真中机动目标为不同角速度多次转弯的机动目标, 以验证算法对高机动目标的跟踪性能. 图7显示了机动目标的运动轨迹图, 仿真环境^[22-23]的具体参数如下: 仿真时长为 400 s, 初始状态为 $x_0 = 6 \text{ km}$, $v_{x0} = -172 \text{ m/s}$, $y_0 = 4 \text{ km}$, $v_{y0} = 246 \text{ m/s}$, 目标在 [56, 150] s, [182, 228] s, [268, 299] s 和 [328, 351] s 分别以向心加速度 $1g$, $-2g$, $3g$, $-4g$ 发生转弯运动, 雷达距离和角度量测噪声为 $\sigma_r^2 = (85 \text{ m})^2$ 和 $\sigma_\varphi^2 = (0.0175 \text{ rad})^2$.

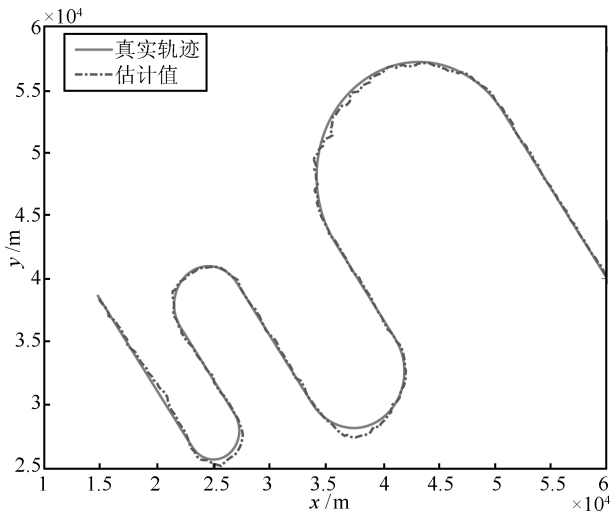


图7 目标运动轨迹图

Fig. 7 Trajectory of the maneuvering target

使用 3 种粒子滤波算法对上述机动目标进行跟踪. 在算法 IMM-PF、MMPF 和 IMTM 中, 均使用一个 CV (Constant velocity) 模型和两个 CT (Constant turn) 模型对目标进行估计. 设 $\mathbf{X}_k = (x_k, v_{xk},$

$y_k, v_{yk})^T$ 为系统状态变量, CV 模型的状态转移方程为

$$\mathbf{X}_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X}_{k-1} + \begin{bmatrix} \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & 0 & T \end{bmatrix} \times Q_{CV} \quad (19)$$

其中, $T = 1$ 代表时间间隔, Q_{CV} 为 CV 模型的过程噪声协方差. CT 模型的状态转移函数为

$$\mathbf{X}_k = \begin{bmatrix} 1 & \frac{\sin wT}{\cos wT} & 0 & -\frac{\cos T - 1}{w} \\ 0 & 1 & 0 & -\sin wT \\ 0 & \frac{1 - \cos wT}{w} & 1 & \frac{\sin wT}{w} \\ 0 & \sin wT & 0 & \cos wT \end{bmatrix} \mathbf{X}_{k-1} + \begin{bmatrix} \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & 0 & T \end{bmatrix} \times Q_{CT} \quad (20)$$

其中, w 为 CT 模型中的转弯角速度, 两个 CT 模型的角速度值为 $\pm 0.3 \text{ rad/s}$. Q_{CT1} 和 Q_{CT2} 为两个不同角速度 CT 模型的过程噪声协方差. 本文设置 3 个模型的协方差均为

$$Q_{CT1} = Q_{CT2} = Q_{CV} = \begin{bmatrix} 100 & & & \\ & 10 & & \\ & & 100 & \\ & & & 10 \end{bmatrix} \quad (21)$$

设定一个模型概率的马尔科夫矩阵和初始模型概率, 马尔科夫矩阵均设为

$$P = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix} \quad (22)$$

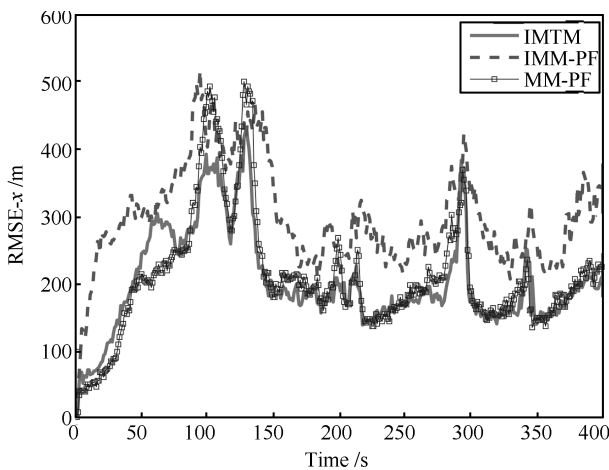
量测方程如下:

$$\mathbf{Z}(k) = h(\mathbf{X}_k) + v_k = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \arctan(\frac{y_k}{x_k}) \end{bmatrix} + \begin{bmatrix} v_k^r \\ v_k^\varphi \end{bmatrix} \quad (23)$$

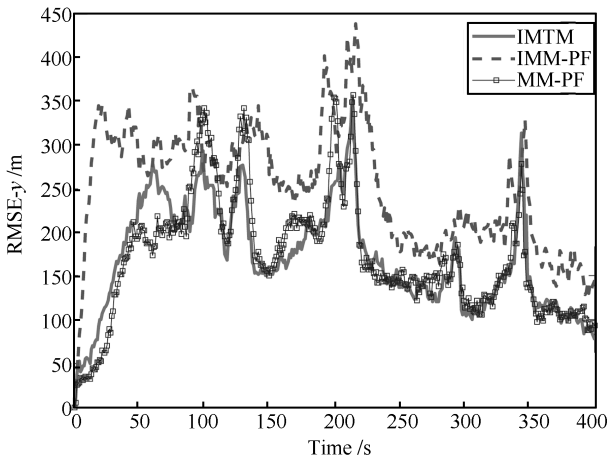
其中, v_k^r 和 v_k^φ 为距离观测噪声和角度观测噪声, 均为零均值的高斯白噪声, 协方差分别为 σ_r^2 和 σ_φ^2 .

每个模型所用的粒子数 $N = 1000$, IMTM 算法 $N_b = 100$, 进行 100 次蒙特卡洛仿真, 利用均方根误差 (RMSE) 对算法进行比较, 图 8 为位置均方根误差, 图 9 为速度均方根误差, 其中 RMSE 的定义如下:

$$RMSE_k = \left(\frac{1}{MC} \sum_{l=1}^{MC} (\hat{x}_{k,l} - x_k)^2 \right)^{\frac{1}{2}} \quad (24)$$

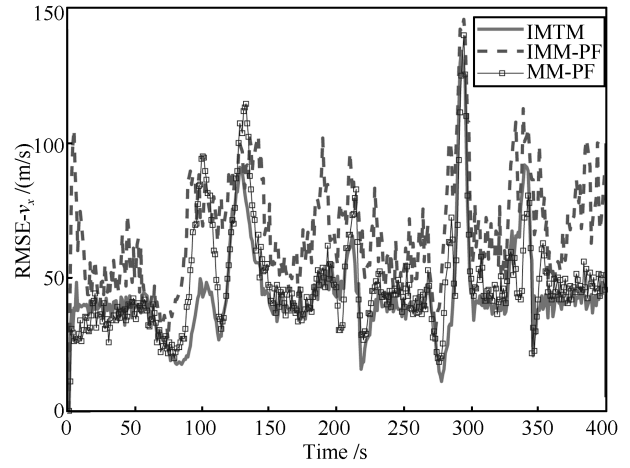


(a) X 轴方向
(a) X coordinate

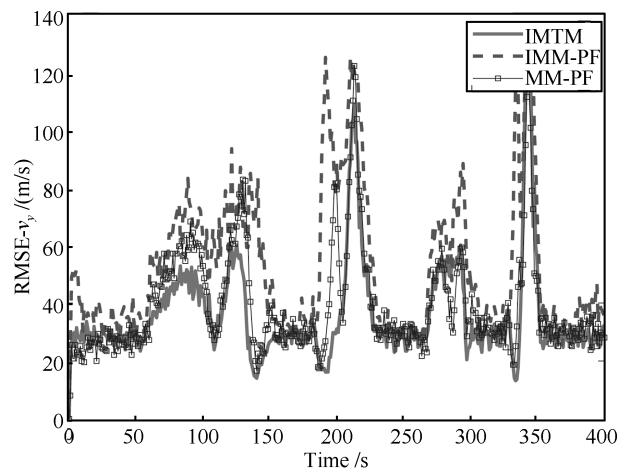


(b) Y 轴方向
(b) Y coordinate

图 8 位置估计误差均方误差
Fig. 8 RMSE of position estimate



(a) X 轴方向
(a) X coordinate



(b) Y 轴方向
(b) Y coordinate

图 9 速度估计误差均方误差

Fig. 9 RMSE of velocity estimate

其中, $MC = 100$ 为蒙特卡洛次数, $\hat{x}_{k,l}$ 为第 l 次蒙特卡洛仿真中 k 时刻的状态的估计值, x_k 为 k 时刻的真实值.

从图 8 和图 9 可以看出, 三种算法在目标发生机动时速度和位置估计误差均增大, 同时还可以看出, IMM-PF 的估计精度最低, 本文提出的算法 IMTM 在高机动目标跟踪中估计精度较好, 与算法 MM-PF 相当. 比较峰值误差发现, IMTM 的峰值误差最小, 而 IMM-PF 和 MM-PF 峰值误差大小相仿, 在第一次和第二次发生机动目标转弯时, MM-PF 的峰值误差大于 IMM-PF, 其余时刻均为 IMM-PF 最大, 但需要注意的是, MM-PF 算法在模型转移矩阵 P 严重不符合运动模型时可能出现失锁现象. 仿真结果表明, 本文算法 IMTM 在高机动目标跟踪中具有较好的适用性. 表 4 为各算法运行时间.

表 4 算法所需的平均时间 (s)
Table 4 Average time for estimate (s)

| N | IMM-PF | MM-PF | IMTM |
|-------|--------|-------|------|
| 1 000 | 1.7 | 4.4 | 0.45 |

下面通过不同粒子数 N 下的 IMTM 算法的位置估计误差 RMSE 分析 IMTM 算法与粒子数目 N 的关系, 其中 RMSE 的定义如下:

$$RMSE_N = \left(\frac{1}{K} \sum_{k=1}^K \frac{1}{MC} \sum_{l=1}^{MC} (\hat{x}_{N,k,l} - x_k)^2 \right)^{\frac{1}{2}} \quad (25)$$

其中, 模型粒子数 N 从 100 到 3000, 间隔为 100, $K = 400$ 为仿真总时长, $\hat{x}_{N,k,l}$ 为模型粒子数为 N 时第 l 次蒙特卡洛仿真中 k 时刻的状态的估计值, 其余各参数的定义与式 (24) 相同. 仿真结果如图 10 所示, 其中, $RMSE_x$ 代表 x 轴方向的位置估计均方根误差, $RMSE_y$ 代表 y 轴方向的位置估计均方根误差.

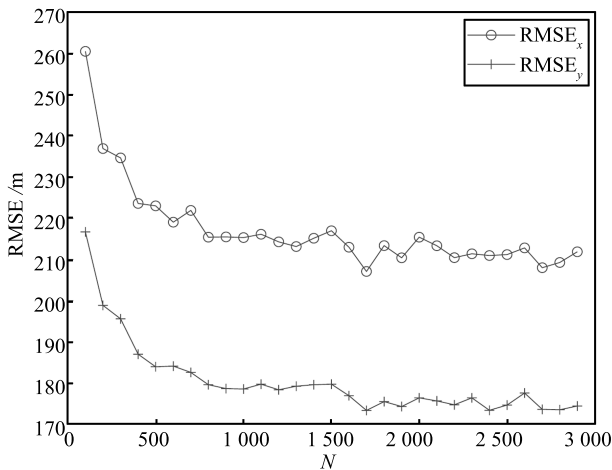


图 10 IMTM 算法与粒子数 N 的关系

Fig. 10 The relationship between IMTM method and the particle numbers N

从图 10 可以看出, 随着粒子数 N 的增加, RMSE 降低, 算法的估计性能提高, 当粒子数 $N > 1500$ 时, RMSE 趋于稳定. 这与第一种机动目标仿真所得的结果相同, 即 IMTM 算法对粒子数 N 不敏感, 表明 IMTM 算法能够利用较少的粒子描述目标分布, 其原因在于 IMTM 算法能够快速而准确地实现模型之间的交互. 对粒子数不敏感的特性将使得 IMTM 算法在相同总粒子数的情况下, 较 MM-PF 和 IMM-PF 容纳更多的模型数以提高目标估计精度.

4 结论

本文提出了一种多点测试多模型粒子滤波算法 (IMTM). IMTM 算法在目标发生机动时能够有效地跟随运动模型的变化, 峰值误差较传统的多模型粒子滤波算法得到了较大的降低, 整体跟踪精度较传统多模型粒子滤波算法得到一定的提高, 在运算时间方面较传统的算法得到较大的提高, 实时性较好, 算法复杂度得到降低.

References

- Blom H A P, Bar-shalom Y. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 1988, **33**(8): 780–783
- Li X R, Jilkov V P. Survey of maneuvering target tracking, Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 2005, **41**(4): 1255–1321
- Shi Yong, Han Chong-Zhao. Adaptive UKF method with applications to target tracking. *Acta Automatica Sinica*, 2011, **37**(6): 755–759
(石勇, 韩崇昭. 自适应 UKF 算法在目标跟踪中的应用. *自动化学报*, 2011, **37**(6): 755–759)
- Doucet A, Johansen A M. A tutorial on particle filtering and smoothing: fifteen years later. *Handbook of Nonlinear Filtering*, 2009, **12**: 656–704
- Foo P H, Ng G W. Combining the interacting multiple model method with particle filters for manoeuvring target tracking. *IET Radar, Sonar and Navigation*, 2011, **5**(3): 234–255
- Gordon N J, Maskell S, Kirubarajan T. Efficient particle filters for joint tracking and classification. In: *Proceedings of the 2002 SPIE 4728: Signal and Data Processing of Small Targets*. Orlando, FL, USA: SPIE, 2002. 439–449
- Hong S H, Shi Z G, Chen K S. Easy-hardware-implementation MMPF for maneuvering target tracking: algorithm and architecture. *Journal of Signal Processing Systems*, 2010, **61**(3): 259–269
- McGinnity S, Irwin G W. Multiple model bootstrap filter for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 2000, **36**(3): 1006–1012
- Yang N, Tian W F, Jin Z H. An interacting multiple model particle filter for manoeuvring target location. *Measurement Science and Technology*, 2006, **17**(6): 1307–1311
- Boers Y, Driessen J N. Interacting multiple model particle filter. *IEE Proceedings — Radar, Sonar, and Navigation*, 2003, **150**(5): 344–349
- Sankaranarayanan A C, Chellappa R, Srivastava A. Algorithmic and architectural design methodology for particle filters in hardware. In: *Proceedings of the 2005 IEEE International Conference on Computer Design: VLSI in Computers and Processors*. San Jose, CA, USA: IEEE, 2005. 275–280

- 12 Miao L F, Zhang J J, Chakrabarti C, Papandreou-Suppappola A. A new parallel implementation for particle filters and its application to adaptive waveform design. In: Proceedings of the 2010 IEEE Workshop on Signal Processing Systems (SIPS). San Francisco, CA: IEEE, 2010. **978**(1): 19–24
- 13 Liu J S. *Monte Carlo Strategies in Scientific Computing*. New York: Springer Science and Business Media, 2004. 111–123
- 14 Liu J S, Liang F M, Wong W H. The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 2000, **95**(449): 121–134
- 15 Qin Z S, Liu J S. Multi-point Metropolis method with application to hybrid Monte Carlo. *Journal of Computational Physics*, 2001, **172**(2): 827–840
- 16 Storvik G. On the flexibility of Metropolis-Hastings acceptance probabilities in auxiliary variable proposal generation. *Scandinavian Journal of Statistics*, 2011, **38**(2): 342–358
- 17 Pandolfi S, Bartolucci F, Friel N. A generalization of the multiple-try Metropolis algorithm for Bayesian estimation and model selection. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics. 2010. 581–588
- 18 Casarin R, Craiu R, Leisen F. Interacting multiple try algorithms with different proposal distributions. *Statistics and Computing*, 2013, **23**(2): 185–200
- 19 Blom H A P, Bloem E A. Exact Bayesian and particle filtering of stochastic hybrid systems. *IEEE Transactions on Aerospace and Electronic Systems*, 2007, **43**(1): 55–70
- 20 Sankaranarayanan A C, Srivastava A, Chellappa R. Algorithmic and architectural optimizations for computationally efficient particle filtering. *IEEE Transactions on Image Processing*, 2008, **17**(5): 737–748
- 21 Arulampalam M S, Maskell S, Gordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 2002, **50**(2): 174–188
- 22 Linder S P, Schell C. A non-Bayesian segmenting tracker for highly maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 2005, **41**(4): 1168–1177
- 23 Wang Xiao, Han Chong-Zhao. A multiple model particle filter for maneuvering target tracking based on composite sampling. *Acta Automatica Sinica*, 2013, **39**(7): 1152–1156 (王晓, 韩崇昭. 基于混合采样的多模型机动目标跟踪算法. *自动化学报*, 2013, **39**(7): 1152–1156)



王 伟 哈尔滨工程大学自动化学院教授. 2006 年获得哈尔滨工程大学自动化学院博士学位. 主要研究方向为高精度惯性器件与捷联惯性导航, 先进控制理论及应用. 本文通信作者.

E-mail: wangwei407@hrbeu.edu.cn

(WANG Wei Professor at the College of Automation, Harbin Engineering University. He received his Ph.D. degree from Harbin Engineering University in 2006. His research interest covers high-precision inertial device and strapdown inertial navigation, advanced control theory and its application. Corresponding author of this paper.)



余玉揆 哈尔滨工程大学自动化学院博士研究生. 2011 年获得合肥工业大学自动化学院学士学位. 主要研究方向为非线性滤波, 光纤非线性, 光纤通信.

E-mail: yykmaidou@gmail.com

(YU Yu-Kui Ph.D. candidate at the College of Automation, Harbin Engineering University. He received his bachelor degree from Hefei University of Technology in 2011. His research interest covers nonlinear filtering, non-linearity of fiber, and optical communication.)