

基于 Stiefel 流形的粒子滤波器研究^{*}

朱志宇[†] 杨官校

(江苏科技大学电子信息学院, 镇江 212003)

(2009 年 8 月 16 日收到; 2010 年 7 月 5 日收到修改稿)

为了解决粒子滤波的粒子退化和粒子多样性丧失问题, 提出了一种基于 Stiefel 流形的粒子滤波算法. 该算法将系统模型置于 Stiefel 流形上, 用朗之万分布描述过程转移概率分布, 用矩阵正态分布表示似然函数分布, 在流形分布上进行粒子采样. 在计算加权粒子的均值时, 将流形嵌入到欧氏空间中, 先计算欧氏空间中的粒子均值, 再将计算结果投影到嵌套流形上, 这就排除了噪声统计特性对粒子权重方差的影响, 得到了一种受系统状态模型限制较少的重要性概率密度函数通用选择方案. 仿真时选取单变量非静态增长模型, 仿真结果验证了该算法的实时性、鲁棒性, 滤波精度和滤波效率均比无味粒子滤波算法更好.

关键词: 粒子滤波, Stiefel 流形, 单变量非静态增长模型

PACC: 0250, 0260, 0547, 0290

1. 引言

粒子滤波是一种基于蒙特卡罗方法^[1,2]和 Bayes 估计的滤波方法, 在处理非高斯非线性系统的参数估计和状态滤波问题方面具有独特的优势, 已被广泛应用于导航、计算机视觉、信号处理、目标跟踪、故障诊断等领域^[3-8]. 该算法采用样本形式而不是函数形式描述先验信息和后验信息, 能解决扩展卡尔曼滤波(EKF)的非线性误差积累问题, 精度逼近最优.

粒子滤波一般选取重要性概率密度作为先验概率密度. 这种方法丢失了当前时刻的量测, 如果模型不准确或者量测噪声突然增大, 从重要性概率密度采样得到的样本与从真实后验概率密度采样得到的样本之间会有很大偏差. 解决办法是设法将粒子向似然函数的峰值区移动, 如“预编辑(prior editing)”及类似的“取舍法(accept reject procedure)”采样; 或者选用更合适的建议分布, 如采用先验转移密度的退火形式作为建议分布^[9]、用似然函数作为建议分布^[10]等; 此外, 还有采用 EKF、无味卡尔曼滤波^[11]或高斯-厄米特滤波将最新量测计入建议分布的 EKF^[12]、无味粒子滤波(UPF)^[13]和高斯-厄米特粒子滤波^[14]等等.

为了获得粒子滤波重要性概率密度函数的通用选择方案, 使得重要性概率密度函数的选择不会较多地依赖系统的状态模型, 并由此解决粒子退化和粒子多样性丧失问题, 本文提出了一种基于 Stiefel 流形的粒子滤波(SMPF)算法, 将状态空间模型和量测模型置于流形上, 用粒子表示流形分布, 通过在流形上采样实现状态的预测和样本集、权值的更新.

2. 粒子滤波

粒子滤波通过非参数化的蒙特卡罗模拟方法实现递推 Bayes 滤波, 适用于非线性目标运动模型、非线性传感器量测模型和非高斯噪声的状态估计, 它通过一组加权粒子的演化与传播来近似状态后验概率密度函数. 当粒子数足够多时, 这组加权粒子能完全描述后验概率分布, 此时粒子滤波就是最优的 Bayes 估计.

假设动态系统的状态空间模型如下:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k), \quad (2)$$

其中 \mathbf{x}_k 表示系统在 k 时刻的状态, \mathbf{z}_k 表示 k 时刻的量测向量, $\mathbf{f}_k: \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ 和 $\mathbf{h}_k: \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow$

^{*} 江苏省高等学校自然科学基金(批准号: 06KJB510030)和国家自然科学基金(批准号: 61075028)资助的课题.

[†] E-mail: zzydzz@163.com

\mathfrak{R}^{n_z} 分别是系统的状态转移函数和量测函数, \mathbf{v}_k 和 \mathbf{u}_k 分别表示系统的过程噪声和量测噪声.

粒子滤波通过两个步骤计算 k 时刻的后验概率密度 $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$, 即预测和更新. 用 $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^N$ 表示系统后验概率密度函数 $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ 的粒子集合, 其中 $\{\mathbf{x}_{0:k}^i\}_{i=1}^N$ 是支持样本集, 相应的权值为 $\{w_k^i\}_{i=1}^N$, 且满足 $\sum_{i=1}^N w_k^i = 1$, 而 $\mathbf{x}_{0:k} = \{\mathbf{x}_j, j = 0, \dots, k\}$ 表示到时刻 k 系统所有状态的集合. 所以, k 时刻的后验密度可以近似为

$$p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i), \quad (3)$$

其中

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}. \quad (4)$$

这里 $q(\cdot)$ 为重要性概率密度函数, 通常选取 $q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$ 作为重要性概率函数的先验分布, 将其代入(4)式可得

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i). \quad (5)$$

这种选取方法在量测精度要求低的场合能够获得较好的滤波效果. 但是该方法没有考虑当前时刻的量测, 因此从重要性概率密度采样得到的样本与从真实后验概率密度采样得到的样本之间存在着很大偏差, 特别是当似然函数位于系统状态转移概率密度的尾部或似然函数呈尖峰状态时, 这种偏差就更加明显. 似然函数与先验分布之间的关系如图 1 所示.

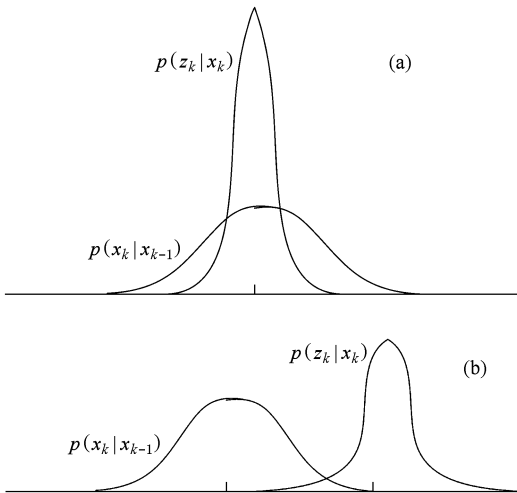


图1 似然函数与先验分布的关系 (a) 似然函数呈尖峰状态, (b) 似然函数位于先验分布的尾部

3. Stiefel 流形

Stiefel 流形 $\mathbf{V}_{n,m}$ 由 $n \times m$ 维每列都是正交向量的矩阵组成, 它是 m 维欧氏空间 R^m 中的正交 n 标架集, 其数学描述如下:

$$\mathbf{V}_{n,m} = \{\mathbf{X} \in R^{m \times n}; \mathbf{X}^T \mathbf{X} = \mathbf{I}_n\}. \quad (6)$$

这里 \mathbf{I}_n 表示 $n \times n$ 维单位矩阵, 而 $\mathbf{V}_{1,m}$ 是 R^m 中的单位球面, $\mathbf{V}_{n,m}$ 是正交群 $O(m)$.

为了解 Stiefel 流形上的概率分布, 首先引入不变测度的概念.

定义 1 \mathbf{M} 是一黎曼流形, \mathbf{G} 是作用在 \mathbf{M} 上的可测子集, 如果对于每一个可测量 $\mathbf{S} \subset \mathbf{M}$, 有

$$\begin{aligned} \mu(\mathbf{S}) &= \mu(\mathbf{T}\mathbf{S}) \\ &= \mu(\mathbf{S}\mathbf{T}) \quad (\mathbf{T} \in \mathbf{G}), \end{aligned}$$

则 μ 就是关于可测子集 \mathbf{G} 的不变测度. 若 $\mu(\mathbf{M}) = 1$, 则 μ 表示单位不变测度.

定义矩阵 $\mathbf{X} \in R^{m \times n}$, 其点集 $\mathbf{X} \in R^{mn}$, 则 Stiefel 流形上的概率分布的数学表达为 $\int_{\mathbf{S}} p(\mathbf{S}) [\mathrm{d}\mathbf{X}]$, 其中 $\mathbf{S} \subset \mathbf{V}_{n,m}$, $p(\cdot)$ 为关于不变测度微分形式 $[\mathrm{d}\mathbf{X}]$ 的矩阵变量密度函数.

4. SMPF 算法

Stiefel 流形上的概率分布是关于不变测度微分形式的多重积分, 相当于欧氏空间的勒贝格测度. 本文提出的 SMPF 算法将状态空间模型和量测模型置于流形上, 在流形分布上进行粒子采样. 用朗之万分布描述过程转移概率分布, 用矩阵正态分布表示似然函数分布, 这两个分布的优势是易于实现近点选取和样本采样, 且计算简单. SMPF 算法在计算加权样本的均值时, 将流形嵌入到欧氏空间中, 先计算欧氏空间中的样本均值, 再将计算结果投影到嵌套流形上, 从而排除了噪声统计特性对粒子权重方差的影响, 在选择非线性模型的重要性概率密度函数时受系统状态模型的限制较少.

4.1. 矩阵变量分布

矩阵变量概率分布对应流形上的不变测量.

$m \times n$ 维随机矩阵正态分布如下:

$$\begin{aligned} N(\mathbf{X}; \mathbf{M}, \boldsymbol{\Omega}, \boldsymbol{\Sigma}) \mathrm{d}\mathbf{X} &= [(2\pi)^{mn} |\boldsymbol{\Omega}|^n |\boldsymbol{\Sigma}|^m]^{-1/2} \\ &\times \exp \left(-\frac{1}{2} \mathrm{tr} [\boldsymbol{\Omega}^{-1} (\mathbf{X} - \mathbf{M})^T \right. \end{aligned}$$

$$\times \Sigma^{-1}(\mathbf{X} - \mathbf{M})] \mathrm{d}\mathbf{X}, \quad (7)$$

其中 \mathbf{M} 是矩阵 \mathbf{X} 的均值, $\mathbf{\Omega}$ 是协方差矩阵, Σ 是 $R^{m \times n}$ 上的概率密度函数, $\mathrm{d}\mathbf{X}$ 表示勒贝格测度, $\mathrm{tr}(\cdot)$ 表示矩阵的迹.

$m \times n$ 维随机矩阵朗之万分布^[15]如下:

$$D_{\mathrm{vMF}}(\mathbf{X}; \mathbf{F})[\mathrm{d}\mathbf{X}] = C \exp(\mathrm{tr}(\mathbf{X}^T \mathbf{F}))[\mathrm{d}\mathbf{X}], \quad (8)$$

其中 C 是标准化常量. 朗之万分布又称为 von Mises-Fisher 分布, 因此(8)式中用 $D_{\mathrm{vMF}}(\cdot)$ 表示矩阵朗之万分布. $m \times n$ 维随机矩阵朗之万分布和一般欧氏空间 $R^{m \times n}$ 上的分布是紧密相关的. 对于随机矩阵 \mathbf{X} , 令参数 $\Sigma = \mathbf{I}$, 则有常数 $\mathbf{X}^T \mathbf{X} = \mathbf{I}$, 这时所得到的 $R^{m \times n}$ 空间上的概率密度就是一个朗之万分布.

4.2. 流形上的状态空间模型^[16]

状态方程为

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}).$$

由 $\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \rightarrow p(\mathbf{x}_k | \mathbf{x}_{k-1})$ 得到过程转移概率为

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}) = D_{\mathrm{vMF}}(\mathbf{x}_k; \mathbf{x}_{k-1}^T \boldsymbol{\phi}) = C \exp(\mathrm{tr}(\mathbf{x}_k^T \mathbf{x}_{k-1} \boldsymbol{\phi})). \quad (9)$$

取朗之万分布作为过程转移概率分布.

量测方程为

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k).$$

由 $\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{u}_k) \rightarrow p(\mathbf{z}_k | \mathbf{x}_k)$ 得到似然函数为

$$\begin{aligned} P(\mathbf{z}_k | \mathbf{x}_k) &= N(\mathbf{z}_k; \mathbf{x}_k, \sigma^2 \mathbf{I}, \mathbf{I}) \\ &= [(2\pi)^n \sigma^2]^{-m/2} \exp\left(-\frac{1}{2\sigma^2} \right. \\ &\quad \left. \times \mathrm{tr}[(\mathbf{z}_k - \mathbf{x}_k)^T (\mathbf{z}_k - \mathbf{x}_k)]\right). \end{aligned} \quad (10)$$

取正态分布函数作为似然函数概率分布.

4.3. SMPF 算法描述

SMPF 算法共分 7 步进行.

第一步获得量测值 $\mathbf{z}_k, \mathbf{z}_k$ 服从 $N(\mathbf{z}_k; \mathbf{x}_k, \mathbf{\Omega}, \Sigma)$ 分布.

$$\begin{aligned} N(\mathbf{z}_k; \mathbf{x}_k, \mathbf{\Omega}, \Sigma) &= N(\mathbf{z}_k; \mathbf{x}_k, \sigma^2 \mathbf{I}, \mathbf{I}) \\ &= [(2\pi)^n \sigma^2]^{-m/2} \exp\left(-\frac{1}{2\sigma^2} \right. \\ &\quad \left. \times \mathrm{tr}[(\mathbf{z}_k - \mathbf{x}_k)^T (\mathbf{z}_k - \mathbf{x}_k)]\right), \end{aligned}$$

其中 \mathbf{I} 表示单位矩阵向量.

第二步初始化, 在 $k=0$ 时刻, 从重要性概率密度函数采样. 取先验分布作为重要性概率密度函数

$$\begin{aligned} q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k) &= p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i) \\ &= D_{\mathrm{vMF}}(\mathbf{x}_k; \mathbf{x}_{k-1}^T \boldsymbol{\phi}) \\ &= C \exp(\mathrm{tr}(\mathbf{x}_k^T \mathbf{x}_{k-1} \boldsymbol{\phi})). \end{aligned}$$

这里的 \mathbf{x}_k^i 服从朗之万分布.

第三步计算重要性权值, 即

$$\begin{aligned} w_k^i &= w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) \\ &= w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} \\ &= w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i) \\ &= w_{k-1}^i N(\mathbf{z}_k; \mathbf{x}_k, \mathbf{\Omega}, \Sigma) \\ &= w_{k-1}^i N(\mathbf{z}_k; \mathbf{x}_k, \sigma^2 \mathbf{I}, \mathbf{I}) \\ &= w_{k-1}^i [(2\pi)^k \sigma^2]^{-m/2} \exp\left(-\frac{1}{2\sigma^2} \mathrm{tr}[(\mathbf{z}_k - \mathbf{x}_k^i)^T \right. \\ &\quad \left. \times (\mathbf{z}_k - \mathbf{x}_k^i)]\right). \end{aligned}$$

第四步计算归一化重要性权值

$$\sum_{i=1}^N w_k^i = 1.$$

第五步判断是否有必要进行重采样.

第六步输出状态估计和方差估计. 状态估计为

$$\hat{\mathbf{x}}_k = \sum_{i=1}^N w_k^i \mathbf{x}_k^i.$$

方差估计为

$$\mathbf{P} = \sum_{i=1}^N w_k^i (\mathbf{x}_k^i - \hat{\mathbf{x}}_k)(\mathbf{x}_k^i - \hat{\mathbf{x}}_k)^T.$$

第七步判断是否结束, 若是则退出本算法, 否则返回第二步.

5. 算法仿真分析

选用单变量非静态增长模型^[17]作为仿真模型, 验证算法的性能.

下面给出仿真对象的过程模型和量测模型. 过程模型为

$$\begin{aligned} x(t) &= 0.5x(t-1) + \frac{25x(t-1)}{1 + [x(t-1)]^2} \\ &\quad + 8\cos(1.2(t-1)) + w(t). \end{aligned}$$

量测模型为

$$z(t) = \frac{x(t)^2}{20} + v(t).$$

这里的 $w(t)$ 和 $v(t)$ 均为零均值高斯噪声. 该系统具有高度非线性, 似然函数呈双峰, 很难用传统方法解决其滤波问题. 本文分别采用粒子数 $N=1000$

的 UPF 算法^[9]和 $N=400$ 的 SMPF 算法对该系统进行状态估计和跟踪,当过程噪声方差 $Q = 10$, 量测噪声方差 $P = 1$ 时,两种算法的仿真结果如图 2 和图 3 所示,图中的横坐标 t 为迭代次数,纵坐标 x 为状态值.

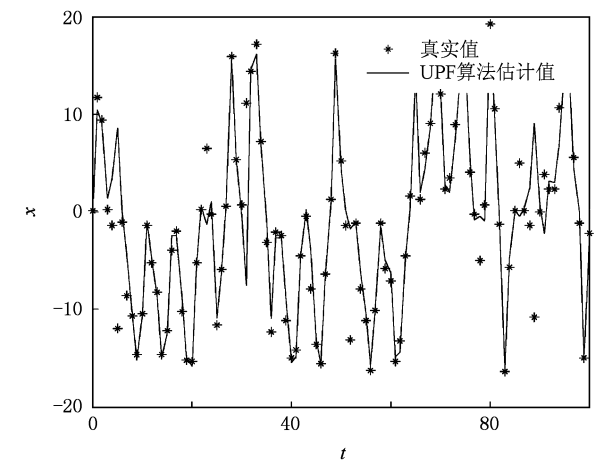


图 2 UPF 算法的估计结果 $N=1000$

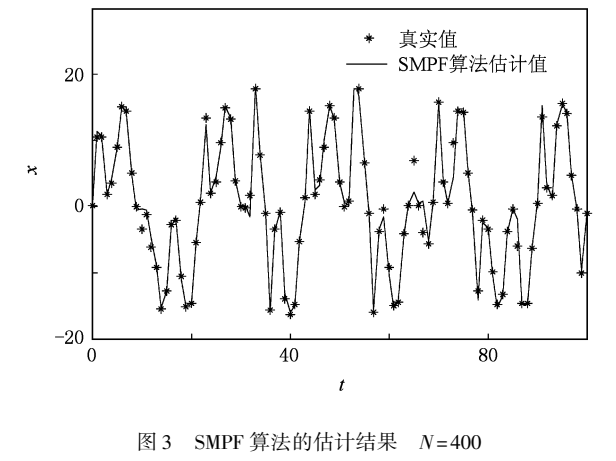


图 3 SMPF 算法的估计结果 $N=400$

从图 2 和图 3 中的仿真结果可以直观地看出, SMPF 算法只需采用较少的粒子就可以获得更好的状态估计结果. 为了进一步比较两种算法的估计精度,图 4 给出了 UPF 算法和 SMPF 算法的均方根误差平均值(ARMSE)曲线,图中的纵坐标 E_{ARMSE} 表示 ARMSE.

图 4 中的结果表明, SMPF 算法具有比 UPF 算法更高的估计精度.

当过程噪声方差 $Q = 10$, 量测噪声方差 $P = 1$ 时,分别选用不同的粒子数,进一步考察上述因素对不同滤波器滤波性能的影响,得到的仿真结果如表 1 所列.

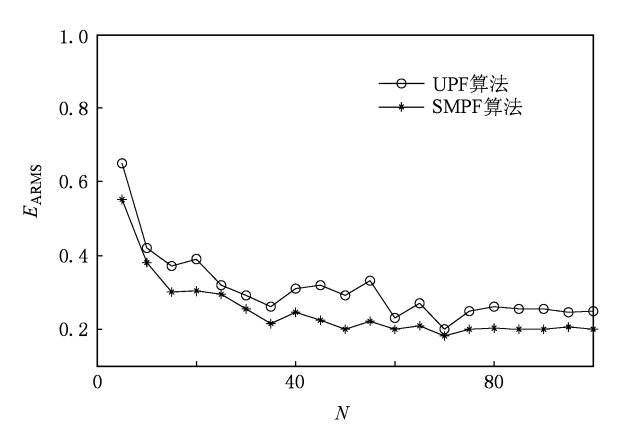


图 4 UPF 算法和 SMPF 算法的 ARMSE 曲线

表 1 不同粒子数下的仿真结果比较

算法	N	$\eta/\%$	E_{ARMSE}	估计时间/s
UPF	100	4.228	0.2365	0.215
	500	1.768	0.1131	0.456
	1000	2.057	0.0486	1.232
SMPF	100	6.891	0.1451	0.188
	300	5.012	0.0665	0.221
	400	11.161	0.0224	0.312

表 1 中的滤波效率 η 可表示为

$$\eta = \frac{1}{E_{\text{ARMSE}} N}.$$

显然,滤波效率越高,表明仅需较少的粒子数 N 就能获得更高的滤波精度(较小的 ARMSE),它能从一个侧面反映出粒子滤波的综合性能.

从表 1 可知,当取相同的粒子数 $N = 100$ 时, SMPF 算法比 UPF 算法具有更高的估计精度和滤波效率、更短的状态估计时间. 当 UPF 算法的粒子数 N 增加至 1000 时, $E_{\text{ARMSE}} = 0.0486$, 有了较大提高,但是滤波效率 $\eta = 2.057\%$, 反而有所下降,同时所需时间比 $N = 100$ 时增加了近 6 倍,说明滤波精度的获得是以牺牲实时性和滤波效率为代价的. 而当 SMPF 算法的粒子数 $N = 400$ 时,滤波效率 η 就达到了 11.161%, $E_{\text{ARMSE}} = 0.0224$, 比 UPF 算法 ($N = 1000$) 的指标均提高了数倍. 此时 SMPF 算法的状态估计时间为 0.312 s, 只是 UPF 算法 ($N = 1000$) 的 1/4 左右. 这说明 SMPF 算法能以较少的粒子获得更高的滤波精度、更好的滤波效率,且实时性也得到了较大改善. UPF 算法 ($N = 1000$) 和 SMPF 算法 ($N = 400$) 的状态估计曲线如图 3 和图 4 所示.

为了进一步表明 SMPF 算法在鲁棒性和实时性

方面的优势,分别选用不同的过程噪声方差 Q 和量测噪声方差 P , 采用 $N=300$ 的 UPF 算法和 SMPF 算法对该系统进行状态估计和跟踪,得到的仿真结果如表 2 所列.

表 2 不同噪声下的仿真结果比较

噪声方差	算法	$\eta/\%$	E_{ARMS}	估计时间/s
$Q = 10$	UPF	2.131	0.1564	0.406
$P = 1$	SMPF	5.012	0.0665	0.221
$Q = 20$	UPF	0.0913	0.3652	0.563
$P = 8$	SMPF	4.796	0.0695	0.263
$Q = 30$	UPF	0.0437	0.7512	1.188
$P = 15$	SMPF	4.474	0.0745	0.321

从表 2 可知,当过程噪声和量测噪声逐步增加时,和 UPF 算法相比, SMPF 算法的滤波效率、ARMSE、状态估计时间等指标均较为平稳,而 UPF 算法各项指标的降幅较大. 这说明本文提出的

SMPF 算法的滤波性能受噪声的影响较小,具有较好的鲁棒性和实时性.

6. 结 论

针对粒子滤波及目前常用的改进算法(如 UPF 等)在处理非线性非高斯问题上所存在的滤波效率低、滤波鲁棒性差等一系列缺陷,本文提出了 SMPF 算法. SMPF 算法将非线性系统置于流形上,用朗之万分布描述过程转移概率分布,用矩阵正态分布表示似然函数分布,以此来实现系统状态预测和样本集、权值的更新. 仿真结果表明, SMPF 算法具有很好的滤波性能,其实时性、鲁棒性、滤波精度和滤波效率均比 UPF 算法好,这表明选择 Stiefel 流形作为粒子滤波器的重要性概率密度函数,其滤波效果受噪声、状态模型的影响较小,是一种较为通用的重要性概率密度函数选择方案.

[1] Cao W F, Ju Z P, Liu X W 2010 *Acta Phys. Sin.* **59** 199 (in Chinese)[曹午飞、鞠志萍、刘小伟 2010 物理学报 **59** 199]

[2] Igor M, Den Y 2010 *Acta Phys. Sin.* **59** 1396 (in Chinese) [Igor M、邓 勇 2010 物理学报 **59** 1396]

[3] Xie W X, Xu W 2005 *Acta Phys. Sin.* **54** 1105 (in Chinese) [谢文贤、徐 伟 2005 物理学报 **54** 1105]

[4] Xiao F H, Yan G R 2004 *Acta Phys. Sin.* **53** 396 (in Chinese) [肖方红、闫桂荣 2004 物理学报 **53** 396]

[5] Cheng C, Ansari R 2005 *IEEE Sig. Proc. Lett.* **12** 242

[6] Gong Y X, Yang H W, Hu W D, Yu W X 2008 *J. Electron. Inform. Techn.* **34** 941 (in Chinese) [龚亚信、杨宏文、胡卫东、郁文贤 2008 电子与信息学报 **34** 941]

[7] Duan Z H, Cai Z X, Yu J X 2008 *Acta Autom. Sin.* **34** 581 (in Chinese) [段琢华、蔡自兴、于金霞 2008 自动化学报 **34** 581]

[8] Du Z C, Liu L X, Tang B 2006 *Chin. Phys.* **15** 2481

[9] Du Z C, Tang B, Li K 2006 *Acta Phys. Sin.* **55** 999 (in Chinese)[杜正聪、唐 斌、李 可 2006 物理学报 **55** 999]

[10] Torma P, Szepesvari C 2005 *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis* (Philadelphia: Springer) p58

[11] Song Z H, Wu X D 2008 *Chin. Phys. B* **17** 3241

[12] Li L Q, Ji H B, Luo J H 2007 *J. Xidian Univ.* **34** 233 (in Chinese)[李良群、姬红兵、罗军辉 2007 西安电子科技大学学报 **34** 233]

[13] Sui S L, Zao X W 2008 *J. Syst. Sim.* **20** 4971 (in Chinese) [隋树林、赵晓伟 2008 系统仿真学报 **20** 4971]

[14] Zhu Z Y, Jiang C S 2007 *J. Syst. Eng. Electron.* **29** 1596 (in Chinese) [朱志宇、姜长生 2007 系统工程与电子技术 **29** 1596]

[15] Hayakawa T 1990 *Ann. Inst. Statis. Math.* **42** 312

[16] Chikuse Y 2006 *Multivar. Anal.* **97** 1284

[17] Zhou G H, Jing Z L, Hu H T 2006 *J. Shanghai Jiaotong Univ.* **40** 1135 (in Chinese)[邹国辉、敬忠良、胡洪涛 2006 上海交通大学学报 **40** 1135]

Stiefel manifold particle filtering^{*}

Zhu Zhi-Yu[†] Yang Guan-Xiao

(School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China)

(Received 16 August 2009; revised manuscript received 5 July 2010)

Abstract

In order to solve the problems of particle degeneration and lackness of diversity of particle filter, a new particle filter based on Stiefel manifold (SMPF) is proposed in this paper. In the SMPF the system model is based on Stiefel manifold, Langevin distribution is used as a prior density, the matrix normal distribution serves as a likelihood function, and particle is sampled on the manifold distribution. First, manifold is embedded in Euclidean space, then the mean of particles is calculated in Euclidean space and its result is projected back to embedded manifold. So the influence on variance of particle weight caused by statistic characteristics of noise is removed, and a kind of universal selecting scheme of important probability density function is acquired which is hardly restrained to system state model. The simulation results based on univariate nonstationary growth model nonlinear system indicate that the SMPF works much better than scentless particle filter in real-time performance, robustness, filtering precision and filtering efficiency.

Keywords: particle filters, Stiefel manifold, univariate nonstationary growth model

PACC: 0250, 0260, 0547, 0290

^{*} Project supported by the Natural Science Foundation of Institution of Higher Education of Jiangsu Province, China (Grant No. 06KJB510030) and the National Natural Science Foundation of China (Grant No. 61075028).

[†] E-mail: zzydzz@163.com