

## 递推更新高斯粒子滤波器

张勇刚, 王刚<sup>†</sup>, 黄玉龙, 李宁

(哈尔滨工程大学 自动化学院, 黑龙江 哈尔滨 150001)

**摘要:** 传统高斯粒子滤波算法(Gaussian particle Filter, GPF)中, 粒子的重要性密度函数是由高斯滤波器结合当前最新量测来构建的. 由于传统高斯滤波器在量测更新阶段直接利用量测对状态进行线性更新, 在某些条件下会导致所构建的重要性密度函数并不能很好地近似状态真实分布. 为了解决这一问题, 结合递推更新的思想, 本文推导出了递推更新高斯滤波器(recursive update Gaussian filter, RUGF)的一般结构. 并在此基础上, 选用RUGF来构建粒子滤波的重要性密度函数, 从而提出了基于递推更新的高斯粒子滤波算法(recursive update gaussian particle filter, RUGPF). 仿真表明, 在非线性和非线性系统状态估计问题中, 递推更新可以很好的利用量测信息, 相比于传统的GPF, 本文所提出的RUGPF滤波算法可以提供更高精度的估计结果.

**关键词:** 高斯滤波器; 粒子滤波; 递推算法; 非线性滤波

**中图分类号:** TP273      **文献标识码:** A

## Recursive update Gaussian particle filter

ZHANG Yong-gang, WANG Gang<sup>†</sup>, HUANG Yu-long, LI Ning

(College of Automation, Harbin Engineering University, Harbin Heilongjiang 150001, China)

**Abstract:** In traditional Gaussian particle filter (GPF), sample importance density function is constructed through combining the latest measurements based on Gaussian filter (GF). However, in measurement update of the traditional GF, since the measurement value is assimilated directly based on the linear update rule, the constructed sample importance density function may not be approximate to the real posterior distribution under certain conditions. To solve this problem, we propose a new recursive update GF (RUGF) based on the recursive update idea and give out its general framework. On this basis, a new sample importance density function is constructed by using RUGF, based on which a new recursive update GPF (RUGPF) can be derived. Simulation results demonstrate that recursive update idea can assimilate the measurement information commendably, and compared with traditional GPF, the proposed filter has higher estimation accuracy for state estimation in nonlinear systems.

**Key words:** Gaussian filter; particle filter; recursive algorithm; nonlinear filtering

### 1 引言(Introduction)

粒子滤波(particle filter, PF)是一种基于Monte Carlo仿真的最优回归贝叶斯滤波算法. 它以适用于任意非线性和不受高斯噪声假设限制的优点而被人们广泛应用于目标跟踪、计算机视觉、无线通讯和机器学习等领域<sup>[1-5]</sup>. 然而, 在实际过程中PF也存在一定的局限性, 主要表现为: 粒子退化与贫化、计算量庞大以及该如何选取重要性密度函数. 其中, 就该如何选取重要性密度函数这一问题, 文献[6]中指出最优重要性密度函数的选取应与当前时刻量测值条件相关. 基于

这一思想, 近年来出现了许多基于PF的非线性滤波新方法, 主要有扩展卡尔曼粒子滤波<sup>[6-7]</sup>(extended Kalman particle filter, EPF)、无迹卡尔曼粒子滤波<sup>[6]</sup>(unscented Kalman particle filter, UPF)、中心差分卡尔曼粒子滤波<sup>[8]</sup>(center difference Kalman particle filter, CDPF)、容积粒子滤波<sup>[9]</sup>(cubature Kalman particle filter, CPF)、高斯粒子滤波<sup>[10]</sup>(Gaussian particle filter, GPF)、高斯求和粒子滤波<sup>[11]</sup>(Gaussian sum particle filter, GSPF)等滤波算法. 这些滤波方法原理相同, 它们利用非线性高斯滤波器结合最新量测

收稿日期: 2015-05-14; 录用日期: 2015-10-12.

<sup>†</sup>通信作者. E-mail: wanggang2016@126.com.

本文责任编辑: 潘泉.

国家自然科学基金项目(61201409, 61371173), 中国博士后科学基金(2013M530147, 2014T70309), 黑龙江省博士后基金(LBH-Z13052, LBH-TZ0505), 哈尔滨工程大学中央高校基本科研业务费专项基金(HEUCFQ20150407)资助.

Supported by National Natural Science Foundation of China (61201409, 61371173), China Postdoctoral Science Foundation (2013M530147, 2014T70309), Heilongjiang Postdoctoral Foundation (LBH-Z13052, LBH-TZ0505) and Fundamental Research Funds for the Central University of Harbin Engineering University (HEUCFQ20150407).

为每个粒子提供重要性密度函数,以抑制粒子的退化现象,从而提高了粒子滤波算法的效率和精度。

然而,由于传统非线性高斯滤波器本质上是线性卡尔曼滤波器的一种近似形式,都是基于线性最小方差准则来近似状态后验概率密度函数。其量测更新部分依据线性最小方差准则使量测对状态一步预测进行线性修正,限制了滤波器精度的进一步提高,进而导致基于传统非线性高斯滤波器的粒子滤波算法精度受限。文献[12]指出,在非线性情况下,依据线性最小方差准则进行量测更新并不能精确得到状态后验均值和方差,甚至可能会导致滤波发散。同时,文献[13]也说明当系统量测方程为非线性时,基于线性最小方差准则的滤波器在均方误差意义上并不是最优的。文中在EKF的基础上,提出一种递推更新(recursive update, RU)的思想,依据量测函数本身的非线性梯度逐步地对状态一步预测进行更新,仿真结果表明递推量测更新相比于传统的线性最小方差更新可以提供更高精度的状态估计,但由于EKF对非线性函数进行线性化处理,导致其状态估计误差协方差更新方程在非线性情况下并不准确。因此当滤波增益矩阵发生显著变化时,由EKF得到的状态后验估计误差协方差与真实的状态估计误差协方差严重不符,影响其滤波精度。

为了提高传统高斯粒子滤波器精度,本文在文献[13]的基础上,首先将RU思想推广至非线性高斯滤波器中,采用数值方法对高斯积分进行近似,避免了上述由EKF线性化引发的问题,从而得到了递推更新高斯滤波器(recursive updated Gaussian filter, RUGF)的一般结构。并在此基础上,选用RUGF为PF提供重要性密度函数,从而提出了基于递推更新的高斯粒子滤波算法(recursive updated Gaussian particle filter, RUGPF)。仿真结果表明结合了RU思想的非线性高斯滤波器所构造的重要性密度函数更能接近状态真实的后验分布,从而使得RUGPF具有更高的滤波精度。

本文首先介绍了传统非线性高斯滤波器的一般结构,并指出在传统非线性高斯滤波器量测更新阶段中,根据线性最小方差准则进行滤波估计并不能得到很好的估计效果;其次,利用递推更新可以紧密联系量测函数非线性这一优点,推导出RUGF的一般结构;最后,本文以RUGF来构造PF中的重要性密度函数,提出了RUGPF滤波算法。试验仿真验证了该改进算法的有效性和优越性。

## 2 非线性高斯滤波器(Nonlinear Gaussian filter)

考虑如下所示非线性离散动态系统模型:

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k) + \mathbf{w}_k, \quad (1)$$

$$\mathbf{z}_{k+1} = h_{k+1}(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1}, \quad (2)$$

其中:  $\mathbf{x}_k \in \mathbb{R}^n$ ,  $\mathbf{z}_{k+1} \in \mathbb{R}^m$ 分别为系统的状态向量和量测向量,  $n, m$ 为系统的状态维数和量测维数,  $f(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $h(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}^m$ 分别为系统的状态转移函数和量测函数,  $\mathbf{w}_k, \mathbf{v}_k$ 分别为系统噪声和量测噪声,且  $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$ ,  $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$ ,  $\mathbf{w}_k$ 与 $\mathbf{v}_k$ 不相关。依据线性最小方差准则,非线性高斯滤波器的一般结构如下<sup>[14]</sup>:

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + K_{k+1}(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}), \\ P_{k+1|k+1} = P_{k+1|k} - K_{k+1}P_{k+1|k}^z K_{k+1}^T, \\ K_{k+1} = P_{k+1|k}^{xz} (P_{k+1|k}^z)^{-1}, \end{cases} \quad (3)$$

其中

$$\begin{cases} \hat{\mathbf{x}}_{k+1|k} = E[\mathbf{x}_{k+1} | \mathbf{Z}_k], \\ P_{k+1|k} = E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})^T | \mathbf{Z}_k], \\ \hat{\mathbf{z}}_{k+1|k} = E[\mathbf{z}_{k+1} | \mathbf{Z}_k], \\ P_{k+1|k}^z = E[(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k})(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k})^T | \mathbf{Z}_k], \\ P_{k+1|k}^{xz} = E[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k})(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k})^T | \mathbf{Z}_k], \end{cases} \quad (4)$$

式中:  $\mathbf{Z}_k = \{z_1, z_2, \dots, z_k\}$ ,表示1到 $k$ 时刻的量测序列集合。非线性滤波就是根据含有噪声的量测序列 $\mathbf{Z}_k$ 来获得系统状态的最优估计。

从上述非线性高斯滤波器一般结构中可知,其量测更新阶段主要利用了线性最小方差准则使量测 $\mathbf{z}_{k+1}$ 对状态一步预测 $\hat{\mathbf{x}}_{k+1|k}$ 进行修正。而文献[12-13]指出,在系统量测函数为非线性情况时,采用线性最小方差估计准则来融合量测信息并不能获得最优的状态估计结果。因此,当系统量测函数为非线性时,需要考虑量测与状态之间的非线性关系,寻找一种更为合适的量测更新算法来提高状态估计精度。

## 3 基于递推更新的高斯粒子滤波算法(The Gaussian particle filter based on recursive update)

为了解决第2节中提出的问题,文献[13]紧密结合量测函数的非线性梯度,提出一种RUF算法,核心在于对状态一步预测进行多次、递推式地更新。但RUF算法是在EKF的基础上实施的,其滤波精度受到限制。原因在于递推更新过程中,每经过一次雅克比矩阵计算后,沿用了标准卡尔曼滤波结构,导致其状态估计误差协方差更新方程在非线性情况下并不准确,若滤波增益矩阵发生显著变化时,由EKF得到的状态后验估计误差协方差与真实的状态估计误差协方差将严重不符,进而导致RUF精度下降。基于以上情况,本文将递推思想推广至一般高斯滤波器的结构中,得到更高滤波精度的RUGF,并在此基础上,提出基于递推更新的高斯粒子滤波算法。下面本文将首先介绍递推更

新思想.

### 3.1 递推更新(Recursive update)

在RU过程中, 最优增益被假设为时刻随着量测函数的梯度而变化, 而且每一次更新时, 量测信息按一定比例系数对状态一步预测进行修正. 其具体实施过程如下<sup>[13]</sup>:

$$\begin{aligned} & \mathbf{C}_{k+1}^{(0)} = 0, \mathbf{P}_{k+1|k+1}^{(0)} = \mathbf{P}_{k+1|k}, \hat{\mathbf{x}}_{k+1|k+1}^{(0)} = \hat{\mathbf{x}}_{k+1|k}; \\ & \text{for } i = 1 : N, \\ & \mathbf{H}_{k+1}^{(i-1)} = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k+1}^{(i-1)}}; \\ & \mathbf{K}_{k+1}^{(i-1)} = \\ & \quad \frac{1}{N-i+1} (\mathbf{P}_{k+1|k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)} + \mathbf{C}_{k+1}^{(i-1)}). \\ & \quad (\mathbf{H}_{k+1}^{(i-1)} \mathbf{P}_{k+1|k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)\text{T}} + \mathbf{R}_{k+1} + \\ & \quad \mathbf{H}_{k+1}^{(i-1)} \mathbf{C}_{k+1}^{(i-1)} + \mathbf{C}_{k+1}^{(i-1)\text{T}} \mathbf{H}_{k+1}^{(i-1)\text{T}})^{-1}; \\ & \hat{\mathbf{x}}_{k+1|k+1}^{(i)} = \\ & \quad \hat{\mathbf{x}}_{k+1|k+1}^{(i-1)} + \mathbf{K}_{k+1}^{(i-1)} (\mathbf{z}_{k+1} - h(\hat{\mathbf{x}}_{k+1|k+1}^{(i-1)})); \\ & \mathbf{P}_{k+1|k+1}^{(i)} = \\ & \quad (\mathbf{I}_{n \times n} - \mathbf{K}_{k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)}) \mathbf{P}_{k+1|k+1}^{(i-1)} \cdot \\ & \quad (\mathbf{I}_{n \times n} - \mathbf{K}_{k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)})^{\text{T}} + \mathbf{K}_{k+1}^{(i-1)} \mathbf{R}_{k+1} \mathbf{K}_{k+1}^{(i-1)\text{T}} - \\ & \quad (\mathbf{I}_{n \times n} - \mathbf{K}_{k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)}) \mathbf{C}_{k+1}^{(i-1)} \mathbf{K}_{k+1}^{(i-1)\text{T}} - \\ & \quad \mathbf{K}_{k+1}^{(i-1)} \mathbf{C}_{k+1}^{(i-1)\text{T}} (\mathbf{I}_{n \times n} - \mathbf{K}_{k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)})^{\text{T}}; \\ & \mathbf{C}_{k+1}^{(i)} = \\ & \quad (\mathbf{I}_{n \times n} - \mathbf{K}_{k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)}) \mathbf{C}_{k+1}^{(i-1)} - \mathbf{K}_{k+1}^{(i)} \mathbf{R}_{k+1} \\ & \text{end for} \\ & \hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k+1}^{(N)}, \mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k+1}^{(N)}. \end{aligned}$$

其中:  $N$ 为递推次数,  $\mathbf{C}_{k+1}$ 定义为 $\mathbf{x}_{k+1}$ 与量测噪声之间的协方差.

分析上述流程可知, 当 $N = 1$ 时, 上述更新过程与传统高斯滤波器量测更新阶段等价. 当递推次数 $N \geq 1$ 时, 在递推更新中, 量测信息被递推式的用于修正状态一步预测, 而且随着非线性量测函数的梯度, 连续计算雅克比矩阵 $\mathbf{H}_{k+1}^{(i-1)}$ , 因此递推更新的总体更新效果是非线性的, 而且与量测函数本身非线性度紧密相关. 值得强调的是, 递推更新与迭代滤波器并不相同. 以文献[15]中提出的迭代sigma点卡尔曼滤波器为例, 迭代过程的本质是采用Gauss-Newton法来求解系统的极大后验估计. 每一次的迭代使用了全部的量测信息对状态一步预测进行修正, 目的是为了状态一步预测可以更进一步的接近状态真实值. 因此直观上可以认为, 迭代的作用仅局限于减小状态一步预测与真实值之间的误差. 而且Gauss-Newton法也存在要求系统初始误差较小、不能保证局部收敛性等问题. 递推更新则不同, 其更新过程可以理解将当前时刻

的量测按照递推次数分为 $N$ 等份, 每次对状态一步预测进行渐进式地更新. 其创新意义在于突破传统卡尔曼类滤波器线性更新的思想, 实现量测对状态的非线性更新.

### 3.2 递推更新高斯滤波器(Recursive update Gaussian filter)

接下来本文将递推更新思想推广至一般非线性高斯滤波器结构中, 利用数值计算方法对高斯积分进行近似, 以提高RUF的精度.

借鉴文献[13]提出的RU思想, 即结合系统量测函数的非线性, 渐近地利用量测信息对状态一步预测进行修正. 这里定义 $\mathbf{D}_{k+1} = \text{E}[(h(\mathbf{x}_{k+1}) - \hat{\mathbf{z}}_{k+1|k}) \mathbf{v}_{k+1}^{\text{T}}]$ , 参照文献[13], 可以得到在非线性高斯滤波器中, 其最优增益矩阵为

$$\mathbf{K}_{k+1} = (\mathbf{P}_{xz,k+1} + \mathbf{C}_{k+1}) \cdot (\mathbf{P}_{z,k+1} + \mathbf{D}_{k+1} + \mathbf{D}_{k+1}^{\text{T}})^{-1}, \quad (5)$$

其中 $\mathbf{P}_{z,k+1}$ ,  $\mathbf{P}_{xz,k+1}$ 分别为量测一步预测协方差、状态和量测一步预测互协方差. 注意, 在文献[13]中, 采用了一阶泰勒展开线性化的方式求得

$$\begin{aligned} \mathbf{P}_{xz,k+1} &= \mathbf{P}_{k+1|k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)\text{T}}, \\ \mathbf{P}_{z,k+1} &= \mathbf{H}_{k+1}^{(i-1)} \mathbf{P}_{k+1|k+1}^{(i-1)} \mathbf{H}_{k+1}^{(i-1)\text{T}} + \mathbf{R}_{k+1}. \end{aligned}$$

这种求解方法的问题在于方差的求解精度受限.

此时,  $\mathbf{x}_{k+1}$ 的估计为

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}), \quad (6)$$

则状态后验估计误差为

$$\begin{aligned} \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1} &= \\ \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k} - \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}) &= \\ \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k} - \mathbf{K}_{k+1} (h(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} - \hat{\mathbf{z}}_{k+1|k}) &= \\ \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k} - \mathbf{K}_{k+1} (h(\mathbf{x}_{k+1}) - \hat{\mathbf{z}}_{k+1|k}) - \\ \mathbf{K}_{k+1} \mathbf{v}_{k+1}. & \quad (7) \end{aligned}$$

根据式(7), 可以得到状态后验估计误差协方差 $\mathbf{P}_{k+1|k+1}$ 为

$$\begin{aligned} \mathbf{P}_{k+1|k+1} &= \\ \text{E}[(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1})(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k+1})^{\text{T}} | \mathbf{Z}_{k+1}] &= \\ \mathbf{P}_{k+1|k} - (\mathbf{P}_{xz,k+1|k} + \mathbf{C}_{k+1}) \mathbf{K}_{k+1}^{\text{T}} - \\ \mathbf{K}_{k+1} (\mathbf{P}_{xz,k+1|k} + \mathbf{C}_{k+1})^{\text{T}} + \\ \mathbf{K}_{k+1} (\mathbf{P}_{z,k+1|k} + \mathbf{D}_{k+1} + \mathbf{D}_{k+1}^{\text{T}}) \mathbf{K}_{k+1}^{\text{T}}. & \quad (8) \end{aligned}$$

在递推过程中, 由于 $\mathbf{D}_{k+1}^{(i)}$ 与 $\mathbf{D}_{k+1}^{(i-1)}$ 无法建立直接意义上的解析表达式, 因此可采用如下方式进行近似处理:

$$\begin{aligned} \mathbf{D}_{k+1} &= \text{E}[(h(\mathbf{x}_{k+1}) - \hat{\mathbf{z}}_{k+1|k}) \mathbf{v}_{k+1}^{\text{T}}] = \\ \text{E}[(h(\mathbf{x}_{k+1}) - h(\hat{\mathbf{x}}_{k+1|k})) \mathbf{v}_{k+1}^{\text{T}}] &= \\ \text{E}[\mathbf{H}_{k+1} (\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1|k}) \mathbf{v}_{k+1}^{\text{T}}] &= \end{aligned}$$

$$\mathbf{H}_{k+1} \mathbf{C}_{k+1}, \quad (9)$$

其中  $\mathbf{H}_{k+1} = \frac{\partial h}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}}$ .

利用式(9)进行简化处理从而得到递推更新高斯滤波器的结构:

1) 状态预测.

$$\begin{aligned} \hat{\mathbf{x}}_{k+1|k} &= \int f(\mathbf{x}_k) \times N(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) d\mathbf{x}_k, \\ \mathbf{P}_{k+1|k} &= \int f(\mathbf{x}_k) f^T(\mathbf{x}_k) \times N(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) d\mathbf{x}_k - \\ &\quad \hat{\mathbf{x}}_{k+1|k} \hat{\mathbf{x}}_{k+1|k}^T + \mathbf{Q}_k. \end{aligned}$$

2) 递推量测更新.

$$\text{a) } \mathbf{C}_{k+1}^{(0)} = 0, \mathbf{P}_{k+1|k+1}^{(0)} = \mathbf{P}_{k+1|k}, \hat{\mathbf{x}}_{k+1|k+1}^{(0)} = \hat{\mathbf{x}}_{k+1|k};$$

b) for  $i = 1 : N$ ,

$$\begin{aligned} \hat{\mathbf{z}}_{k+1|k}^{(i-1)} &= \\ &\int h(\mathbf{x}_{k+1}) \times N(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1|k+1}^{(i-1)}, \mathbf{P}_{k+1|k+1}^{(i-1)}) d\mathbf{x}_{k+1}, \end{aligned}$$

$$\begin{aligned} \mathbf{P}_{z,k+1|k}^{(i-1)} &= \\ &\int h(\mathbf{x}_{k+1}) h^T(\mathbf{x}_{k+1}) \times \\ &N(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1|k+1}^{(i-1)}, \mathbf{P}_{k+1|k+1}^{(i-1)}) d\mathbf{x}_{k+1} - \end{aligned}$$

$$\hat{\mathbf{z}}_{k+1|k}^{(i-1)} \hat{\mathbf{z}}_{k+1|k}^{(i-1)T} + \mathbf{R}_{k+1},$$

$$\mathbf{P}_{xz,k+1|k}^{(i-1)} =$$

$$\int \mathbf{x}_{k+1} h^T(\mathbf{x}_{k+1}) \times$$

$$N(\mathbf{x}_{k+1}; \hat{\mathbf{x}}_{k+1|k+1}^{(i-1)}, \mathbf{P}_{k+1|k+1}^{(i-1)}) d\mathbf{x}_{k+1} -$$

$$\hat{\mathbf{x}}_{k+1|k}^{(i-1)} \hat{\mathbf{z}}_{k+1|k}^{(i-1)T},$$

$$\mathbf{H}_{k+1}^{(i-1)} = \frac{\partial h}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k+1}^{(i-1)}},$$

$$\mathbf{D}_{k+1}^{(i-1)} = \mathbf{H}_{k+1}^{(i-1)} \mathbf{C}_{k+1}^{(i-1)},$$

$$\mathbf{K}_{k+1}^{(i)} =$$

$$\frac{1}{N - i + 1} (\mathbf{P}_{xz,k+1|k}^{(i-1)} + \mathbf{C}_{k+1}^{(i-1)}).$$

$$(\mathbf{P}_{z,k+1|k}^{(i-1)} + \mathbf{D}_{k+1}^{(i-1)} + \mathbf{D}_{k+1}^{(i-1)T})^{-1},$$

$$\hat{\mathbf{x}}_{k+1|k+1}^{(i)} = \hat{\mathbf{x}}_{k+1|k+1}^{(i-1)} + \mathbf{K}_{k+1}^{(i)} (\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1|k}^{(i-1)}),$$

$$\mathbf{P}_{k+1|k+1}^{(i)} =$$

$$\mathbf{P}_{k+1|k+1}^{(i-1)} - (\mathbf{P}_{xz,k+1|k}^{(i-1)} + \mathbf{C}_{k+1}^{(i-1)}) \mathbf{K}_{k+1}^{(i)T} -$$

$$\mathbf{K}_{k+1}^{(i)} (\mathbf{P}_{z,k+1|k}^{(i-1)} + \mathbf{C}_{k+1}^{(i-1)})^T +$$

$$\mathbf{K}_{k+1}^{(i)} (\mathbf{P}_{z,k+1|k}^{(i-1)} + \mathbf{D}_{k+1}^{(i-1)} + \mathbf{D}_{k+1}^{(i-1)T}) \mathbf{K}_{k+1}^{(i)T},$$

$$\mathbf{C}_{k+1}^{(i)} = (\mathbf{I} - \mathbf{K}_{k+1}^{(i)} \mathbf{H}_{k+1}^{(i-1)}) \mathbf{C}_{k+1}^{(i-1)} - \mathbf{K}_{k+1}^{(i)} \mathbf{R}_{k+1}$$

end for

$$\text{c) } \hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k+1}^{(N)}, \mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k+1}^{(N)},$$

其中  $N$  为递推次数.

比较可知,上述过程中量测更新过程严格服从递

推更新思想,而且状态预测和递推量测更新中的高斯积分可以采用不同的数值计算方法,如无迹变换或容积求积准则等计算获得,避免了EKF中由一阶泰勒展开线性化引发的状态估计误差协方差更新方程不准确的问题.虽然在该滤波器结构中,依然需要计算雅克比矩阵  $\mathbf{H}_{k+1}^{(i-1)}$ ,但它的作用只是用来近似处理  $\mathbf{D}_{k+1} = \mathbf{E}[(h(\mathbf{x}_{k+1}) - \hat{\mathbf{z}}_{k+1|k}) \mathbf{v}_{k+1}^T]$ ,与文献[13]中的RUF是有所不同的.基于此,RUGF将会比文献[13]中提出的RUF性能更为优越.

### 3.3 基于递推更新的高斯粒子滤波改进算法(The modified algorithm of Gaussian particle filter based on recursive update)

标准的PF按照先验概率分布来随机抽取粒子,当系统的量测处于先验密度函数尾部或者量测精度较高时,会出现粒子退化、枯竭的现象.解决这一问题的有效途径之一就是选取合适的重要性密度函数.理论上,重要性密度函数的选取应当尽量接近状态的真实分布<sup>[3]</sup>.第3.1节中已说明递推更新可以更好地利用量测对状态进行修正,从而获得更接近真实分布的状态后验估计.因此,本文利用RUGF在PF中设计重要性密度函数以得到每个粒子的建议分布,避免出现粒子退化,进而得到基于递推更新的高斯粒子滤波改进算法.

RUGPF算法代码如下:

$$\{[\mathbf{x}_k^j, \omega_k^j]_{j=1}^M\} = \text{RUGPF}[\{\mathbf{x}_{k-1}^j, \omega_{k-1}^j\}_{j=1}^M, \mathbf{z}_k].$$

1) 初始化:  $k = 0$ .

for  $j = 1, 2, \dots, M$ ,  $\mathbf{x}_0^j \sim p(\mathbf{x}_0)$ ,  $\omega_0^j = 1/M$

$$\hat{\mathbf{x}}_0^j = \mathbf{E}[\mathbf{x}_0^j]$$

$$\hat{\mathbf{P}}_0^j = \mathbf{E}[(\mathbf{x}_0^j - \hat{\mathbf{x}}_0^j)(\mathbf{x}_0^j - \hat{\mathbf{x}}_0^j)^T]$$

end for

2)  $k = 1$ .

重要性函数采样:

for  $j = 1, 2, \dots, M$ .

① RUGF设计重要性密度函数并更新每个粒子:

$$[\hat{\mathbf{x}}_k^j, \hat{\mathbf{P}}_k^j] = \text{RUGF}[\hat{\mathbf{x}}_{k-1}^j, \hat{\mathbf{P}}_{k-1}^j, \mathbf{z}_k].$$

② 重新产生粒子:

$$\mathbf{x}_k^j \sim q(\hat{\mathbf{x}}_k^j | \mathbf{x}_{k-1}^j, \mathbf{z}_k) = N(\hat{\mathbf{x}}_k^j, \hat{\mathbf{P}}_k^j).$$

③ 计算粒子权值:

$$\omega_k^j = \omega_{k-1}^j \frac{p(\mathbf{z}_k | \mathbf{x}_k^j) p(\mathbf{x}_k^j | \mathbf{x}_{k-1}^j)}{q(\hat{\mathbf{x}}_k^j | \mathbf{x}_{k-1}^j, \mathbf{z}_k)}.$$

end for

权值归一化:  $\omega_k^j = \omega_k^j / \sum_{i=1}^M \omega_k^i$ .

重采样:

$$[\{\mathbf{x}_k^j, \omega_k^j\}_{j=1}^M] = \text{resample}[\{\mathbf{x}_{k-1}^j, \omega_{k-1}^j\}_{j=1}^M].$$

状态估计:

$$\hat{\mathbf{x}}_k = \frac{1}{M} \sum_{j=1}^M \mathbf{x}_k^j,$$

$$\hat{\mathbf{P}}_k = \frac{1}{M} \sum_{j=1}^M (\mathbf{x}_k^j - \hat{\mathbf{x}}_k)(\mathbf{x}_k^j - \hat{\mathbf{x}}_k)^T.$$

3) 令  $k = k + 1$ , 重复步骤2)直至滤波结束.

其中  $M$  为粒子总数, RUGF 的实施步骤已在第3.2节中给出.

本文目的不仅在于给出一个特定的RUGPF算法, 而是给出了RUGPF算法的一般框架, 利用不同的数值计算方法, 如UKF, CDKF, CKF, 可以获得不同的RUGF算法, 如RUUKF, RUCDKF和RUCKF, 进一步, 利用这些算法为每个粒子提供建议分布, 从而可以获得不同的RUGPF算法. 在接下来RU过程计算复杂度分析和实验仿真中, 本文将利用容积计算准则来实施RUGF算法.

#### 4 RUGF计算复杂度分析(Computation complexity analysis of RUGF)

滤波算法的计算复杂度是一个在工程实用中必须考虑的一个问题. 相比于传统的GPF算法, 本文提出的RUGPF算法的不同点在于对每个粒子设计重要性密度函数时所用到的RUGF算法中增加了RU过程. 因此这里将着重分析对于单个粒子而言, 传统的用来设计粒子重要性密度函数的滤波算法与本文提出的RUGF算法之间计算复杂度的差别. 以容积求积准则为例, 容积粒子滤波中用CKF来设计粒子重要性密度函数, 而相对应的递推更新容积粒子滤波则以RUCKF来产生. RU过程的计算复杂度主要由递推更新次数  $N$  定, 相比于CKF, RUCKF的  $N$  次递推更新过程多出了  $N - 1$  次的递推更新计算量和1次额外的  $\mathbf{C}_{k+1}^{(i)}$  的计算. 其中单次递推更新计算量如表1所示.

表1 单次递推更新过程计算量

Table 1 Calculation amount of single recursive update process

计算项	计算量
$\hat{\mathbf{z}}_{k+1 k}^{(i-1)}$	$4n^2m$
$\mathbf{P}_{z,k+1 k}^{(i-1)}$	$4m^2n + 3m^2$
$\mathbf{P}_{xz,k+1 k}^{(i-1)}$	$4n^2m + 2mn$
$\mathbf{H}_{k+1}^{(i-1)}$	$mn$
$\mathbf{K}_{k+1}^{(i)}$	$m^3 + 6m^2n + 3mn$
$\hat{\mathbf{x}}_{k+1 k+1}^{(i)}$	$2mn + m$
$\mathbf{P}_{k+1 k+1}^{(i)}$	$5m^2n + 5n^2m + n^2 + 6mn$
$\mathbf{C}_{k+1}^{(i)}$	$2m^2n + 4n^2m - n^2$

表1中  $\hat{\mathbf{z}}_{k+1|k}^{(i-1)}$ ,  $\mathbf{P}_{z,k+1|k}^{(i-1)}$  和  $\mathbf{P}_{xz,k+1|k}^{(i-1)}$  的计算量参照

文献[16]. 由文献[16]可知, CKF算法的计算复杂度  $f_{\text{CKF}}$  为  $\frac{10}{3}n^3 + 10n^2 + 10n^2m + 8m^2n + 2mn + m^3 + 3m^2 + m$ .

而结合表1, 可以得到RUCKF算法的计算复杂度  $f_{\text{CKF}}$  为  $\frac{10}{3}n^3 + 10n^2 + N(17n^2m + 17m^2n + 14mn + m^3 + 3m^2 + m)$ . 由上式可以很直观的得出: 相比于CKF, RUCKF计算量的增加主要是由于  $N$  次递推更新过程的引入. 因此, 在实际工程实用中, RUGF中递推更新参数  $N$  的选取需要权衡工程计算量和所需精度.

#### 5 仿真(Simulation)

下面本文将采用经典的粒子滤波仿真数值模型—单变量非增长模型和工程实例中的目标跟踪问题来验证本文提出的RUGPF算法的有效性.

##### 5.1 单变量非平稳增长模型(Univariate non-stationary growth model)

单变量非平稳增长模型为经济领域广泛使用的非线性模型, 为典型的一维非线性模型<sup>[3,9,17]</sup>, 其数学模型描述如下:

$$\begin{cases} \mathbf{x}_t = \frac{1}{2}\mathbf{x}_{t-1} + \frac{25\mathbf{x}_{t-1}}{1 + \mathbf{x}_{t-1}^2} + 8 \cos(1.2(t-1)) + \mathbf{w}_{t-1}, \\ \mathbf{z}_t = \frac{\mathbf{x}_t^2}{20} + \mathbf{v}_t, \end{cases} \quad (10)$$

其中: 系统噪声  $\mathbf{w}_{t-1} \sim N(0, 1)$ , 量测噪声  $\mathbf{v}_t \sim N(0, 0.1)$ .

在这里, 定义状态估计均方根误差RMSE来评价滤波器的性能:

$$\text{RMSE} = \sqrt{\frac{1}{L} \sum_{i=1}^L (\mathbf{x}_k^i - \hat{\mathbf{x}}_{k|k}^i)^2}, \quad (11)$$

式中:  $\mathbf{x}_k^i$  表示第  $i$  次仿真时  $k$  时刻的状态真实值,  $\hat{\mathbf{x}}_{k|k}^i$  表示第  $i$  次仿真时  $k$  时刻的状态滤波估计值,  $L$  为 Monte Carlo 仿真次数. 仿真条件为: 状态初始真实值  $\mathbf{x}_{0|0} = 0.1$ , 状态仿真初始条件取  $\hat{\mathbf{x}}_{0|0} = 0$ ,  $\mathbf{P}_{0|0} = 1$ , 粒子数  $M = 500$ , RUGPF 中递推更新次数和ICPF 中的迭代次数  $N = 20$ , 仿真次数  $L = 100$ , 时间  $T = 60$  s.

在这里, 本文采用容积准则验证所提出的RUGPF算法. 因此本文将具体比较RUF, RUCKF, CPF, ICPF 和RUCPF的滤波性能. 为了更直观的比较各滤波器的估计性能, 表2列出了RUF, RUCKF, CPF, ICPF, RUCPF各滤波器RMSE均值. 结合图1和表2中可以看出, CPF的滤波效果最差, ICPF优于CPF. 原因在于ICPF选取的重要性密度函数为迭代CKF, 比CKF可以更精确的近似状态真实分布. 但CPF, ICPF的性能都

要差于RUF和RUCKF. 这说明了递推更新具有明显的优势. RUCKF的滤波效果要优于RUF, 说明了本文在文献[13]基础上推广而来的RUCKF算法的有效性. 在此基础上, RUCPF的滤波性能最优. 分析其原因, 是由于在RUCPF中, 递推更新可以更好的融合量测信息, 所以基于RUCKF构造的建议分布可以更好的近似状态后验分布, 使得RUCPF相比CPF, ICPF具有更高的滤波精度.

表3给出了RUF, RUCKF, CPF, ICPF, RUCPF各滤波器单步运行时间. CPF, ICPF和RUCPF的运行时间要明显高于RUF, RUCKF两个数量级, 其直接原因是由于粒子滤波中大量粒子造成的. ICPF进行了迭代更新, RUCPF进行了递推更新, 所以运行时间都要大于CPF. RUCPF的运行时间最长, 但其滤波性能也是最优的.

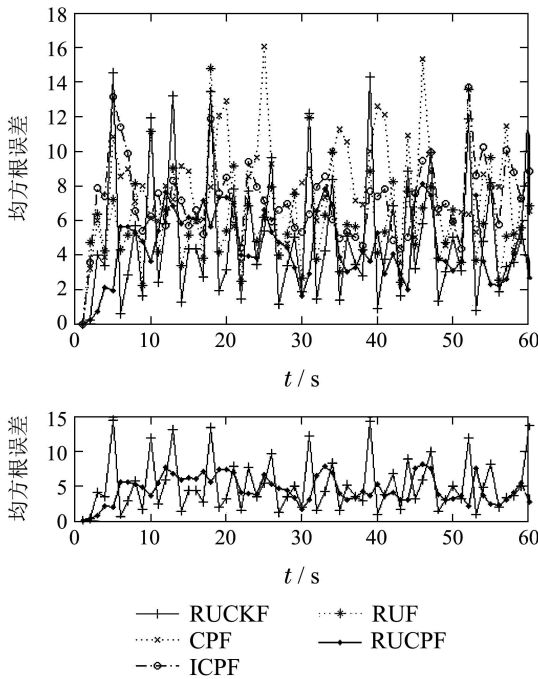


图1 RUF, RUCKF, CPF, ICPF, RUCPF状态估计均方根误差比较

Fig. 1 Comparisons of state estimate RMSEs of RUF, RUCKF, CPF, ICPF and RUCPF

表2 RUF, RUCKF, CPF, ICPF, RUCPF各滤波器RMSE均值比较

Table 2 Comparisons of averaged RMSEs of RUF, RUCKF, CPF, ICPF and RUCPF

滤波器	RMSE均值
RUF	5.9030
RUCKF	5.0450
CPF	7.7381
ICPF	7.1225
RUCPF	4.4753

表3 RUF, RUCKF, CPF, ICPF, RUCPF各滤波器单步运行时间比较

Table 3 Comparisons of one-step operation time of RUF, RUCKF, CPF, ICPF and RUCPF

滤波器	t/s
RUF	0.0012
RUCKF	0.0017
CPF	0.1353
ICPF	0.5957
RUCPF	0.8572

图2和表4给出了递推次数  $N = 2, 5, 10, 20$  时RUCPF的估计性能.  $N = 2$ 时, RUCPF的RMSE均值为7.3820, 与表2中CPF的RMSE均值7.7381较为接近, 说明此时由于递推更新步数较少, 其作用并不明显. 但随着 $N$ 的增大, RUCPF的RMSE均值逐渐减小, 说明滤波性能有所提升. 但RMSE均值下降的幅度在减缓, 说明当 $N$ 增加到一定程度时, RUCPF的性能不再提高. 因此, 在选用RUCPF时, 应当合适选取递推步数, 避免引入不必要的计算复杂度.

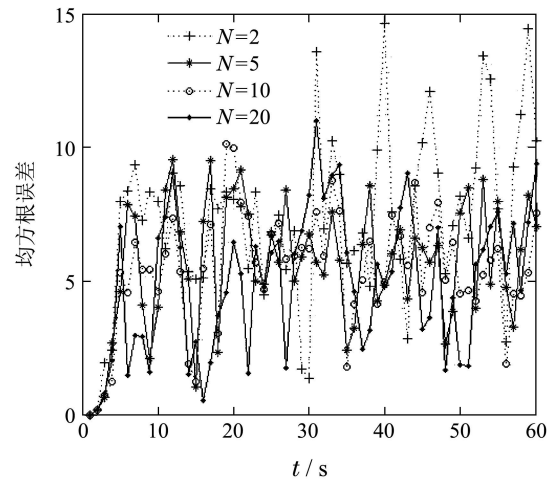


图2  $N = 2, 5, 10, 20$ 时, RUCPF状态估计均方根误差比较

Fig. 2 Comparisons of state estimate RMSEs of RUCPF when  $N = 2, 5, 10, 20$

表4  $N = 2, 5, 10, 20$ 时, RUCPF状态RMSE均值比较

Table 4 Comparisons of state averaged RMSEs of RUCPF when  $N = 2, 5, 10, 20$

RUCPF	RMSE均值/s
$N = 2$	7.3820
$N = 5$	5.6417
$N = 10$	5.0236
$N = 20$	4.7825

### 5.2 方位跟踪(Bearing tracking)

方位跟踪模型为二维非线性模型, 其离散模型如下<sup>[9, 17]</sup>:

$$\mathbf{x}_k = \begin{bmatrix} 0.9 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{w}_{k-1}. \quad (12)$$

状态  $\mathbf{x} = [\mathbf{x}(1), \mathbf{x}(2)]^T = [\mathbf{s}, \mathbf{t}]^T$ , 表示  $s-t$  平面内(笛卡尔坐标系)的位置, 系统噪声  $\mathbf{w}_{k-1} \sim N(0, \mathbf{Q})$ ,  $\mathbf{Q} = [1 \ 0.05; 0.05 \ 1]$ . 观测器位于点  $(\cos k, \sin k)$ , 对目标进行含噪声观测:

$$z_k = \tan^{-1}\left(\frac{t_k - \sin k}{s_k - \cos k}\right) + v_k, \quad (13)$$

其中观测噪声  $v_k \sim N(0, \mathbf{R})$ ,  $\mathbf{R} = 0.001$ .

仿真参数: 初始状态值  $\mathbf{x}_0 = [20, 5]^T$ , 初始协方差阵  $\mathbf{P}_0 = [0.1 \ 0; 0 \ 0.1]^T$ , 粒子数  $M = 200$ , RUCPF中递推更新次数和ICPF中迭代次数  $N = 20$ , 仿真时间  $T = 100$ , 进行 50 次 Monte Carlo 仿真. 最后按照式(11)定义的状态估计RMSE来评价滤波算法性能, 仿真结果如图3-4所示.

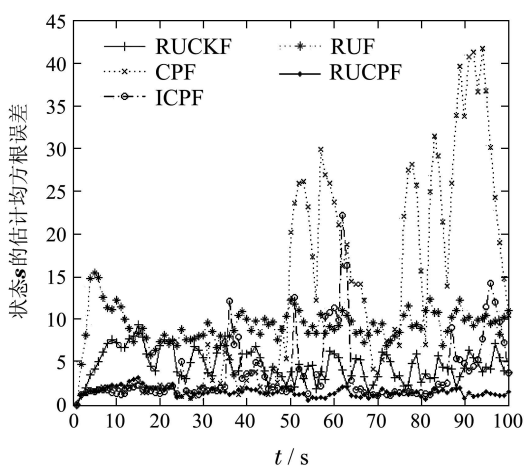


图3 RUF, RUCKF, CPF, ICPF, RUCPF状态s估计均方根误差比较

Fig. 3 Comparisons of state  $s$  RMSEs of RUF, RUCKF, CPF, ICPF and RUCPF

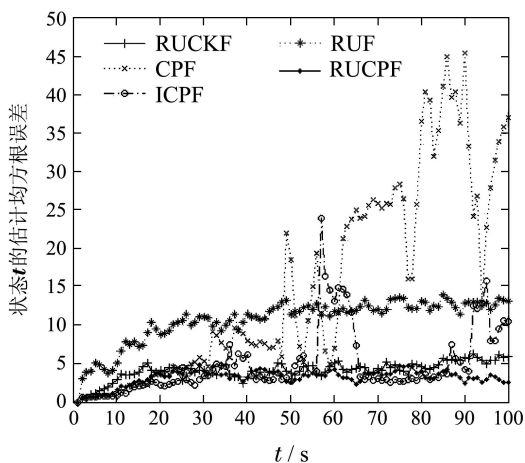


图4 RUF, RUCKF, CPF, ICPF, RUCPF状态t估计均方根误差比较

Fig. 4 Comparisons of state  $t$  RMSEs of RUF, RUCKF, CPF, ICPF and RUCPF

图3-4说明了RUCKF的滤波性能要优于RUF, ICPF要优于CPF, 且RUCPF算法性能最优. 再次证明了递推更新要优于迭代, 改进后的RUGF算法要优于文献[13]中的RUF算法, 以及本文所提出的基于RUCKF的RUCPF算法的有效性. 比较可以发现, 传统的CPF算法存在发散的趋势, ICPF可以缓解这一问题, 但在某些时刻依然会出现较大的估计误差. 而RUF, RUCKF和RUCPF对状态估计效果比较平稳, 且RUCPF估计精度最高. 原因在于, RUCPF在RUCKF的基础上, 采用递推更新思想, 更好的吸收量测信息, 为每个粒子提供更接近真实分布的建议分布, 从而获得更好的滤波效果.

表5给出了各滤波器单步运行时间. 同样, 在粒子滤波框架下, CPF的运行时间要高于RUF, RUCKF一个数量级. 由于迭代和递推, ICPF和RUCPF要比CPF花费更多的时间. RUCPF的运行时间最长.

表5 RUF, RUCKF, CPF, ICPF, RUCPF各滤波器单步运行时间比较

Table 5 Comparisons of one-step operation time of RUF, RUCKF, CPF, ICPF and RUCPF

滤波器	$t/s$
RUF	0.0014
RUCKF	0.0033
CPF	0.0616
ICPF	0.6468
RUCPF	0.7116

图5-6和表6给出了不同递推更新次数条件下, RUCPF对状态  $s, t$  的状态估计均方误差比较结果. 由图5-6可以看出, 当  $N = 2$  时, 递推更新作用并不十分明显, 导致RUCPF在某些时刻出现较大的状态估计均方根误差, 与图3-4中的CPF滤波趋势较接近.

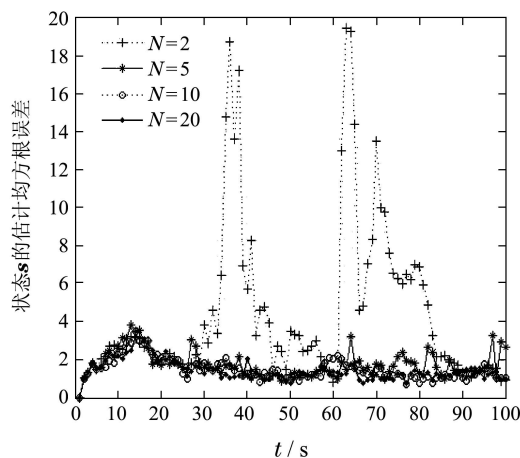


图5  $N = 2, 5, 10, 20$ 时, RUCPF状态s估计均方根误差比较  
Fig. 5 Comparisons of state  $s$  RMSEs of RUCPF when  $N = 2, 5, 10, 20$

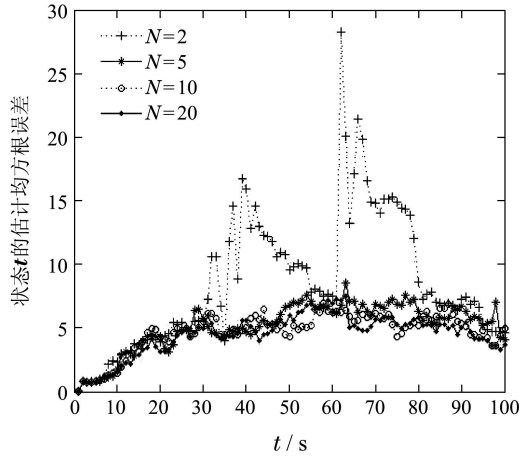


图6  $N = 2, 5, 10, 20$ 时, RUCPF状态 $t$ 估计均方根误差比较  
Fig. 6 Comparisons of state  $t$  RMSEs of RUCPF when  $N = 2, 5, 10, 20$

比较表6中状态 $s, t$ 均方根误差均值随着 $N$ 增加的变化过程可以发现, 当递推次数 $N$ 按5, 10, 20增加时, RUCPF的滤波性能逐渐提升, 但提升幅度也有减缓的趋势. 结合表5的分析结果可知, 在使用RUCPF进行状态估计时, 需要权衡系统对状态估计精度和所需时间的要求, 从而选取合适的递推更新次数.

表6  $N = 2, 5, 10, 20$ 时, RUCPF状态 $s, t$ 的RMSE均值比较

Table 6 Comparisons of state  $s$  and  $t$  averaged RMSEs of RUCPF when  $N = 2, 5, 10, 20$

RUCKF	$s$ RMSE均值	$t$ RMSE均值
$N = 2$	4.3650	8.4755
$N = 5$	1.7165	5.1248
$N = 10$	1.4831	4.6646
$N = 20$	1.4215	4.4612

## 6 结论(Conclusions)

在传统非线性高斯滤波器结构中, 量测更新阶段在面临量测函数为非线性时, 由于直接利用了线性最小方差准则导致其滤波估计结果并不是最优的. RU的思想在于紧密结合了量测函数的非线性, 对量测信息进行递推渐进利用, 可以提高滤波精度. 本文在传统非线性高斯滤波器的基础上, 推导出了RUGF的一般结构. 并基于这一结构, 具体提出了以RUGF构建重要性密度函数的粒子滤波改进算法RUGPF. 最后仿真结果验证了该算法的有效性和优越性.

## 参考文献(References):

- [1] BOLVIKEN E, ACKLAM J, CHRISTOPHERSEN N, et al. Monte Carlo filters for non-linear state estimation [J]. *Automatica*, 2001, 37(2): 177 – 183.
- [2] LIU J, CHEN R. Sequential Monte Carlo methods for dynamic systems [J]. *Journal of America Statistical Association*, 1998, 93(443): 1032 – 1044.
- [3] ZHAO Lin, WANG Xiaoxu, LI Liang, et al. *Nonlinear System Filtering Theory* [M]. Beijing: National Defense Industry Press, 2012.
- [4] (赵琳, 王晓旭, 李亮, 等. 非线性系统滤波理论 [M]. 北京: 国防工业出版社, 2012.)
- [4] DOUCET A, GODSILL S, ANDRIEU C. On sequential Monte Carlo sampling methods for Bayesian filtering [J]. *Statistics and Computing*, 2000, 10(3): 197 – 208.
- [5] 朱志宇. 粒子滤波算法及其应用 [M]. 北京: 科学出版社, 2010. (ZHU Zhiyu. *Particle Filtering Algorithm and Application* [M]. Beijing: Publishing House of Science, 2010.)
- [6] MERWE R, DOUCET A, FREITAS N, et al. The unscented particle filter [R] // *Technical Report CUED/F-INFENG/TR 380*. Cambridge: Cambridge University, 2000.
- [7] SOUBGUI F, BENHMIDA F, CHAARI A. Bayesian estimation via extended Kalman filtering for nonlinear stochastic systems [C] // *Proceedings of the 14th International Conference on Science and Techniques of Automatic Control & Computer Engineering*, Sousse: IEEE, 2013, 12: 89 – 95.
- [8] WANG Y, SUN F, ZHANG Y, et al. Central difference particle filter applied to transfer alignment for SINS on missiles [J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2012, 48(1): 375 – 387.
- [9] TANG Lijun. *Cubature Kalman filter and application research on navigation* [D]. Harbin: Harbin Engineering University, 2012. (唐李军. Cubature卡尔曼滤波及其在导航中的应用研究 [D]. 哈尔滨: 哈尔滨工程大学, 2012.)
- [10] KOTTECHA J H, DJURIC P M. Gaussian particle filter [J]. *IEEE Transaction on Signal Processing*, 2003, 51(10): 2592 – 2601.
- [11] KOTTECHA J H, DJURIC P M. Gaussian sum particle filtering [J]. *IEEE Transaction on Signal Processing*, 2003, 51(10): 2602 – 2612.
- [12] HUBER M F, HANEBECK U D. Gaussian filtering for polynomial systems based on moment homotopy [C] // *Proceedings of the 16th International Conference on Information Fusion*. Istanbul: IEEE, 2013, 7: 1080 – 1087.
- [13] ZANETI R. Recursive update filtering for nonlinear estimation [J]. *IEEE Transaction on Automatic Control*, 2012, 57(6): 1481 – 1490.
- [14] ZHANG Yonggang, HUANG Yulong, WU Zhemin, et al. A high order unscented Kalman filtering method [J]. *Acta Automatica Sinica*, 2014, 40(5): 838 – 848. (张勇刚, 黄玉龙, 武哲民, 等. 一种高阶无迹卡尔曼滤波方法 [J]. 自动化学报, 2014, 40(5): 838 – 848.)
- [15] SIBLEY G, SUKHATME G, MATTHIES L. The iterated sigma point Kalman filter with applications to long range stereo [C] // *Proceedings of Robotics: Science and Systems*. Philadelphia: MIT Press, 2006.
- [16] ZHANG Zhaoyou, HAO Yanling, WU Xu. Complexity analysis of three deterministic sampling nonlinear filtering algorithms [J]. *Journal of Harbin Institute of Technology*, 2013, 45(12): 111 – 115. (张召友, 郝燕玲, 吴旭. 3种确定性采样非线性滤波算法的复杂度分析 [J]. 哈尔滨工业大学学报, 2013, 45(12): 111 – 115.)
- [17] HUANG Yulong, ZHANG Yonggang, LI Ning, et al. Particle filter with uncertain measurement and unknown false alarm probability [J]. *Control Theory & Applications*, 2015, 32(8): 1012 – 1022. (黄玉龙, 张勇刚, 李宁, 等. 带不确定量测和未知虚警概率的粒子滤波器 [J]. 控制理论与应用, 2015, 32(8): 1012 – 1022.)

## 作者简介:

张勇刚 (1981-), 男, 教授, 博士生导师, 主要研究方向为光纤陀螺、惯性导航、滤波算法和组合导航, E-mail: zhangyg@hrbeu.edu.cn;

王刚 (1991-), 男, 硕士研究生, 主要研究方向为非线性滤波算法、惯性导航和组合导航, E-mail: wanggang2016@126.com;

黄玉龙 (1991-), 男, 博士研究生, 主要研究方向为非线性滤波算法、惯性导航和组合导航, E-mail: heuedu@163.com;

李宁 (1980-), 女, 副教授, 主要研究方向为自适应滤波和组合导航, E-mail: ningli@hrbeu.edu.cn.