FEIXIANXING XITONG DE
ZHUANGTAI GUJI FANGFA

非线性系统的状态估计方法

白晓波 编著

西南交通大学出版社

非线性系统的状态估计方法

白晓波　编著

西南交通大学出版社
·成　都·

图书在版编目（CIP）数据

非线性系统的状态估计方法／白晓波编著．—成都：西南交通大学出版社，2018.12
ISBN 978-7-5643-6638-4

Ⅰ．①非… Ⅱ．①白… Ⅲ．①非线性系统（自动化）－研究 Ⅳ．①TP271

中国版本图书馆 CIP 数据核字（2018）第 275571 号

非线性系统的状态估计方法 | 白晓波 编著 | 责任编辑 张宝华
封面设计 何东琳设计工作室

印张 14 字数 265千
成品尺寸 170 mm × 230 mm
版次 2018年12月第1版
印次 2018年12月第1次
印刷 四川森林印务有限责任公司
书号 ISBN 978-7-5643-6638-4

出版发行 西南交通大学出版社
网址 http://www.xnjdcbs.com
地址 四川省成都市二环路北一段111号
西南交通大学创新大厦21楼
邮政编码 610031
发行部电话 028-87600564 028-87600533
定价 56.00元

前　言

非线性系统的状态估计一直受到国内外研究人员的广泛关注，具有重要的理论意义及使用价值。对于解决非线性状态估计问题，具有代表性的算法是粒子滤波（Particle Filter，PF）和扩展卡尔曼滤波（Extend Kalman Filter，EKF），它们在统计信号处理、经济学、生物统计学、通信、目标跟踪、故障诊断、卫星导航和声呐定位等领域均有广泛的应用前景。

但是，粒子滤波和扩展卡尔曼滤波在实际应用中需要解决一些影响滤波性能的问题，如在粒子滤波中，主要有粒子权值退化、粒子丧失多样性等问题；在扩展卡尔曼滤波中，主要需要解决的是系统参数和噪声数据不准确导致的估计误差较大等问题。因此，本书主要围绕这两条主线展开：一是粒子滤波的改进和优化，二是对 EKF 的改进和优化。

利用群体智能优化算法（Swarm Intelligence Optimization Algorithm，SIOA）对 PF 进行优化具有较强的代表性，为此，作者首先介绍 8 种常见的群体智能算法，如粒子群算法、鸡群算法和蚁群算法等。而较多的学者都是基于粒子群算法对 PF 进行优化的，其基本思路也为其他研究人员提供了重要的借鉴意义；而基于其他智能优化算法对 PF 进行优化也得以快速发展，如先对萤火虫算法、蝙蝠算法和差分进化算法等进行改进，再对 PF 进行优化。在分析和总结众多学者的研究思路的基础上，作者重点研究了利用布谷鸟算法和烟花算法对 PF 进行改进和优化的方法。其基本思想是：由于蝙蝠算法和萤火虫算法易陷入局部最优问题，而烟花算法具有很好的随机性和全局收敛性，因此，基于这些算法对 PF 进行改进和优化而提出了 FWA-PF。然后重点分析了 FWA-PF 的收敛性，以及烟花爆炸半径、火花数对粒子多样性及 PF 性能的影响。另外，作者也探索了利用布谷鸟算法对 PF 进行优化的方法，以及基于多新息理论对 PF 进行优化的方法。在群体智能优化算法优化粒子滤波的基础上，本书最后一章也介绍了其他学者基于其他方法和思想对 PF 的改进，这也为我们的研究提供了重要的借鉴作用。

EKF 作为另一个解决非线性系统状态估计问题的标准方法，作者首先介绍了 EKF 的基本原理和方法，并重点介绍了基于多新息理论优化 EKF，以及其他如基于雁群 PSO、模糊神经网络等对 EKF 优化的方法。最后，介绍了作者本人的基于核偏最小二乘法（Kernel Partial Least Square, KPLS）对 EKF 进行优化的方法。

鉴于作者的研究成果是在其他学者的已有研究成果基础上进行的，这里首先对粒子滤波和扩展卡尔曼滤波的改进与优化做出研究的学者深表感谢。在成书的过程中，西安工程大学管理学院邵景峰教授给予了必要的指导，硕士研究生王蕊超同学为我提供了全力支持，在此表示特别感谢。

非线性状态估计在科学研究和工程应用领域具有重要价值，如工业过程中的状态反馈控制、航空制导系统、飞行目标跟踪、故障诊断和生化反应状态提取等领域。然而，由于作者的研究成果和工程积累尚不十分丰富，书中难免有不足之处，敬请读者不吝赐教。期望此书能起到抛砖引玉的作用，能为我国的科学研究和工程应用贡献微薄之力。

白晓波

2018 年于西安工程大学

目　录

第 1 章　研究背景

Kalman 于 1960 年提出了针对线性高斯问题领域的最优化估计理论，即卡尔曼滤波器（Kalman Filter，KF），它广泛应用于众多行业[1]。此后，人们对其进行了众多的应用研究，如张智勇[2]等提出了改进卡尔曼滤波短时客流预测模型；谢朔[3]和高向东[4]基于多新息理论提出了改进卡尔曼滤波，并分别将改进算法应用于辨识船舶响应模型和焊缝在线识别；文献[5]中的无人机分析控制系统故障检测；其他如文献[6-8]是基于多新息理论的改进 KF 或者扩展卡尔曼滤波（Extended Kalman Filter，EKF）[9]的研究。

然而，卡尔曼滤波器的重点在于对线性高斯问题的求解，而实际问题并不符合这种理想状态，即更多的实际系统的状态方程和量测方程是非线性非高斯的，即量测值带有噪声[10]，且噪声不一定服从高斯分布，这就使得 KF 在应用于这样的系统时状态估计误差较大。因此，针对这类非线性问题，人们又在卡尔曼滤波器的基础上提出了扩展卡尔曼滤波器。其核心思想是：通过 Taylor 展开式，对非线性系统局部线性化，这给弱非线性系统带来了较好的滤波效果。但是，对于强非线性系统，EKF 只能逼近到一阶精度，并不能充分体现系统的强非线性特性，甚至导致系统发散。于是，人们又提出了基于 EKF 和无极变换的另外一种改进的解决非线性问题的滤波器，即无极卡尔曼滤波器（Unscented Kalman Filter，UKF）[11]。其基本思想是利用高斯分布来近似任何一种非线性方程。利用 UKF 能够取得三阶矩的后验均值与协方差估计。但是，正是由于它是对非线性系统的后验概率密度的高斯假设，才使得它对非高斯分布模型的适用性不好。

1.1　粒子滤波理论

从线性高斯问题到非线性非高斯问题，产生的另外一种著名方法就是粒子滤波（Particle Filter，PF），也称为序贯蒙特卡罗方法（Sequential Monte Carlo

Methods，SMC）。其基本思想是利用一组粒子近似表示系统在 k 时的后验概率分布，而非线性系统在 k 时的状态就用其近似分布表示。

1.1.1 非线性滤波问题描述

设描述织造过程的非线性动态系统的状态空间模型为：

$$x_k = f_k(x_{k-1}, v_{k-1}) \tag{1-1}$$

$$z_k = h_k(x_k, u_k) \tag{1-2}$$

其中，x_k 表示织造过程在 k 时刻所处的状态，z_k 表示织造过程在 k 时刻的观测向量，$f_k : A'^{n_x} \times A'^{n_x} \to A'^{n_x}$ 为织造状态转移函数，$h_k : A'^{n_x} \times A'^{n_x} \to A'^{n_x}$ 为织造系统量测函数；v_k、u_k 为织造过程噪声及观测噪声。假设织造过程为 m 阶马尔科夫模型，那么，织造数据的非线性滤波问题就转化为带有噪声的织造量测数据，递归估计的非线性织造过程的后验概率密度为 $p(x_{0:k} \mid z_{1:k})$，其中，$x_{0:k} = \{x_0, x_1, \cdots, x_k\}$ 为 k 时织造过程的状态序列，$z_{1:k} = \{z_1, z_2, \cdots, z_k\}$ 为 k 时织造过程的量测序列。文献[16]对 m 阶马尔科夫假设下的后验概率密度函数 $p(x_{0:k} \mid z_{1:k})$ 进行了推导，结论如下：

$$p(x_{0:k} \mid z_{1:k}) = p(z_k \mid x_k) p(x_k \mid x_{k-m:k-1}, z_{1:k-1}) p(x_{0:k-1} \mid z_{1:k-1}) \tag{1-3}$$

1.1.2 粒子滤波的基本原理

20 世纪 40 年代，Metropolis 等人提出了蒙特卡罗方法（Monte Carlo Method）[17]。其后，国内外很多学者对粒子滤波中的主要问题进行了深入研究，如粒子权值退化问题、粒子匮乏现象和自适应粒子数量选择策略等[16]。

粒子滤波基本原理是利用贝叶斯滤波理论来解决状态估计问题的，要计算粒子权重，要使用蒙特卡罗序列方法来确定状态的后验概率。其过程如下：

已知状态的初始概率密度函数为 $p(x_0 \mid z_0) = p(x_0)$，状态更新方程为：

$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \tag{1-4}$$

蒙特卡罗方法是解决上述贝叶斯估计问题的有效方法，其基本思想是：将积分运算转化为有效样本点的求和运算。通常使用的采样算法是：从已知容易采样且极度近似后验概率分布的重要性密度函数 $q(x_{0:k} \mid z_{1:k})$ 中采样，再

通过样本加权近似 $p(x_{0:k} \mid z_{1:k})$，然后进行重采样可得 N 个随机样本点 $\{x_{k-1}^i\}_{i=1}^N$，则其概率密度函数为：

$$p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N} \omega_k^i \delta(x_k - x_k^i) \tag{1-5}$$

式中

$$\omega_k^i = \omega_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} \tag{1-6}$$

然而，在实际应用时存在抽取有效样本困难的问题，为此，引入了重要性采样方法，文献[16]中对此进行了详细阐述。主要计算公式如下：

$$\omega_k^i = \omega_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-m:k-1}^i)}{q(x_k^i \mid x_{k-m:k-1}, z_k)} \tag{1-7}$$

式中 ω_k^i 为 k 时刻、第 i 个粒子的权值。在粒子滤波过程中，主要计算量在于重新采样，以防粒子权重退化，因此，其时间复杂度为 $O(N^2)$，N 为采样时粒子数。

基于以上内容，m 阶粒子滤波表示如下：

For $k = 0,1,2,\cdots$,

（1）根据重要性采样密度函数初始化 N 个粒子 $x_k^i \sim q(x_k \mid x_{k-m:k-1}^i, z_k)$，得粒子集合 $x_{0:k}^i = (x_{0:k-1}^i, x_k^i), i = 1,\cdots,N$。

（2）利用公式（1-7）计算 N 个粒子权值。

（3）粒子权值归一化：

$$\omega_k^i = \frac{\omega_k^i}{\sum_j \omega_k^j}, i = 1,\cdots,N \tag{1-8}$$

（4）$k = k + 1$，返回步骤（1）。

end For

1.1.3　标准粒子滤波 Matlab 程序实例

通过 Matlab 源程序，以更加直观的方式说明 1.1.2 节中的粒子滤波原理，下面给出状态方程和量测方程：

$$x=\frac{1}{2}x+20\frac{x}{1+x^2}+8(\cos(1.5(k-1))^2)+sqrt(Q)\times rand\ n \tag{1-9}$$

$$y=\frac{x^2}{20}+sqrt(R)\times rand\ n \tag{1-10}$$

其中 $Q=10$ 为系统过程噪声方差，$R=10$ 为量测噪声方差，rand n 为正态分布的随机数。基于式（1-9）和（1-10），下面给出粒子滤波仿真程序：

```
x=5; % initial state 初始状态
Q=10; % process noise covariance 过程噪声方差
R=10; % measurement noise covariance 测量噪声方差
tf=200; % simulation length 模拟长度
N=100; % number of particles in the particle filter 粒子过滤器中的粒子数
xhat=x; %xhat=x=0.1
P=2;
xhatPart=x; %xhatPart=x=0.1
% initialize the particle filter
   初始化粒子滤波，xpart 值用来在不同时刻生成粒子
t1=clock;
for i=1 : N
   xpart(i)=x + sqrt(P)*rand n;
   % rand n 产生标准正态分布的随机数或矩阵的函数
end %初始化 xpart(i)为生成的 100 个随机粒子
xArr=[x]; %xArr=x=0.1
xhatPartArr=[xhatPart]; %xhatPartArr=[xhatPart]=0.1
close all;
%粒子分布
particles=zeros(1,N);
estValue=0;
for k=1 : tf  %tf 为时间长度，k 可以理解为时间轴上的 k 时刻
  % system simulation 系统仿真
  % x 数据为时刻 k 的真实状态值
  x=1/2*x + 20*x / (1 + x^2) + 8*(cos(1.5*(k-1))^2) + sqrt(Q)*rand n;
  %状态方程(1)
  y=x^2 / 20 + sqrt(R)*rand n; %观测方程(2)
```

观测方程是在观测值和待估参数之间建立的函数关系式。

```
statusArr(k)=x;
observeArr(k)=y;
% particle filter
  生成 100 个粒子，并根据预测值和观测值的差值计算各个粒子的权重
for i=1 : N
  xpartminus(i)=1/2*xpart(i) + 20*xpart(i)/(1 + xpart(i)^2) + 8*(cos(1.5*(k-1))^2) +
              sqrt(Q) *rand n;
  ypart=xpartminus(i)^2 / 20;
  vhat=y-ypart; %观测值和预测值的差
  q(i)=(1 / sqrt(R) / sqrt(2*pi)) * exp(-vhat^2 / 2 / R);
  %根据差值给出 100 个粒子对应的权重
end
% normalize the likelihood of each a priori estimate
  每一个先验估计正常化的可能性
qsum=sum(q);
for i=1 : N
  q(i)=q(i) / qsum; %归一化权重，较大的权重除以 qsum 后不为零，
                  大部分较小的权重除以 qsum 后为零
end
% resample 重新取样
for i=1 : N
  u=rand; % uniform random number between 0 and 1
            0 和 1 之间的均匀随机数
  qtempsum=0;
  for j=1 : N
    qtempsum=qtempsum + q(j);
    if qtempsum >=u  %重采样对低权重进行剔除，同时保留高权重，
                  防止退化
      xpart(i)=xpartminus(j);
      break;
    end
  end
```

```
    end
    % the particle filter estimate is the mean of the particles
      粒子滤波的估计是颗粒的平均值
    xhatPart=mean(xpart); %经过粒子滤波处理后的均值
    if k==100
        particles=xpart;
        estValue=xhatPart;
    end
    % plot the estimated pdf's at a specific time
      绘制在特定的时间估计的概率密度函数
    if k==20 % particle filter pdf
      pdf=zeros(81,1);
      for m=-40 : 40
        for i=1 : N
          if (m <=xpart(i)) && (xpart(i) < m + 1)
         %pdf 为概率密度函数，这里是 xpart(i)值落在[m, m + 1)上的次数
            pdf(m + 41)=pdf(m + 41) + 1;
          end
        end
      end
      figure;
      m=-40 : 40;
      %此图 1 绘制 k=20 时刻 xpart(i)区间分布密度
      plot(m,pdf/N,'r');
      hold;
      title('Estimated pdf at k=20');
      disp(['min,max xpart(i) at k=20 : ',num2str(min(xpart)),',',num2str (max(xpart))]);
    end
    % save data in arrays for later plotting
    xArr=[xArr x];
    xhatPartArr=[xhatPartArr xhatPart];
end
t2=clock;
t3=etime(t2,t1);
```

```
%xlswrite('outData_store.xls',[xArr',xhatPartArr']);
t=0:tf;
figure;
plot(t,xArr,'b.',t,xhatPartArr,'g'); % xArr 为真值，xhatPartArr 为粒子滤波值
xlabel('time step'); ylabel('state');
legend('True state','Particle filter estimate');
```

1.1.4　粒子滤波存在的问题

PF 的粒子采样存在两个主要问题：一是粒子权值退化。为了解决该问题，Gordon[12]等人引入重采样（Resampling）方法；二是粒子贫化，丧失多样性。该问题又是由重采样导致的。因为重采样方法是复制权值较大的粒子以留作下一次迭代使用，然而在舍弃权值较小的粒子时，会导致部分权值较大的粒子可能被多次采样，这就导致粒子丧失了多样性。因此，为了保证算法的有效性，需要有足够多的粒子数，但粒子过多会增加运算量，影响算法实时性。为此，针对粒子滤波的粒子权值退化和丧失多样性问题，很多学者进行了研究。

1.1.5　重要性采样密度函数设计

在粒子滤波中，主要的步骤在于粒子的重要性采样。文献[13]指出，最优的重要性采样密度函数为 $p(x_k \mid x_{0:k-1}, z_{1:k})$，以其作为重要性采样密度函数可使得粒子权值的方差最小，进而降低权值退化的影响。然而，实现最优的重要性采样函数非常困难。为此，研究人员设计了次优的重要性采样密度函数构造方法，主要有三类[10]：一是利用非线性的滤波器 EKF 或 UKF 来构造重要性采样密度函数；二是 Auxiliary 粒子滤波（APF）[14]；三是利用优化算法发现非规范的最优采样密度模式，并拟合分散的学生 t-分布(Student's t-distribution)，进而模仿最优采样。但该方法需为每个粒子进行优化求解，算法的时间复杂度很高。

1.2　小　结

这一章重点阐述了非线性系统的研究背景，以及一个主要的非线性系统状态估计的方法——粒子滤波，以便为后续章节理论的阐述起到引领作用。

参考文献

[1] KIM P，HUH L. Kalman filter for beginners：with matlab examples[M]. Georgia：Create Space Independent Publishing Plat，2011.

[2] 张智勇，张丹丹，贾建林，等. 基于改进卡尔曼滤波的轨道交通站台短时客流预测[J]. 武汉理工大学学报:交通科学与工程版，2017，41(06)：974-977.

[3] 谢朔，陈德山，初秀民，等. 改进多新息卡尔曼滤波法辨识船舶响应模型[J]. 哈尔滨工程大学学报，2018，39(02)：282-289.

[4] 高向东，许二娟，李秀忠. 多新息理论优化卡尔曼滤波焊缝在线识别[J]. 焊接学报，2017，38(03)：1-4＋129.

[5] 刘晓东，钟麦英，柳海. 基于 EKF 的无人机飞行控制系统故障检测[J]. 上海交通大学学报，2015，49(6)：884-888.

[6] 刘毛毛，秦品乐，吕国宏，等. 基于多新息理论的 EKF 改进算法[J]. 计算机应用研究，2015，32(05)：1568-1571.

[7] 刘毛毛，吕国宏，常江. 基于多新息理论的卡尔曼滤波改进算法[J]. 中北大学学报(自然科学版)，2015，36(02)：234-239.

[8] 吕国宏，秦品乐，苗启广，等. 基于多新息理论的 EKF 算法研究[J]. 小型微型计算机系统，2016，37(03)：576-580.

[9] DOUCET A，JOHANSEN A M. A tutorial on particle filtering and smoothing：Fifteen years later//CRISAN D，ROZOVSKII B，eds. The Oxford Handbook of Nonliner Filtering. New York：Oxford University Press，2011：656-704.

[10] 王法胜，鲁明羽，赵清杰，等. 粒子滤波算法[J]. 计算机学报，2014，37(08)：1679-1694.

[11] 王璐，李光春，乔相伟，等. 基于极大似然准则和最大期望算法的自适应 UKF 算法[J]. 自动化学报，2012，38(07)：1200-1210.

[12] GORDON N J，SALMOND D J，SMITH A F M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation [J]. IEE Proceedings F：Radar and Signal Processing，1993，140(2)：107-113.

[13] CORNEBISE J，MOULINES É，OLSSON J. Adaptive methods for sequential importance sampling with application to state space models[J]. Statistics & Computing，2013，18(4)：461-480.

[14] JOHANSEN A M，DOUCET A. A note on auxiliary particle filters[J]. Statistics & Probability Letters，2008，78(12)：1498-1504.

[15] 胡士强，敬忠良. 粒子滤波算法综述[J]. 控制与决策，2005(04)：361-365+371.

[16] 李孟敏. 粒子滤波算法综述[J]. 中国新通信，2015，17(10):5.

[17] Chan V. Theory and applications of Monte Carlo simulations. Rijeka, Croatia: InTech,2013.

第 2 章　群体智能算法

自然界中大多数动物往往以群居的方式生活及生存，如狼群、羊群、鸟群、鱼群、蜂群等群体。其原因主要是群体中单个成员的生存能力有限，而整个群体却能够表现出强大的战斗力和生命力，这种战斗力和生命力不仅表现在个体能力的单方面叠加，还存在着各种信息的传递和交换，而个体依据接收到的信息可对自己的行为进行调整，最终体现出群体智能。群体智能（Swarm Intelligence，SI）是一类具有自组织行为智能群体的总称，即基于个体群成员的聚集，也表现出独立的智能。1989 年，Gerardo Beni 和 Jing Wang 在文章《*Swarm Intelligence*》中第一次提出了“群体智能”概念[1]。SI 可以表达为由个体内部之间、个体与内外部环境之间的相互作用最终实现的智能信息交互行为，而且群体中的个体都遵循简单的行为规则，同时，群体之间没有统一的控制中心，个体之间的相互作用最终表现为整个种群上的智能[2]。SI 的优点在于[3]：

（1）灵活性：整个种群能够快速适应变化的环境。

（2）鲁棒性：即使少数个体无法工作，整个种群依然能够正常运转。

（3）自组织性：整个种群只需要相对较少的监督或自上而下的控制。

基于此，近年来国内外学者通过观察、研究、模拟动物的群体信息交互模式，提出许多新型的群体智能算法，这些算法主要表现为算法参数较少、进化过程相对简单、运算速度快、全局搜索能力较强，它适用于解决高维和多目标优化问题。对此，本章将选取部分经典群体智能算法，如粒子群优化算法、萤火虫算法、布谷鸟算法、鸡群算法、蝙蝠算法、烟花算法、差分进化算法、混合蛙跳算法、蚁群算法等进行阐述及介绍。

2.1　粒子群优化算法

2.1.1　产生背景

优化问题是在工业生产和设计过程中经常遇到的现实问题，许多问题最