最优滤波器在机器人定位中的应用

余丛杉，郑撼，张含蕾

摘 要：随机信号分析与处理是信号分析与处理中的重要内容。本文先主要介绍了随机信号的一些基础知识，后介绍了两种处理随机信号的方法：卡尔曼滤波和粒子滤波。卡尔曼滤波是一种利用线性系统状态方程，通过融合上一时刻估计值和当前时刻测量值来得到当前时刻的最优估计的方法，能够有效滤去信号中的白噪声。而扩展卡尔曼滤波破除了卡尔曼滤波只能应用在线性系统中的局限性，将非线性系统线性化，从而得到非线性系统中的最优估计。由于其非线性的特点，扩展卡尔曼滤波在广泛应用于各个领域各个场景。本文从扩展卡尔曼滤波出发，讨论并仿真模拟了扩展卡尔曼滤波在机器人定位中的应用。粒子滤波是通过非参数化的蒙特卡洛模拟方法来实现递推贝叶斯滤波，适用于任何能用状态空间模型描述的非线性系统，精度可以逼近最优估计，广泛应用于目标跟踪、信号处理以及自动控制等领域。从贝叶斯滤波理论出发，探讨普遍适用性的粒子滤波器，最后介绍当前粒子滤波在机器人定位中应用中的算法原理。

关键词：最优滤波器；定位；卡尔曼滤波；粒子滤波

**Abstract:**Random signal analysis and processing is an important part of signal analysis and processing. This article first mainly introduces some basic knowledge of random signals, and then introduces two methods of processing random signals: Kalman filter and particle filter. Kalman filter is a method that uses the linear system state equation to obtain the best estimate of the current time by fusing the estimated value at the previous time and the measured value at the current time, which can effectively filter the white noise in the signal. The extended Kalman filter eliminates the limitation that Kalman filter can only be applied to linear systems, linearizes the nonlinear system, and obtains the optimal estimation in the nonlinear system. Due to its non-linear characteristics, the extended Kalman filter is widely used in various fields and various scenarios. This article starts from the extended Kalman filter, discusses and simulates the application of the extended Kalman filter in robot positioning. Particle filtering is a non-parametric Monte Carlo simulation method to achieve recursive Bayesian filtering. It is suitable for any nonlinear system that can be described by a state space model. The accuracy can be approximated to the optimal estimation. It is widely used in target tracking and signal Processing and automatic control and other fields. Starting from Bayesian filtering theory, this paper discusses the universal applicability of particle filters, and finally introduces the current algorithm principles in the application of particle filters in robot positioning.

**Key words:** optimal filter; positioning; Kalman filter; particle filter

目录

[一、 随机信号分析 4](#_Toc75868945)

[二、 卡尔曼滤波（KF） 5](#_Toc75868946)

[2.1卡尔曼滤波(KF) 5](#_Toc75868947)

[2.2扩展卡尔曼滤波（EKF） 6](#_Toc75868948)

[2.3随机信号估计效果 7](#_Toc75868949)

[2.4扩展卡尔曼滤波在机器人定位中的应用 9](#_Toc75868950)

[三、 贝叶斯滤波原理 11](#_Toc75868951)

[3.1贝叶斯滤波原理 11](#_Toc75868952)

[3.2 粒子滤波 12](#_Toc75868953)

[3.2.1 贝叶斯重要性采样 12](#_Toc75868954)

[3.2.2重采样 13](#_Toc75868955)

[3.2.3粒子滤波定位算法 13](#_Toc75868956)

[3.3随机信号估计效果 13](#_Toc75868957)

[3.4粒子滤波在机器人定位中的应用 14](#_Toc75868958)

[3.4.1初始化 14](#_Toc75868959)

[3.4.2预测 16](#_Toc75868960)

[3.4.3更新粒子权重 17](#_Toc75868961)

[3.4.4坐标系转换 17](#_Toc75868962)

[3.4.5重采样 17](#_Toc75868963)

[附录1 MATLAB程序代码 18](#_Toc75868964)

[1.1 EKF滤波的应用 18](#_Toc75868965)

[1.2 SIR粒子滤波的应用 21](#_Toc75868966)

[参考文献 25](#_Toc75868967)

# 随机信号分析

信号有确定信号和随机信号之分，两者的根本区分点在于随机信号的规律性只有在大量样本的基础上经过统计分析后才能呈现出来。也因此随机信号有不同于确定信号的独特的描述与分析方法。

首先，由于随机信号的不确定重复性，对于随机信号的描述需要建立在大量样本的基础上，而我们用“样本空间”或“集合”来描述这些大量样本，用符号X(t)来进行表示“样本空间”，用x(t)来表示“样本空间”中的一个确定信号。其次，我们用于描述随机信号的工具是随机信号的概率结构，如下所示：

1. n维联合概率分布函数

1. n维联合概率密度函数

为了分析随机信号，我们规定了随机信号的一些数字特征。在了解这些数字特征之前，我们需要先了解两个概念：平稳过程与非平稳过程。平稳过程是一种重要的随机过程，随机信号在平稳过程中主要的统计特性不会随时间推移而改变。但在非平稳过程中，随机信号的统计特性是会随时间推移而改变的。

而平稳随机信号的数字特征，我们在本文中主要用到的是均值与均方值：

1. 均值

对于连续取值的随机信号，

1. 均方值

对于非平稳信号，表现在均值、方差、相关函数等统计特征是与时间有关的，这时各态遍历性和统计特性的时间平均已失去意义。这时候我们有两种方法来解决这个问题：第一种方法，将信号参数调整的短暂过程视为平稳过程的近似方法；第二种方法，从本质上分析非平稳随机信号。本文中的卡尔曼滤波是采用了第一种方法，粒子滤波则采用了第二种从本质上分析非平稳信号的方法。

# 卡尔曼滤波（KF）

## 2.1卡尔曼滤波(KF)

卡尔曼滤波（Kalman filtering）是一种利用线性系统状态方程，通过系统输入输出观测数据，对系统状态进行最优估计的算法，主要应用于估计非平稳随机信号。卡尔曼滤波采用递推算法，利用前一时刻的估计值和当前时刻的新的观测值，实现自动跟踪随机信号统计特性的非平稳变化的功能，在估计的同时滤去了其他噪声信号，起到了对随机信号滤波的作用。

卡尔曼滤波最核心的思想体现在其对信号、观测值的表示和估计算法的递推表达上。下面我们给出其向量形式的表达式

**信号模型：**

其中，为n维信号向量；为n维服从高斯分布的零均值白噪声向量。为时序k-1到时序k的一步转移nn矩阵。

**观测模型：**

其中，为m维信号向量；为m维服从高斯分布的零均值白噪声向量。为mn的观测矩阵。

对于以这种方式估计的模型，其核心目标是使估计值和真实值的均方误差最小，即

根据两种模型的建立和估计判据的确立，卡尔曼滤波的估计过程由以下几个步骤确定:

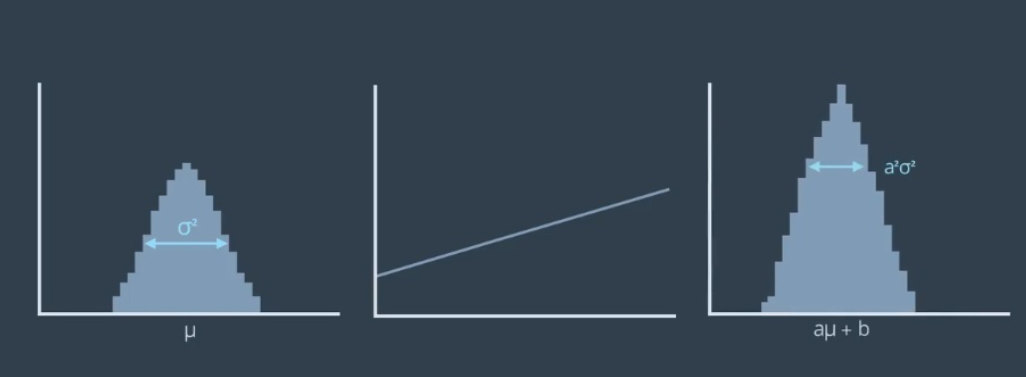
首先要计算预测值、预测值和真实值之间误差协方差矩阵，有了这两个就能计算卡尔曼增益矩阵B，再然后得到估计值.最后还要计算估计值和真实值之间的误差协方差矩阵，为下次递推做准备

设为k时刻的系统协方差矩阵，则有

其中和分别表示信号噪声和观测噪声的方差，当给出系统参数和估计的均方误差初值时，即可通过以上迭代得到滤去噪声的估计值，从而实现我们想要得到的功能。

## 2.2扩展卡尔曼滤波（EKF）

尽管卡尔曼滤波已经给我们提供了很强大的处理非平稳随机信号的滤波手段，但是在更多的滤波工程应用场景中，我们处理的系统都是非线性的。在利用卡尔曼滤波的场景中，如果系统是线性的，则能够应用，因为一个单峰的高斯分布在经过线性计计算后，单峰性并未发生改变。



但若系统是非线性的系统则卡尔曼滤波就无法应用于处理随机信号，以下图为例



一个满足单峰的高斯分布信号经过KF滤波后，如果系统为非线性话，滤波后的信号结构就明显不是单峰的分布了。因此，扩展卡尔曼滤波器的引入就显得十分必要。而对于控制机器人小车的位姿变换的控制方程来说，其转移矩阵就是非线性的。

EKF的基本思想是将非线性系统线性化，然后进行卡尔曼滤波，因此EKF是一种次优滤波。其后，多种二阶广义卡尔曼滤波方法的提出及应用进一步提高了卡尔曼滤波对非线性系统的估计性能。二阶滤波方法考虑了Taylor级数展开的二次项，因此减少了由于线性化所引起的估计误差，但大大增加了运算量，因此在实际中反而没有一阶EKF应用广泛。

在状态方程或测量方程为非线性时，我们需要模拟近似的从这个非线性算法中抽出线性的算法来近似计算。这样就可以让我们的卡尔曼滤波算法用于非线性环境。EKF对非线性函数的Taylor展开式进行一阶线性化截断，忽略其余高阶项，从而将非线性问题转化为线性，可以将卡尔曼线性滤波算法应用于非线性系统中。这样一来，解决了非线性问题。

根据这种思想，扩展卡尔曼滤波的递推关系式相较于卡尔曼滤波就需要进行一定的改动。在卡尔曼算法中我们注意到协方差是通过线性的F计算的，但是在EKF中不行，非线性的F会导致不是单峰的高斯分布，因此我们需要用近似的函数F进行的计算，此时有

此处f满足,表示此处的信号值

此时满足，由此扩展卡尔曼滤波模型的建立和估计的过程应修正为以下几个步骤

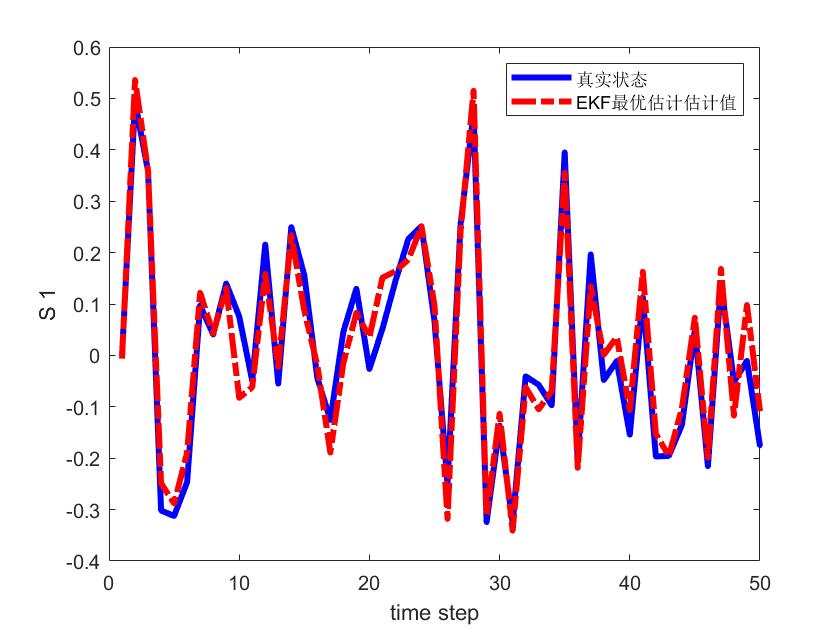
但是，尽管EKF应用于非线性状态估计系统中已经得到了学术界认可并为人广泛使用，这种方法也存在一定的缺陷，当Taylor展开式中被忽略的高阶项带来大的误差时，EKF算法可能会使滤波发散。

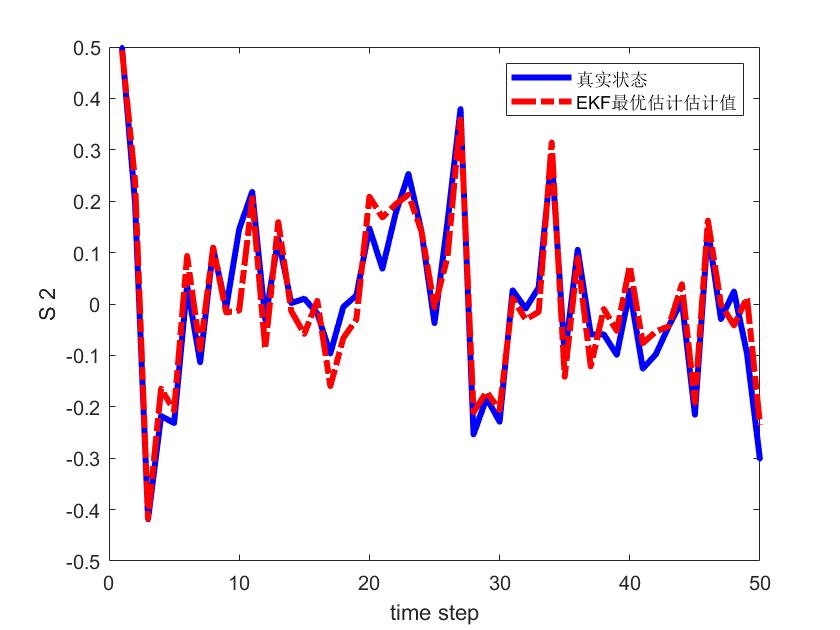
## 2.3随机信号估计效果

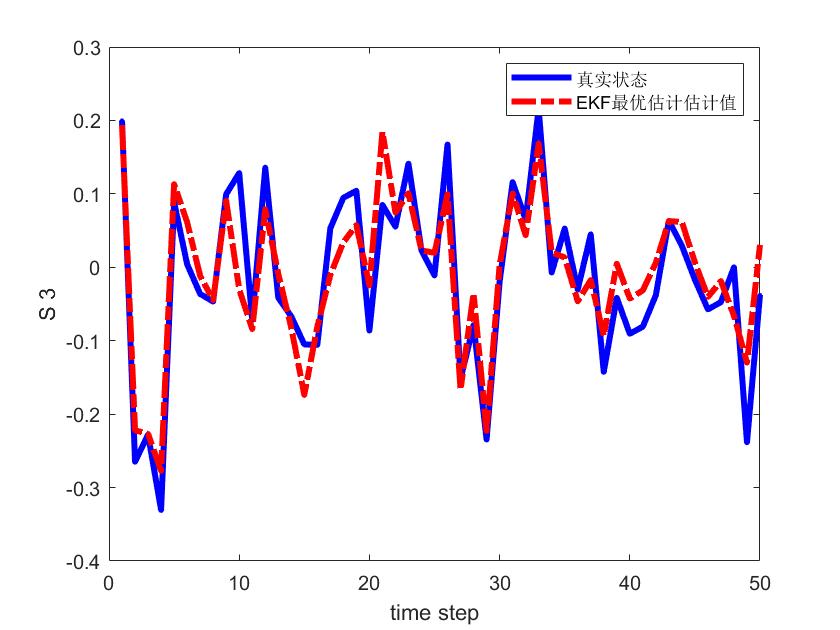
下面我们给出以下简单的非线性模型并利用matlab给出随机信号估计效果

我们假设信号满足

下列图表为利用matlab编程进行EKF迭代得到的初始估计值为的对系统随机信号估计的最优估计曲线







从这个例子中我们可以很好的看出，EKF在非线性随机信号估计中的优越性。

## 2.4扩展卡尔曼滤波在机器人定位中的应用

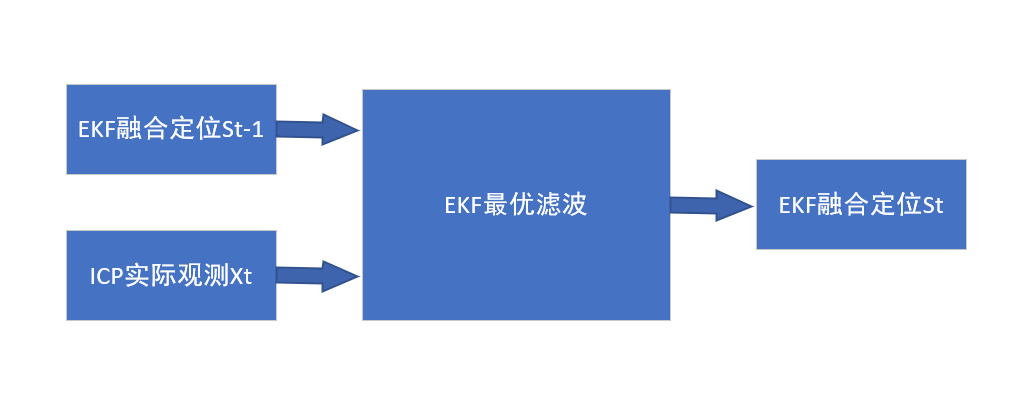
对于机器人的定位模型来说，核心思想是通过确定世界坐标系下的,从而来确定自己的位置。并且根据机器人在二维平面的运动学方程，很容易得到，机器人位姿的转移方程为

(t)

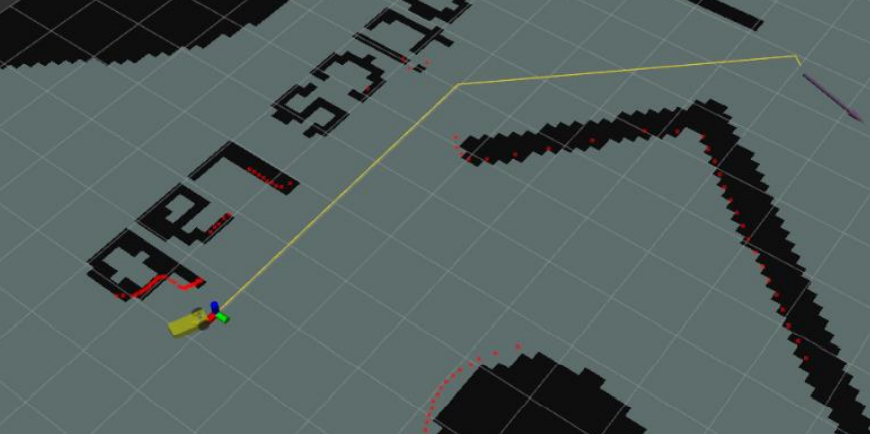
观测模型为

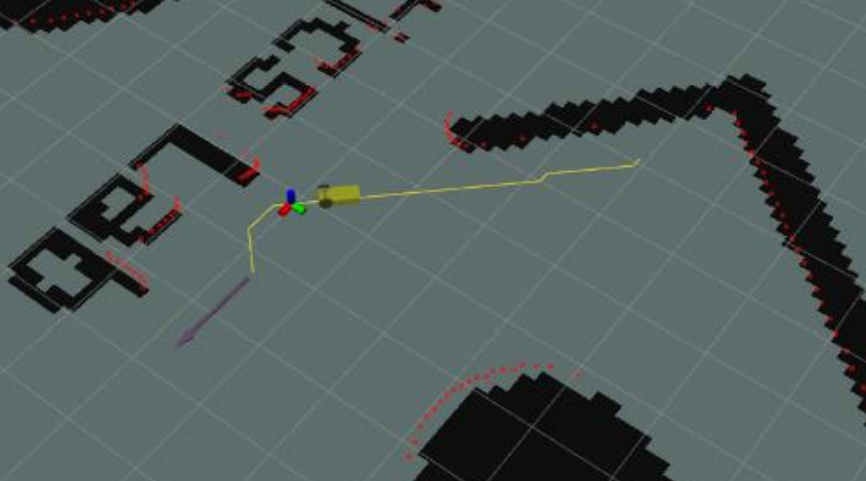
对此同样，我们进行EKF的迭代估计，即可通过当前机器人测得的值和上一时刻估计的位姿来融合估计得到最优的位姿估计值，从而进一步提高估计的精度并一定程度上解决机器人在运动过程中出现的位姿测量误差积累的问题，提高机器人定位的可靠性。

对此，我们基于ROS对机器人的定位估计进行了仿真，我们将机器人利用ICP得到的位姿视作观测得到的位姿，与上一时刻EKF迭代得到的最优估计值进行融合，得到了这一时刻的EKF最优估计值，从而提高了定位的准确性。



下面为在rviz仿真软件中关于实时定位的演示图：





在RVIZ中小车根据自己通过EKF和ICP融合得到的定位结合目标点的坐标根据相应的路径规划算法即可规划出合理的路线，根据路线不断下发速度，从而实现小车沿着规划好的路径行驶的功能。小车在行进过程中不断更新自己的位姿，从而动态的二维平面运动轨迹规划。

# 贝叶斯滤波原理

## 3.1贝叶斯滤波原理

贝叶斯滤波即借助已知的观测数据求计算后验概率，即用先前经验总结预测先验密度，进而求出系统动态转移密度，然后利用C-K方程进行预测，代入Bayesian更新公式，然后得出更新后的条件概率密度，重新代入系统状态转移密度。整个过程的核心为“预测-修正”的一个迭代过程。具体过程如下

（1）预测。由系统动态转移密度在未知k时刻观测值，得到先验密度p (xk-1|z1:k-1)到先验密度p (xk|z1:k-1)的预测。

若我们假定k-1时刻，p (xk-1|z1:k-1)已知，对于一阶Markov过程(即(k-1)时的概率只与(k-2)时有关)，验证下面的概率密度函数：

等号两端对(k-1)积分得C-K方程：

由此获取不含有k时刻观测值的先验密度,该先验密度可以通过系统的状态转移概率p (xk|xk-1)来计算。

(2)更新。即由系统观测模型,在获得k时刻的观测值zk后实现先验概率密度p (xk|z1:k-1)至=后验概率密度p (xk|z1:k)的推导。

获得观测值后，由Bayesian公式p(a|b) p(b)=p(a) p(b|a)有：

将观测量zk独立出来，有：

将上式代入得

由条件概率定义p(a,b)=p(a|b)p(b)有：

由联合分布概率公式有:p(a,b|c)=p(a|b,c)p(b|c)

又由贝叶斯公式有

假设各个的观测是相互独立的，则有：

代入可得：

## 3.2 粒子滤波

粒子滤波算法（Bootstrap算法）是一种源于Monte Carlo思想的递推贝叶斯滤波算法，即以某事件出现的频率来指代该事件的概率。核心为利用一系列大量采样的加权和表示后验概率密度，来近似积分运算，进而可以简便分析非线性动态系统。也称蒙特卡洛模拟方法或贝叶斯重要性采样。

### 3.2.1 贝叶斯重要性采样

该算法是从已知的、较方便采样的分布q (x k)中采样,通过对分布的采样粒子点进行加权来近似p (x0:k|z1:k)，简单来讲就是再定义f(xk)的数学期望。具体如下：

其中q(x0:k|z1:k)为重要性函数。引入为重要性采样权函数。

可写为：

借助q (x0:k|z1:k)得到的样本和与f (x0:k)相关量进行估计。前提是可以计算wk(x0:k)，进而推测q(x0:k|z1:k)，虽然分布q (x0:k|z1:k)未知，但能够借助下面式子E[f (x0:k)]近似表示：

按照下面的式子归一化权值：

代入可得：

对任意可积函数有：

在基础粒子滤波算法中，根据观测数据估量后验概率密度时，每当加入新另-一个的观测变量的值时都要再次对整个序列的粒子权值进行计算,序贯性重要采样方法采用了统计学中较为著名的序贯分析方法，首先进行粒子采样，在加入另一个的观测变量的值时生成新的权值和。这样就能够对后验概率进行一系列的递推估算来得到较为可靠的概率值。度和计算机计算存储资源的消耗有很大的影响，所以，网格参数的选取至关重要。综合考虑了对计算结果的精度要求和计算机的性能，该文最终采用默认网格化的大小,采用综合曲率划分网格，以提高网格化质量。

### 3.2.2重采样

但在序贯性重要采样中存在较为重要的较大的问题就是权值退化现象,重采样方法则可以有限减弱这一问题的影响，实质为减少或舍弃权值较小的粒子而复制较大的粒子，最后将粒子权值设置为合适值。

粒子滤波定位算法

### 3.2.3粒子滤波定位算法

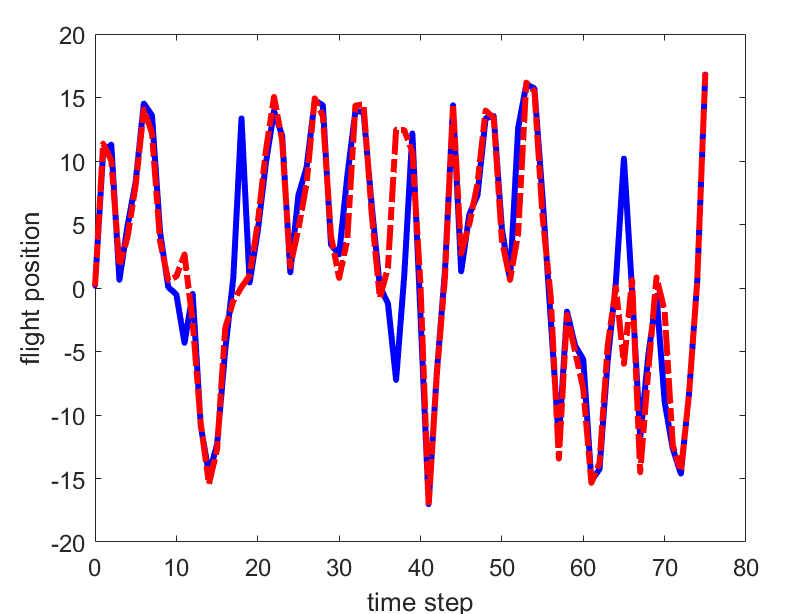
粒子滤波定位算法的核心即为上述粒子滤波原理，在机器人定位过程中，跟踪‘有利”的变量，即不同时间机器人的位置，初始时大量采样，接着在不同时刻进行贝叶斯滤波的“预测-更新’的迭代过程，然后利用重采样对粒子轨迹修正，最后得到机器人位置状况的最优状态估计。

## 3.3随机信号估计效果

下面我们给出以下简单的非线性模型并利用matlab给出随机信号估计效果

我们假设信号满足

下列图表为利用matlab编程进行EKF迭代得到的初始估计值为的对系统随机信号估计的最优估计曲线



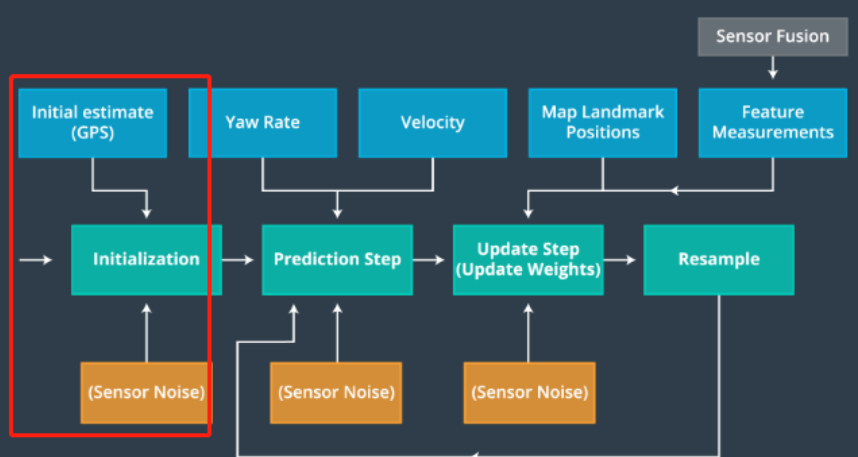
从这个例子中我们可以很好的看出，粒子滤波在非线性随机信号估计中的优越性。

## 3.4粒子滤波在机器人定位中的应用

粒子滤波主要分为四部分：初始化、预测、粒子权重更新、重采样，之后在重复的预测、更新、重采样，使得粒子逐渐向真实位置聚集。

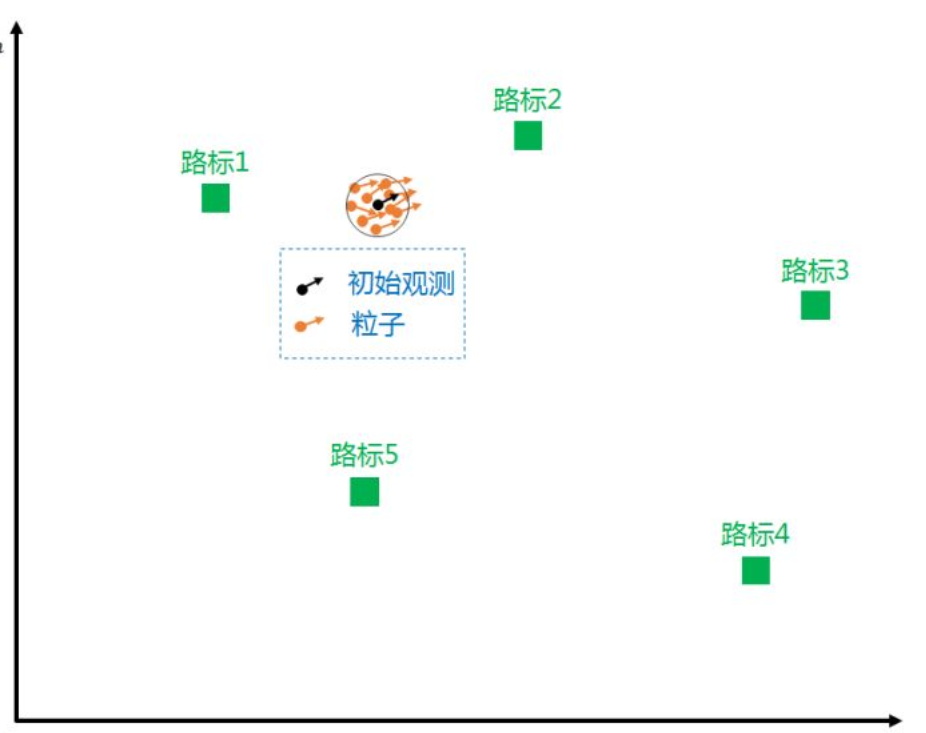
### 3.4.1初始化

粒子滤波初始化需要初始的位置（x,y），航向(yaw)，以及高斯噪声。

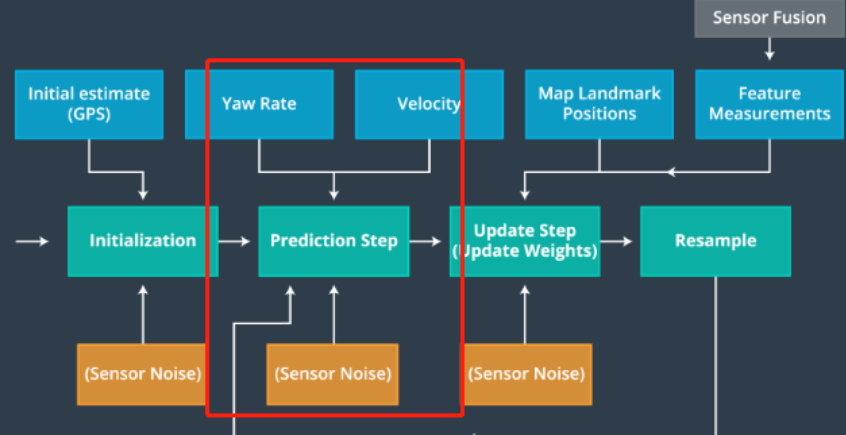


单纯用GPS的结果是有很大误差的，所以在初始化时，创造出N个粒子，并且所有粒子都是在GPS提供的初值附近的高斯分布。这是黑色圆圈出现的原因。每个粒子权重都相等，都是100%。

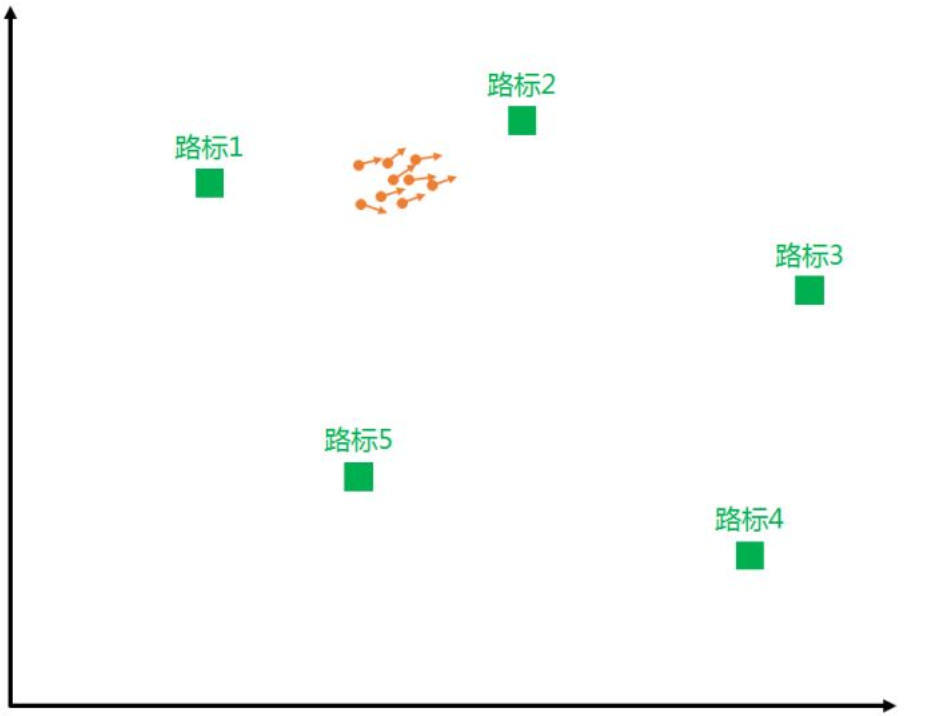
利用GPS的值初始化一定量粒子，粒子的位置（x, y）, 航向角（yaw满足正态分布。GPS测量值为均值，上面黄框中的传感器噪音，即为分布方差。



### 3.4.2预测



定位模块能够受到速度v和航向角变化率yaw\_rate，进而可以推测出运动△t时间后，自身所在的位置。由于每个粒子都可能是机器人的真实位置，当对所有粒子进行里程计预测（考虑运动噪音，也是满足正太分布），每个粒子都会有自己的位置和航向角。

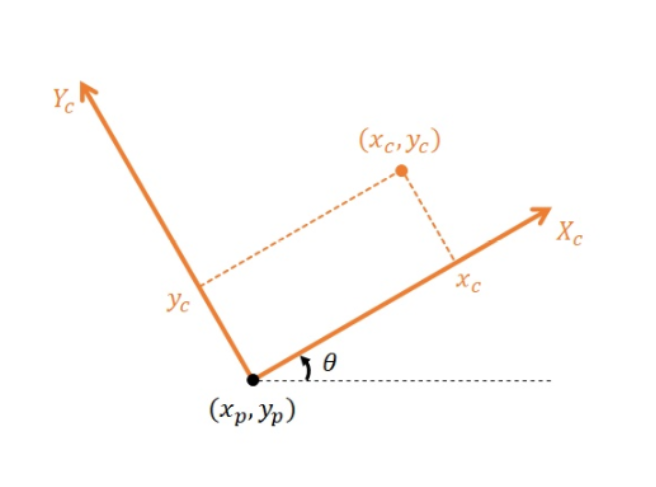


### 3.4.3更新粒子权重

这些粒子哪些才能更好的代表机器人的位置了，所以要给它们相应的权重，权重越大，它被重采样的机会也就越大，这样随着不但迭代，最终会找到最能代表机器人位置的粒子了。

### 3.4.4坐标系转换

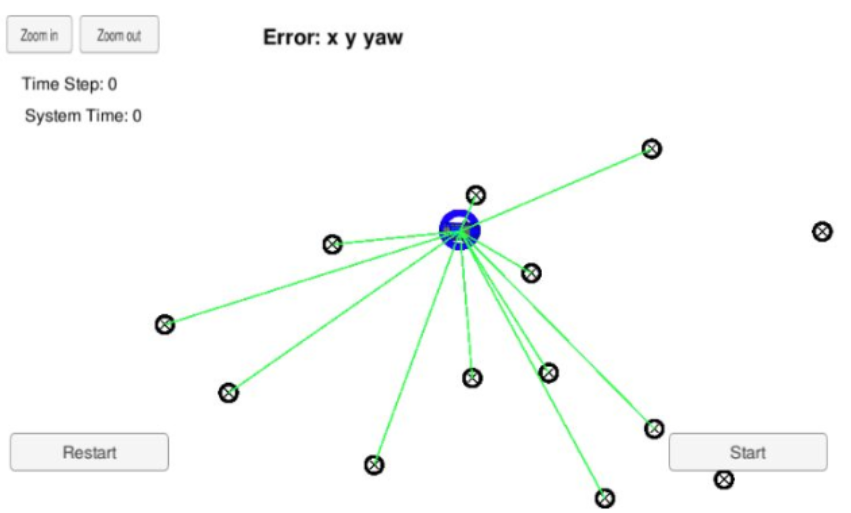
机器人传感器会获得周围路标的位置(比如a,b,c三个路标)，将机器人坐标下的位置(比如a,b,c)转化为全局地图上的位置（a’, b’ ,c’）



### 3.4.5重采样

重采样的目的是，保持粒子总数不变的情况下，删除掉那些权重较低的粒子，同时在权重较高的粒子附近抛洒新的粒子。对这些不同权重的粒子进行重新选取，权重大的粒子，代表的位置准确度更高，所以应该更多的重采样这些权重大的粒子。

随着重采样次数的增加，粒子的集中度会越来越高，留下来的都是极其接近真值的粒子。在这些粒子中筛选出权重最高的粒子，即为机器人的位置。



# 附录1 MATLAB程序代码

## 1.1 EKF滤波的应用

clear all   
close all   
clc   
% initialize the variables   
N = 50; % 计算连续N个时刻  
n = 3; % 状态维度  
q = 0.1; % 过程标准差  
r = 0.1; % 测量标准差  
% eye函数产生单位矩阵  
Q = q^2\*eye(n); % 过程方差  
R = r^2; % 测量值的方差  
  
f = @(x)[x(2);x(3);x(1)\*(x(2)+x(3))]; % 状态方程  
h = @(x)[x(1);x(2);x(3)]; % 测量方程  
s = [0;0.5;0.2]; % 初始状态  
  
% 初始化状态  
x = s+q\*randn(3,1);   
% eye返回单位矩阵  
P = eye(n);  
% 3行50列，一列代表一个数据  
xV = zeros(n,N); % 最优估计值  
% 真实值  
sV = zeros(n,N);  
% 状态测量值  
zV = zeros(n,N);  
  
for k = 1:N  
z = h(s) + r \* randn; % 实际状态  
sV(:,k) = s; % 状态测量值  
zV(:,k) = z;  
  
% 计算f的雅可比矩阵，其中x1对应黄金公式line2  
[x1,A] = jaccsd(f,x);  
% 过程方差预测，对应line3  
P = A\*P\*A'+Q; % 计算h的雅可比矩阵  
[z1,H] = jaccsd(h,x1); % inv返回逆矩阵  
K = P\*H'\*inv(H\*P\*H'+R);  
% 状态EKF估计值，对应line5  
x = x1+K\*(z-z1);  
% EKF方差，对应line6  
P = P-K\*H\*P;  
  
% save  
xV(:,k) = x;  
% update process   
s = f(s) + q\*randn(3,1);  
end  
  
for k = 1:3  
figure();   
clf   
plot(sV(k,:),'.-b','linewidth',3);  
hold on;  
plot(xV(k,:),'-.r','LineWidth',3);  
hold on;  
legend('真实状态', 'EKF最优估计估计值','状态测量值');  
xl = xlabel('time step');  
t = ['S ',num2str(k)] ;  
yl = ylabel(t);  
hold off;  
end

## SIR粒子滤波的应用

clear all   
close all   
clc   
%% initialize the variables   
x = 0.1; % initial actual state   
x\_N = 1; % 系统过程噪声的协方差 (由于是一维的，这里就是方差)   
x\_R = 1; % 测量的协方差   
T = 75; % 共进行75次   
N = 100; % 粒子数，越大效果越好，计算量也越大   
  
%initilize our initial, prior particle distribution as a gaussian around   
%the true initial value   
  
V = 2; %初始分布的方差  
x\_P = []; % 粒子   
% 用一个高斯分布随机的产生初始的粒子   
for i = 1:N   
x\_P(i) = x + sqrt(V) \* randn;   
end   
  
z\_out = [x^2 / 20 + sqrt(x\_R) \* randn]; %实际测量值   
x\_out = [x]; %the actual output vector for measurement values.   
x\_est = [x]; % time by time output of the particle filters estimate   
x\_est\_out = [x\_est]; % the vector of particle filter estimates.   
  
for t = 1:T   
x = 0.5\*x + 25\*x/(1 + x^2) + 8\*cos(1.2\*(t-1)) + sqrt(x\_N)\*randn;   
z = x^2/20 + sqrt(x\_R)\*randn;   
for i = 1:N   
%从先验p(x(k)|x(k-1))中采样   
x\_P\_update(i) = 0.5\*x\_P(i) + 25\*x\_P(i)/(1 + x\_P(i)^2) + 8\*cos(1.2\*(t-1)) + sqrt(x\_N)\*randn;   
%计算采样粒子的值，为后面根据似然去计算权重做铺垫   
z\_update(i) = x\_P\_update(i)^2/20;   
%对每个粒子计算其权重，这里假设量测噪声是高斯分布。所以 w = p(y|x)对应下面的计算公式   
P\_w(i) = (1/sqrt(2\*pi\*x\_R)) \* exp(-(z - z\_update(i))^2/(2\*x\_R));   
end   
% 归一化.   
P\_w = P\_w./sum(P\_w);   
  
for i = 1 : N   
x\_P(i) = x\_P\_update(find(rand <= cumsum(P\_w),1)); % 粒子权重大的将多得到后代   
end % find( ,1) 返回第一个 符合前面条件的数的 下标   
  
%状态估计，重采样以后，每个粒子的权重都变成了1/N   
x\_est = mean(x\_P);   
  
% Save data in arrays for later plotting   
x\_out = [x\_out x];   
z\_out = [z\_out z];   
x\_est\_out = [x\_est\_out x\_est];   
  
end   
  
t = 0:T;   
figure(1);   
clf   
plot(t, x\_out, '.-b', t, x\_est\_out, '-.r','linewidth',3);   
set(gca,'FontSize',12); set(gcf,'Color','White');   
xlabel('time step'); ylabel('flight position');

# 参考文献

[1]郝琰,张斌,李智熙.基于PI自适应卡尔曼滤波的空间轨迹校正算法[J].桂林电子科技大学学报,2021,41(02）：118-124.

[2]黄辉,邹安安,胡鹏,邹媛媛,蔡庆荣.基于Rao-Blackwellized粒子滤波器移动机器人SLAM研究[J].测控技术,2021,40(06):46-50.

[3]张园,苗晓婷,丛丹姝,王睿,刘儿兀.GNSS/INS组合导航拓展卡尔曼滤波和粒子滤波算法对比[J].系统仿真技术,2020,16(04):207-211.

[4]陈勇祺,林盛鑫,彭文翔,郭铭涛,黎兴欢,陈淑也.基于ROS的室内移动机器人研究[J].东莞理工学院学报,2021,28(03):33-39.