**机器人学期中报告（粒子滤波）**

张鸿、贾茗凯

**1 总述**

首先，我们补充了task中提到的五个函数，完成了基本的NP = 50的粒子滤波定位。然后，我们对主程序架构进行修改，以比较不同粒子数目情况下，自定位运行时间和误差大小等的差异，我们设置了四个模式，由程序开头的test\_mode 参数和if\_using\_prepared\_data参数管理。

**2 函数实现**

2.1 观测模型doObservation

1. function [z, xGnd, xOdom, u] = doObservation(xGnd, xOdom, u, landMarks, MAX\_RANGE)

以下为我们添加的d的计算公式,d为机器人本体观测到特征与自身位置的距离，这里采用的是欧氏距离，即2范数。其中，xGnd为x ground truth（仿真中机器人的实际运行位置），landMark(iz)为第i个特性的坐标信息。norm函数为matlab中计算二范数的函数。

1. d = norm(xGnd(1:2)' - landMarks(iz,1:2));

2.2 运动模型 doMotion

1. function [ x ] = doMotion( x, u)

本仿真模型中，机器人基本运动学模型为：

本模型采用v和w作为控制变量，所以模型如下：

可以表示成

因此，运动模型计算如下：

1. Delta = [ dt\*cos(x(3)) 0;
2. dt\*sin(x(3)) 0;
3. 0 dt];
4. x = x + Delta \* u;

2.3 高斯函数 Gaussian

function g = Gaussian(x,u,sigma)

本函数的目的是取得随机变量x在正态分布下的随机变量密度函数(pdf)取值。

我们采用matlab自带的normpdf调用高斯分布的密度函数，得到取值。函数内容如下

g = normpdf(x,u,sigma);

本函数在根据观测模型对立子重新赋权时使用，x为当前粒子预测观测状态和GND下的观测状态差值。

2.4 粒子的归一化：Normalization

重新赋权后的粒子权重虽然有比例重要性关系，但是权重之和不为一，不可以用于重采样，因此需要对立子进行归一化。归一化步骤如下。

function pw=Normalization(pw,NP)

pw = pw +1.e-300; %第一步，将权重加以1e-300，防止粒子权重由于计算原因变为0

pw = pw / sum(pw); % 归一化，将粒子权重除以权重和。

end

2.5 重采样： ResamplingStep

重采样步骤的目的是根据前面重新赋权的粒子权重，得到下一轮的粒子。

（1）重采样判断：为了避免 。。。 首先需要根据重新赋权后有效粒子个数来决定是否重采样。有效粒子数的计算公式为：

本仿真预先设计的阈值如下：

（2）重采样基本原理：重采样的目的是根据前面重新赋权的粒子权重，权重越大的粒子应该更多地被复制，而权重少的粒子应该更少地被复制。我们采用累加权重的方法，将粒子的权重分布映射为0到1上的权重带，若某个粒子权重越大，则他在权重带上的占用空间更大。

如下w\_cumu为累计权重。。得到累加权重后，我们采用rand函数得到0到1上均匀分布的随机数，如果w0 <= w\_cumu(j) && w0 > w\_cumu(j-1)，则证明得到的随机数分布在第j个粒子占用的权重带上，以此让粒子j得到复制。采用0-1均匀分布随机数的方法，既可以保证得到的粒子在统计规律上大致满足权重越大粒子复制越多的重采样，又使重采样带入了微弱的随机性，从而使模型更具鲁棒性。

函数如下：

function [px,pw]=ResamplingStep(px,pw,NTh,NP)

Neff=1/sum(pw.\*pw);

if Neff>NTh

w\_cumu=pw;

pxnew = px;

for i = 2 : NP

w\_cumu(i) = w\_cumu(i - 1) + w\_cumu(i);

end

w\_cumu(NP+1) = 1.001;

for i = 1 : NP

w0 = rand;

if w0 <= w\_cumu(1)

pxnew(:,i) = px(:,1);

continue;

end

for j = 2 : NP

if w0 <= w\_cumu(j) && w0 > w\_cumu(j-1)

pxnew(:,i) = px(:,j);

break;

end

end

pw(i) = 1/NP;

end

px = pxnew;

end

end

3 仿真测试模式：

针对本任务的误差分析要求，我们设置了两个参数来进行快速的分析切换

test\_mode;

if\_using\_prepared\_data;

test\_mode具有1到4种模式。

mode 1：普通仿真模式，在NP = 50的情况下运行定位仿真，将得到GND，Odom和Est三个位置的动画和最终预测结果。

mode 2：运行时间测试模式，将会将NP从10以10为步长增大到500（共五十次定位仿真），运行结束将得到粒子滤波定位运行时间随粒子数目的变化图像。由于运行时间较长，可以指定if\_using\_prepared\_data = 1，从而使用已经得到的runtime与NP数据，得到图像。数据已经存储在了runtime.mat中

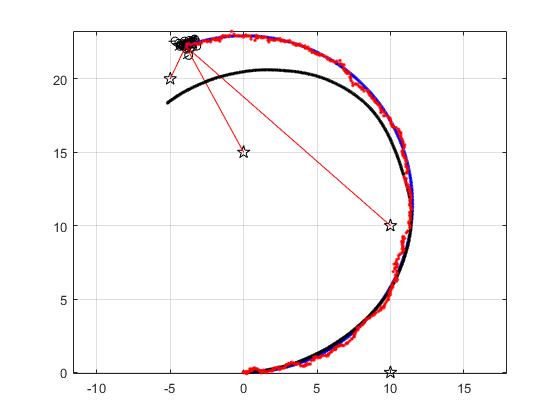
mode 3:定位误差测试模式，将会将NP从10以10为步长增大到500（共五十次定位仿真），运行结束将得到粒子滤波定位运行误差随粒子数目的变化图像。由于运行时间较长，可以指定if\_using\_prepared\_data = 1，从而使用已经得到的avgerror(average error)与NP数据，得到图像。数据已经存储在了avgerror.mat中。

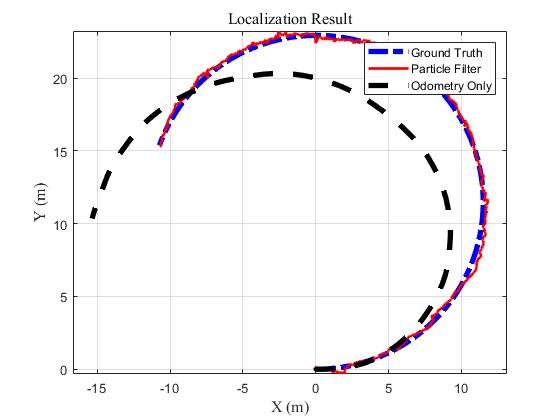
mode 4：误差随时间变化测试模式。将会将NP从10以10为步长增大到500（共五十次定位仿真）当NP 为50 200 400 时，程序将绘制对应粒子数下的定位过程中，误差随时间（60s的仿真过程）的变化曲线，绘制完后，程序将暂停，在命令行按任意键继续运行。整个程序结束后，也会绘制粒子滤波定位运行误差随粒子数目的变化图像。

4、粒子滤波任务结果：

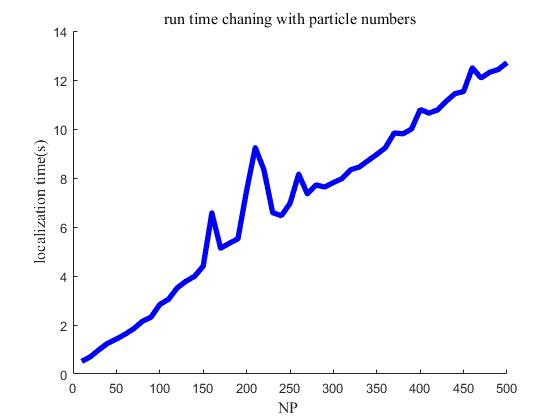
（1）仿真结果，NP = 50

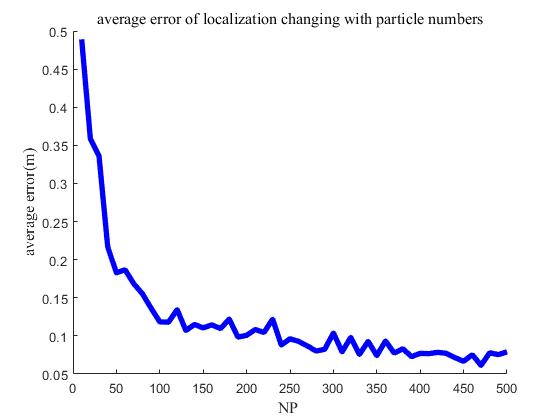
下两图分别为定位过程中的动画一帧和这个定位过程ground truth,odometry和estimation位置。



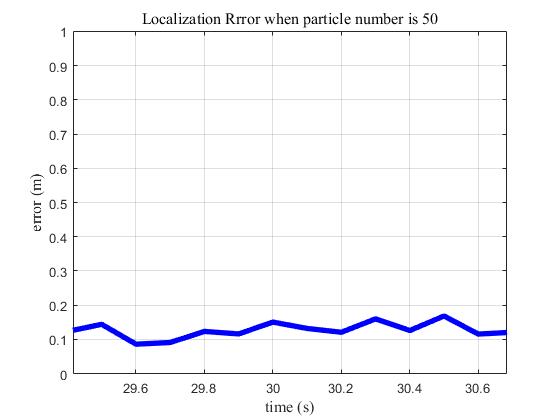


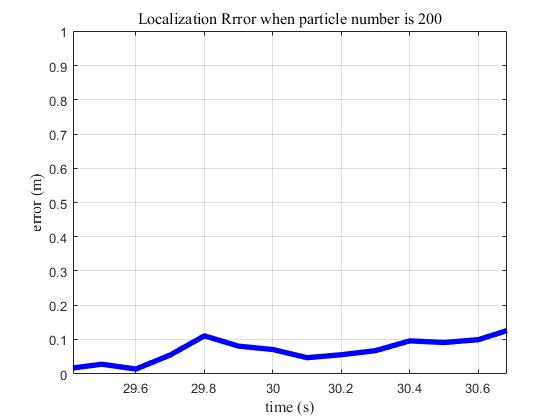
（2）误差随定位粒子数变化曲线



（3）误差随粒子数变化曲线：

（4）特定粒子数下过程中误差随时间变化曲线

NP = 50

NP = 200

NP = 400

