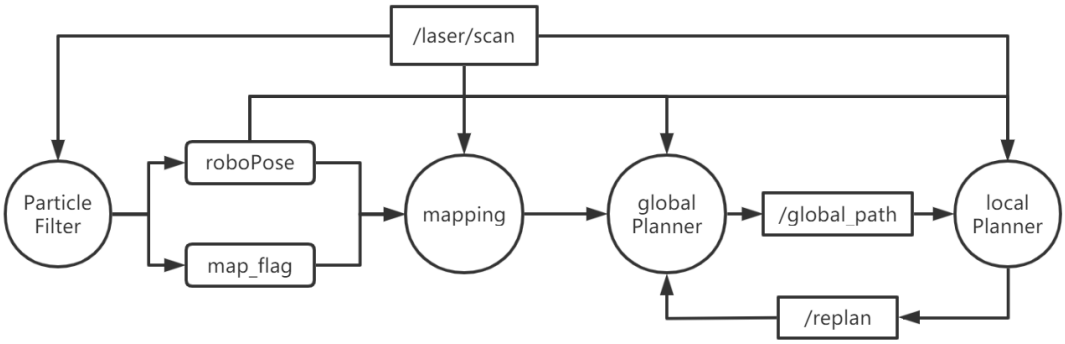
面向动态环境的MCL定位导航

Michael Gao

1.整体结构



`Particle Filter`：

- 接收一次MapServer发出的初始地图用于构建似然地图，似然地图后续不再更新
- 根据laser信息计算粒子权重，输出：
  - `roboPose`：小车的粒子滤波估计位置
  - `map_flag`：根据粒子状态决定是否mapping，向`mapping`节点发出信号

`mapping`：

- 在`map_flag=1`时对此帧建图

`global Planner`：

- 根据mapping的结果地图带权A*规划，接收到`replan`信息则重规划，生成`global_path`路径

`local Planner`：

- 根据`global_path`用DWA局部规划跟踪路径，DWA失效时发出`replan`请求

2.Kidnap Detection

由于要求初始状态是机器人绑架状态，所以机器人初始估计位置并不在真值附近。如果在机器人绑架状态下建图则必定导致建图错误，进而影响全局路径规划。

同时由于只有在DWA失效时才会发出replan请求，即使机器人再次在定位准确状态下对错误建图区域重新建图，也不会触发replan，最终会导致机器人沿着错误规划的路径一直执行。

机器人绑架到右上角回归到正确位置后，错误建图情况下生成路径如图所示：


为了避免这种情况，需要做Kidnap Detection，在Kidnap状态下不mapping.

共尝试了3种Kidnap Detection方法：

2.1.Kidnap Detection方法

a.信息熵法

> Yi C and Choi BU. Detection and recovery for kidnapped robot problem using measurement entropy. In: Kim TH, et al, Grid and distributed computing, 2011, Berlin, Heidelberg: Springer, pp. 293–299.

定义粒子群的信息熵如下：

$$H(t) = -\sum_{n=1}^{N} \omega_t^n \log \omega_t^n$$

信息熵表示了信息的不确定程度，ω_t是每个粒子的归一化权重，可以理解为机器人处于该点的概率分布。

判定kidnap的准则如下：

$$\text{kidnapped}_t = \left\{ \begin{array}{ll} 1 & H(t) > \pi \\ 0 & \text{Otherwise} \end{array} \right\}$$

由于信息熵是对于所有粒子权重的分布描述，导致在不同情况下信息熵差值并不显著，不能很好地识别出kidnap状态。

b.速度限制法

> Bukhori, I., & Ismail, Z. H. (2017). Detection of kidnapped robot problem in Monte Carlo localization based on the natural displacement of the robot. *International Journal of Advanced Robotic Systems*.

该方法的主要思想是当机器人的估计位置运动速度超出机器人的实际最大速度时，认定此时在kidnap状态。

但这种方法的问题是，如果在绑架收敛到真值附近过程中可能陷入局部极大值，很可能在局部极大值处停留，此时也会发生在局部极大值处的错误建图。

c.w_max_min法

对所有粒子权重排序，取权重最大的前10个粒子和权重最小后10个粒子：

```cpp
vector<double> w_step;
for(int i=0; i<particle_num; i++){
    w_step.push_back(particles[i].weight);
}
sort(w_step.begin(), w_step.end());

vector<double> w_max;
vector<double> w_min;
double w_num = 10;
for(int i = 0; i < int(w_num); i++){
    w_min.push_back(w_step[i]);
    w_max.push_back(w_step[w_step.size()-i-1]);
}
```

分别计算平均权重：`w_max_mean`，`w_min_mean`：

```cpp
double w_max_mean = accumulate(w_max.begin(), w_max.end(),0.0);
double w_min_mean = accumulate(w_min.begin(), w_min.end(),0.0);
H = w_max_mean - w_min_mean;
```

定义H为前10个最大权重和后10个最小权重差值：

$$H(t) = \sum_{i=1}^{10} \omega_{max} i - \sum_{j=1}^{10} \omega_{min} j$$

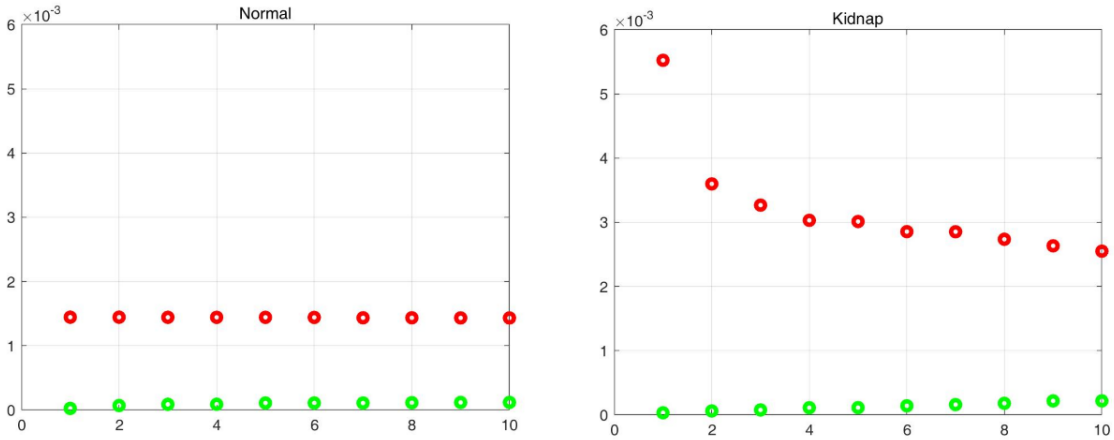
判定kidnap的准则如下：

$$\text{kidnapped}_{\ t} = \begin{Bmatrix} 1 & H(t) > H_{th} \\ 0 & \text{Otherwise} \end{Bmatrix}$$

直观理解，当机器人处于kidnap状态时，只有少数几个粒子在真值附近位置，或者在局部极大值处，ω较大，又由于ω是归一化值，故kidnap状态下ω_{max}集合中的值更大，H值更大。

当机器人在真值附近时，大多数粒子都在真值附近位置，相对的ω_{max}集合中的值会比kidnap状态下更小，H值更小。

下图展示了Normal状态下$H = 1.35$和Kidnap状态下$H = 3.07$的两集合情况：



通过实践发现，这种方法能有效地区别出机器人是否在真值附近，让`mapping`节点在非Kidnap情况下再建图，大大提高了mapping的准确性。

2.2.有条件的建图

`mapping`节点在非Kidnap情况下再建图，在`Particle Filter`节点判断是否可以mapping，并向mapping节点发出信号。

`Particle Filter`节点将是否mapping的信号整合到`pf_odom` topic中：

```
odom_pub = n.advertise<nav_msgs::Odometry>("pf_odom", 1);
if(w_diff < 1.4){
    odom.pose.pose.position.z = 1.0;
}
odom_pub.publish(odom);
```

`mapping`节点：

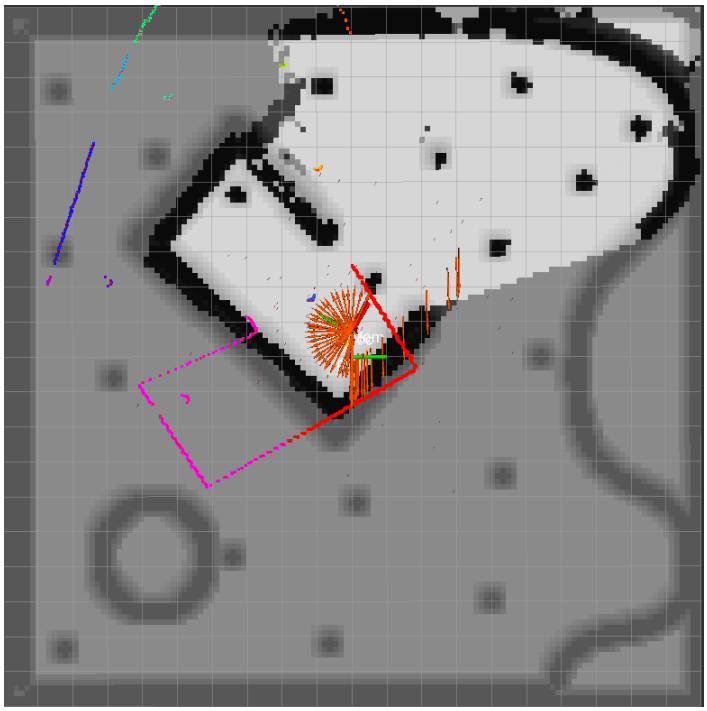
```
void mapping::process(nav_msgs::Odometry odom)
{
    cout<<endl<<"------seq:  "<<laserScan.header.seq<<endl;
    //mapping signal: odom.pose.pose.position.z = 1.0
    if(odom.pose.pose.position.z < 1.0){
        return;
    }
    ...
}
```

3.ICP doMotion 失效问题

在实验过程中发现，在小车运动角速度过大时，ICP会失效。即因为小车运动太剧烈，特征距离太远匹配失败，导致两帧之间匹配失败，最终导致ICP_Odom失效。

为防止这种情况，在之前的代码中已经加入了跳帧操作，即重新选择特定帧的特征作为之后匹配的目标帧。这样一来可以防止ICP_Odom失效，但中间没有完成匹配的帧将遗失ICP_Odom信息，导致Particle Filter中的`doMotion`不能获得任何信息。

该问题发生时情况如下(yaw speed = 1 rad/s)：



遗失的ICP_Odom信息只能依赖于`Particle Filter`的`doObservation`部分补偿，逐渐收敛到真实位置。但是`doObservation`有一定概率不能快速收敛，会导致定位长时间错误。

此问题没有特别好的解决方法，故修改DWA速度和角速度限制，使整体速度下降：

```
self.max_speed = 0.4  # [m/s]
self.min_speed = -0.4  # [m/s]
self.max_yawrate = 0.8  # [rad/s]
```

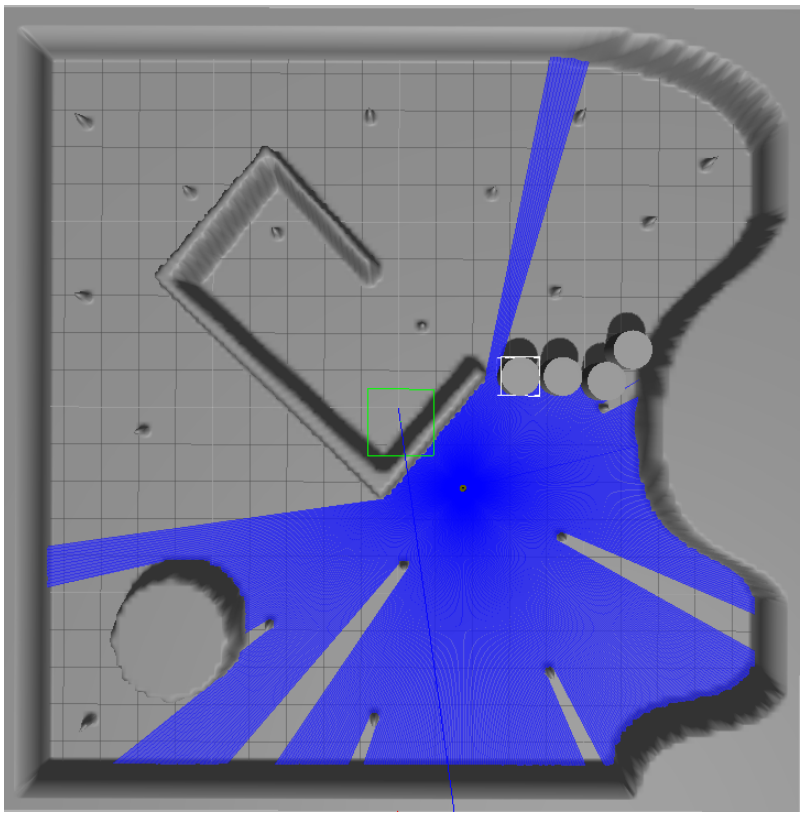
4.实验结果

运行方法：

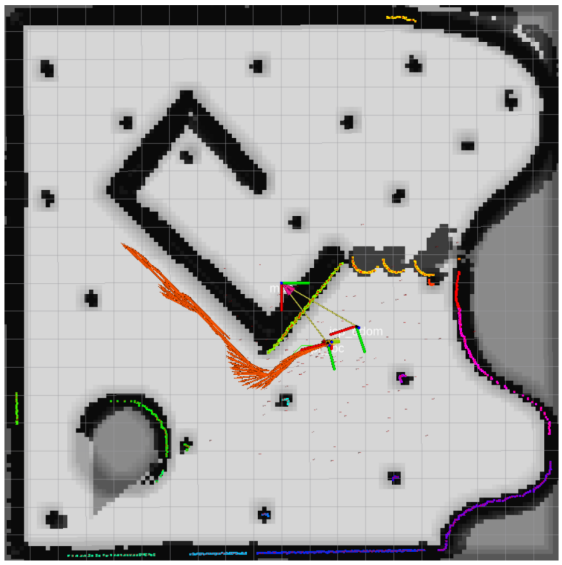
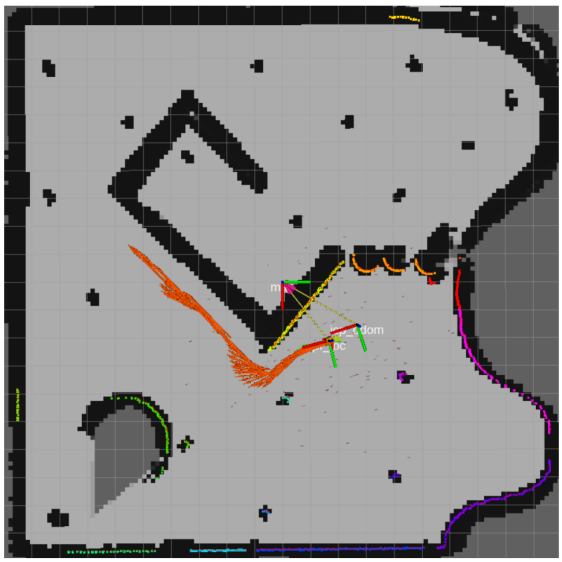
```
roslaunch course_agv_slam_task particle_filter.launch
roslaunch course_agv_nav replan.launch
roslaunch course_agv_slam_task mapping.launch
roslaunch course_agv_slam_task icp_exlm.launch
```

视频结果见`video/mclall.mp4`.

仿真中动态障碍物如下图摆放：



最终到达终点时结果如下，右侧为仅显示建图地图的结果：

整个过程中，除了在经过地图左上角时由于ICP里程计匹配到的landmark数量过少触发了之前报告中解释过的ICP跳帧操作外，其他位置定位的x和y方向误差均控制在0.15m左右。

由于最小栅格大小是0.155m，故次误差产生是正常的，整体定位较为准确。