

简单的二维平面粒子滤波定位

一、粒子滤波

粒子滤波的思想基于蒙特卡罗方法，是通过寻找一组在状态空间中传播的随机样本（粒子）来近似表示概率密度函数，用样本均值替代积分运算，从而获得系统状态的最小方差估计的过程。其核心思想是通过从后验概率中抽取随机状态粒子来表示其分布。

粒子滤波方法的基本步骤如下：

1. 随机生成一组粒子群 `create_uniform_particles(x_range, y_range, hdg_range, N)`

该函数随机生成了 `N` 个粒子。由于每个粒子都代表一个机器人的可能状态，所以需要为粒子设置x坐标 `x_range`，y坐标 `y_range` 以及朝向角 `hdg_range`，并使其服从一定的随机分布。其中，存在反馈的值（如x, y坐标值）可以进行更新与校正，故可以采用全地图上的 均匀分布；而朝向角度不存在反馈，仅依靠距离间接反馈时可能会导致较大误差乃至无法准确预测的后果。

为了解决该问题，考虑通过高斯分布 `create_gaussian_particles(init_state, noise, N)` 生成粒子群时给定确定的初始朝向角（含高斯噪声）、在已知机器人初始位置时也可给定确定的坐标（含高斯噪声）。或者通过给出每步绝对朝向作为反馈，使得朝向偏离的粒子更多的被重新取样，保留朝向偏离较小的粒子，该方法将在权值计算处详细介绍。

2. 预测粒子的下一状态 `predict_ptc_state(particles, step_displacement, noise, N):`

在输入转向 `step_displacement[0]`、速度 `step_displacement[1]` 作为输入信号后，该函数将预测 `N` 个粒子 `particles` 在下一个时刻的状态，即预测机器人在该时刻可能的状态。

考虑到输入信号可能存在不稳定性，故将输入信号也加入参数为 `noise` 的高斯噪声作为干扰，以校验该滤波算法的鲁棒性。

3. 更新粒子权值 `update_ptc_weight(particles, weights, rd, err, landmarks)`

由于粒子 `particles` 生成的随机性与预测粒子状态时的噪声干扰，部分粒子与机器人真实位置距离较远。为了提高预测准确性，通过机器人测量自身与各路标 `landmarks` 之间的真实距离 `rd`（含参数为 `err` 的高斯噪声），并将其和粒子与路标间距比较，依此赋权值 `weights`。具体权值计算方法如下：

- 测量机器人与路标真实距离 `rd`
- 计算粒子与路标预测距离 `particles_distance`
- 生成期望值 `particles_distance`，方差 `err` 的正态分布
- 取得 `distance = rd` 处的概率密度作为权值
- 将粒子与每个路标计算得到的权值 `weights` 累乘
- *若存在朝向反馈，计算朝向 `degree`，并生成以之为期望，方差 `err_degree` 的正态分布，取得 `real_degree` 处的概率密度作为权值，并累乘
- 将每个粒子权值加一个极小量 `1.e-300`，避免权值为0
- 将权值单位化，使其和为1

4. 重取样 `simple_resample(particles, weights)`

一些粒子偏离真实位置较远，若舍去则粒子个数逐渐减少，在移动步数较多时，预测结果将逐步偏离。所以将权值小的粒子重取样，以保证粒子个数与准确性。

通过 `num_of_effected_particles(weights)` 函数判断是否需要重取样，主要判断的是粒子密集程度（权值接近程度）。函数返回值为平方和的倒数，可知当 `weights[i]` 均相等时取最大值N，故将其与N/2比较，若返回值大于N/2，则视为粒子分布足够密集，不需要重取样。

若判定需要重取样，则执行取样函数。使用库 `numpy` 中的 `cumsum` 函数，获得权值按序加和的数列，称 `和权值列`，其中每两个值之差为该索引对应的粒子权值，易知粒子权值越大，对应索引差值越大，在区间内占比越大。再生成 (0,1) 间随机数，使用 `searchsorted` 函数取得随机数在 `和权值列` 中落于哪一区间，从而得到对应的索引值，而权值大的粒子区间较宽，故更易被选中；权值小的粒子区间较窄（或被舍去为0），被舍去。最后根据索引列重新取粒子、权值（并单位化），个数不变，但索引列对应的是权值大的粒子，很可能出现重合的情况，在可视化结果中仅可见几个点。

该取样仍存在的问题是，判断函数仅判断粒子密集程度，不校验准确性。其重取样样本为原本生成粒子中的点，也无法保证准确性。若预测点偏离较大且密集（相对路标），将不会进行重取样；若进行重取样，也无法获得足够精确的点。调试发现重取样对噪声的容忍极限 `err_max` 较高，鲁棒性较强。但若干扰维度增加，其偏离概率也明显增加。

5. 计算估计值 `estimate(particles, weights):`

将重取样后的粒子位置取平均值 `mean`，作为最终预测结果，且计算方差 `var` 并作为返回值输出。该均值与准确值比较可体现预测准确性；方差可体现预测精度及稳定性。

6. 可视化 `all_plot`

为便于看出定位效果，采用 `python` 库 `matplotlib` 进行二维可视化，最后总效果图如下：

其中红点为预测机器人所在位置，黑色十字为理论实际位置，彩色圆为路标。可以看出随着迭代次数的增加，准确度逐渐提升并稳定在误差允许范围内。

