1. 贝叶斯公式：,集和条件概率和全概率公式。其中z为观测量，x为待估量。为后验概率，为似然，为先验。以摄像头为例，z代表观测，x代表相机位姿。

因为往往不能直接求得，而对应传感器的测量模型，可以在前期通过大量测试获得。X已知时，通过x物理模型得到的z是分布的均值，方差则由模型的准确度决定，由于传感器往往受噪声影响，方差一般不为0。若方差为0，则所有状态就确定了。预测和测量巧妙地运用了运动模型和测量模型不同地性质，运动模型从上一状态根据控制量或里程计计算得到当前状态，如控制动作、里程计和IMU，它们是内部传感器，感受机器人自身的运动，各状态具有连续性，测量模型只与当前状态有关，如摄像头、激光雷达，它们是外部传感器，通过环境感受自身状态。不具连续性。

1. p4状态估计：是根据系统的先验模型和测量序列，对系统内在状态进行重构的问题。方法总结
2. p38、p116贝叶斯推断：批量式，全贝叶斯公式
3. P34、p110最大后验估计（MAP）：批量式，比最大似然多了先验项。该方法得到的是高斯分布的最大值（模，极大值，概率最大的点），贝叶斯推断得到的是均值（期望），线性高斯模和均值重合，非线性非高斯模和均值不重合
4. P118最大似然方法：不考虑先验，仅使用测量数据的情况。相机在每个点都会有多个观测，该方法就是使乘积最大。所有观测共用一个x，调整x可得到不同的高斯分布，参数已确定，均值为物理模型由x推算得到观测，如相机的像素坐标，方差为传感器固有特性。再通过传感器测量值z在各观测点分布取概率，然后相乘，得到使该积最大的x。
5. P52卡尔曼滤波方法：递归式，先通过状态转移矩阵得到预测的，控制变量可以直接将相邻两个状态连接起来（状态转移可通过数学公式表达），而观测不行。再使用测量模型更新。线性高斯可以通过优化方法和贝叶斯方法两种途径推出。而非线性非高斯差别就会很显著。
6. IEKF: p94IEKF与MAP相同，都是计算极大值
7. P103 sigmapoint（无迹变换）SPKF:p96当输入概率密度大致为高斯时，蒙特卡罗是最准确的方法，但是计算开销大，sigmapoint（无迹变换）是蒙特卡罗和线性化折中的办法，它比线性化更准确，但计算量稍大
8. P107ISPKF: p110IEKF收敛于MAP的解（模），而ISPKF收敛于全后验概率的均值
9. 粒子滤波：p101粒子滤波额外需求，概率机器人书中描述比较详细
10. p2能观性：动态系统的状态何时能够从该系统的一组观测值中推测出来。
11. p12统计独立和不相关条件，独立必不相关，不相关未必独立。在概率密度函数情况下，两者等价。
12. p13香农信息、负熵：描述某个量的不确定性
13. p14互信息：刻画了已知一个随机变量的信息之后，另一个随机变量的不确定性减少了多少
14. p14费歇尔信息量：明确定义出利用已有观测值估计参数的效果和衡量估计方式好坏的标准
15. p16isserlis定理：计算高阶（二阶以上）
16. 3.1得出批量式线性高斯求解方程，3.2使用三角矩阵分解高效求逆
17. p55边缘化：使某个状态的系数矩阵除自身对应之外非零，其他块为0
18. p64卡尔曼滤波是最优线性无偏估计，即方差正好是克拉美罗下界
19. p87非线性表示和计算的近似方法
20. p91有偏差-期望，不一致-方差
21. p92将后验不断作为先验值，得到更好的线性展开点，一个状态多个观测的，可以迭代不同的观测点
22. p113高斯-牛顿两种推导方法
23. p123全局、滑窗和时间步优化比较，精度和实时性的权衡
24. p133为了有效地抑制偏差带来的影响，我们必须知道偏差的精确值。然而大多数情况下，并没有办法知道偏差的精确值（它甚至会随着时间变化）。因此在估计问题中，我们应该尝试着把偏差也估计出来。
25. p138外部数据关联和内部数据关联，数据关联时所用专业知识相对估计问题来说是外部的
26. p139外点常见评判基准：将超出平均值3个标准差的测量值作为外点
27. p139常见处理外点的方案：随机采样一致性，M估计，M代表最大似然的，如图5-7，梯度更小的代价函数，大的误差对应的权重不会太大，对结果的影响也减弱了，更加鲁棒。代价非常大的项通常被给予较小的置信度。其与协方差估计p143类似。
28. p156欧拉序列的奇异性
29. p170弗莱纳参考系，适合独轮车模型的坐标系
30. p174外参估计的两种方法
31. 通过一些校正的方法测得
32. 直接作为一个待估的状态估计量求解
33. p183IMU模型
34. p296图像拼接
35. p302利用稀疏性：舒尔补（schur？）、cholesky分解
36. p308BA方法是最大似然问题，SLAM是一个最大后验概率
37. p310SLAM与BA的A11的区别，更倾向于cholesky分解
38. 1