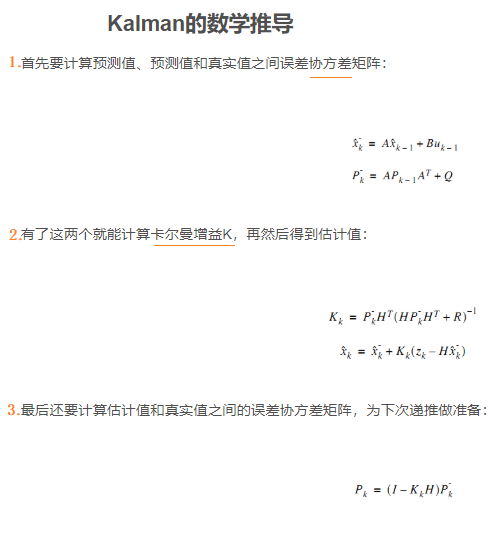
## 1 滤波

### 1.1 卡尔曼滤波

标准的卡尔曼滤波（Kalman Filter，KF）是针对线性系统设计的，只适用于线性系统，是一种时域滤波器。

卡尔曼滤波是以最小均方根误差为估计的最佳准则，来寻求一套递推估计的算法，其基本思想是，采用信号与噪声的状态空间模型，利用前一个时刻的估计值和现在时刻的观测值来更新状态变量的估计，求出现在时刻的估计值。

卡尔曼滤波的实质是由量测值重构系统的状态向量。它以“预测-实测-修正”的顺序递推，根据系统的量测值来消除随机干扰，再现系统的状态，或根据系统的量侧值从被污染的系统中恢复系统的本来面目。



* + 1. **EKF**

扩展卡尔曼算法（Extended Kalman Filter，EKF）利用泰勒展开将非线性系统线性化；是将期望和方差线性化的卡尔曼滤波器。

扩展卡尔曼算法的缺点：

（1）在强非线性系统下的误差很大，对复杂非线性状态估计时收敛速度慢、估计精度低。

（2）

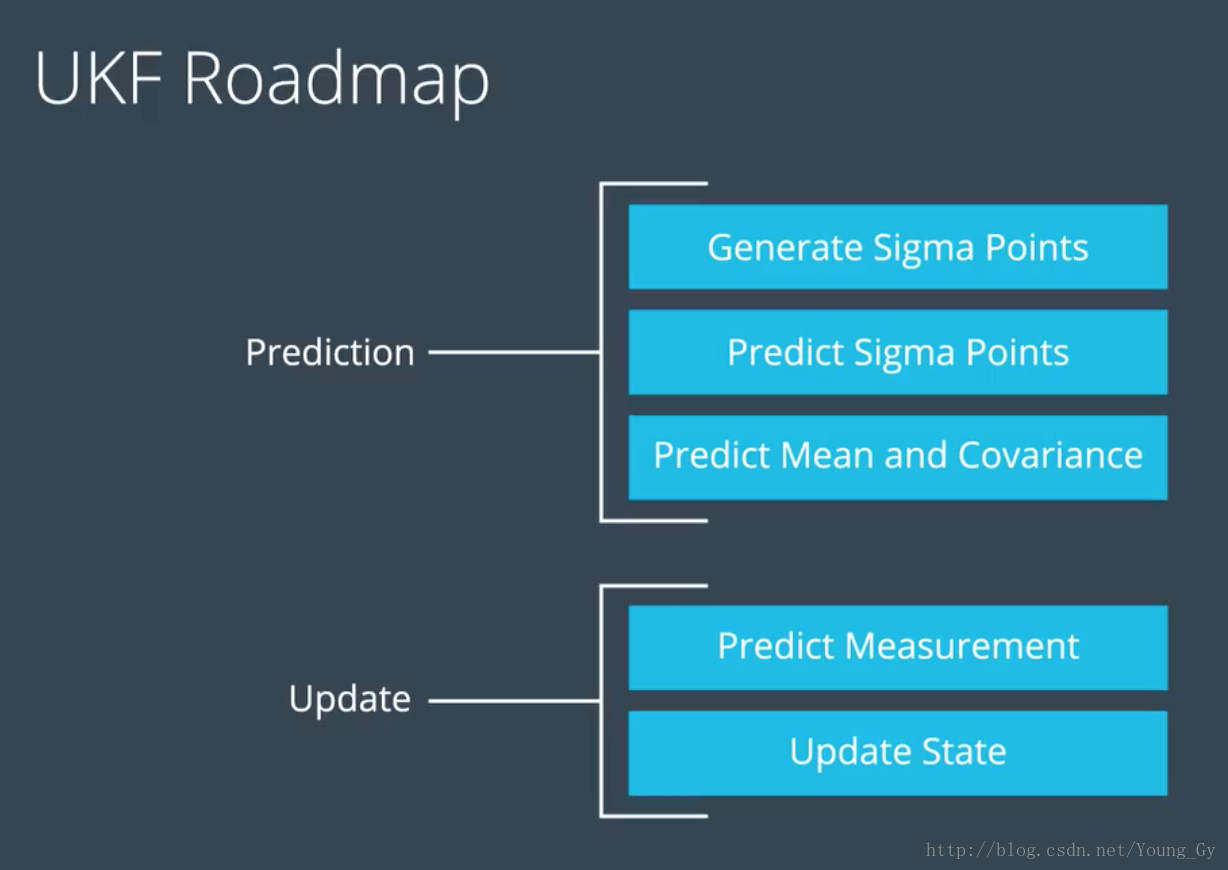
* + 1. **UKF**

随机状态变量沿非线性函数的传播问题是非线性滤波的关键。非线性状态滤波过程的实现包括：{一步预测}与{策略修改}两个阶段。

UKF的核心思想是“近似非线性函数的概率密度分布比近似非线性函数更容易”。UKF=Unscented transform + Kalman Filter，即UKF可以看作是基于UT技术的卡尔曼滤波。

无迹卡尔曼滤波（Unscented Kalman Filter，UKF），其计算精度相比EKF更高并省略了Jacobian矩阵的计算。

无迹卡尔曼滤波通过一组确定的加权采样点来逼近随机变量的分布函数，通过这组采样点的非线性变换，捕获随机变量经非线性转换后的统计特性【[[1]](#footnote-1)】。使用UKF需要选择合适的参数，才能达到较好的滤波效果【[[2]](#footnote-2)】。

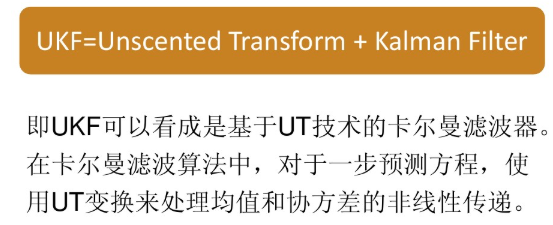


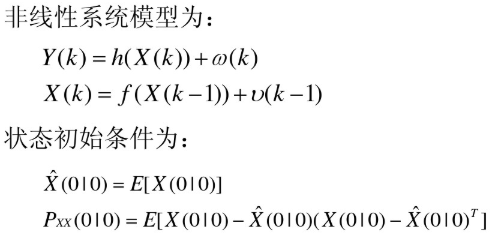
应用场景1：在Google开源SLAM软件cartographer中，相对《SLAM for dummies》使用了更为复杂、性能更好的Scan匹配与UKF算法，这里简单介绍下cartographer中使用的UKF算法。

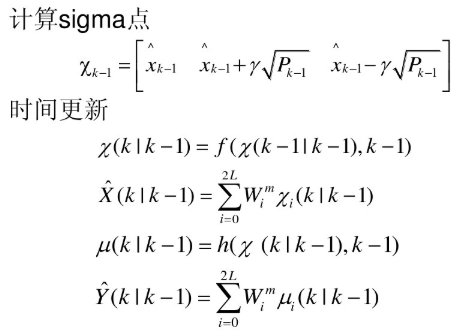
对于一维非线性模型，UKF的MSE均值和标准差明显小于CKF，则UKF精度更高。当状态维数为3时，两种滤波方法性能基本相当。总结，对于低维（n<3）非线性系统，UKF精度高于CKF，建议选择UKF；对于n=3的非线性系统，UKF与CKF精度相同，考虑CKF选择更少的采样点；对于高维（n>3）非线性系统，UKF估计精度低于CKF，建议选择CKF作为滤波方法。若k=0,尽管两种滤波方法精度相同，但CKF采样点更少，因此建议选择CKF【[[3]](#footnote-3)】。

状态的时间更新。

状态的测量更新。







*Q*：表示过程激励噪声的协方差，它是状态转移矩阵与实际过程之间的误差。这个矩阵是KF中比较难确定的一个量，一般有两种思路：一是在某些稳定的过程可以假定它是固定的矩阵，通过寻找最优的Q值使滤波器获得更好的性能，这是调整滤波器参数的主要手段，Q一般是对角阵，且对角线上的值很小，便于快速收敛；二是在自适应卡尔曼滤波（AKF）中Q矩阵是随时间变化的。

*zk* ：表示测量值，是m阶向量。

*H*：表示量测矩阵，是*m×n* 阶矩阵，它把m维测量值转换到n维与状态变量相对应。

*R*：表示测量噪声协方差，它是一个数值，这是和仪器相关的一个特性，作为已知条件输入滤波器。需要注意的是这个值过大过小都会使滤波效果变差，且R取值越小收敛越快，所以可以通过实验手段寻找合适的R值再利用它进行真实的滤波。

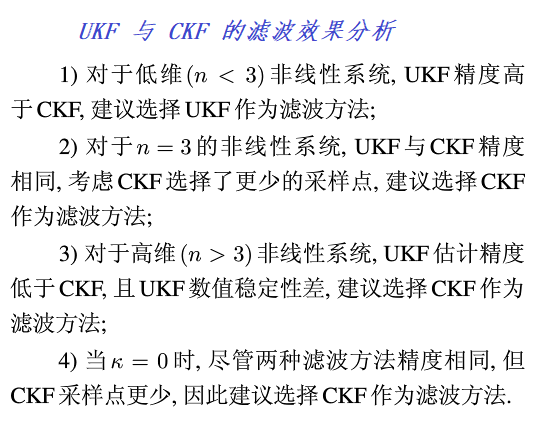
##### 1.1.2.1 关于R、Q的选择依据

时间2018-12-14，目前，还没有很好的调整R、Q的办法，所以暂行的方法是：

1. 固定R为0.5，然后调整Q值。
2. Q值的调整方法是，分析不同接地形式的过渡电阻0Ω、500Ω、1000Ω、2000Ω的零序电流波形，规律是同种接地情况下随着过渡电阻阻值的增大而零序电流幅值会减小，根据该I0波形中采样点原始值（绝对值）最大的前3个值取平均，再与该I0通道系数32768做比例，该比值越小则Q值越小，表示滤波越依赖预测值；可能会推导出一个Q值与比例值关系的方程。
   * 1. **CKF**

容积卡尔曼滤波（Cubature Kalman Filters Algorithm, CKF），是一种非线性滤波方法。

CKF通过2个等权值容积点来传播系统状态的均值和方差，能获得较高的滤波精度。CKF具备UKF的优点，并可以较好地处理非线性系统的估计问题，使用更少的采样点，进一步降低了计算代价，且无需像UKF一样调节各参数因子，其容积点及其权值仅由状态维数唯一确定，可以预先计算和存储，所以该算法设计和实现更为简单。



传统的CKF基于三阶容积准则而提出，因此滤波精度受到限制，为进一步提高CKF滤波性能，有些论文中将三阶扩展到五阶。

近年由Arasaratnam等人提出了容积卡尔曼滤波(CKF)算法【[[4]](#footnote-4)】，该算法的核心思想是利用球面径向积分原则选取一组确定的采样点来逼近状态后验分布。从CKF的实施流程来看，其数学理论严谨，参数选取方便，收敛效果好，相比于UKF和CDKF优势明显，因此该算法一经提出，便在多个领域得到推广应用【[[5]](#footnote-5)】。

##### 1.1.3.1 正定矩阵

广义定义：设M是n阶方阵，如果对任何非零向量z，都有zTMz> 0，其中zT 表示z的转置，就称M为正定矩阵。正定矩阵的主对角线上的元素都大于0。例如：B为n阶矩阵，E为单位矩阵，a为正实数。在a充分大时，aE+B为正定矩阵。（B必须为对称阵）。

它要求矩阵的所有特征值必须大于零，故分解的下三角的对角元也是大于零的。

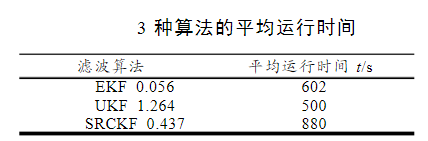
##### 1.1.3.2 Cholesky 分解

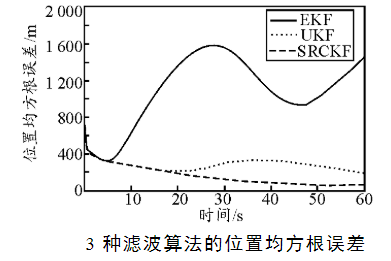
Cholesky 分解是把一个对称正定的矩阵表示成一个下三角矩阵L和其转置的乘积的分解。

* + 1. **SRCKF**

平方根容积滤波算法（Square-Root Cubature Kalman Filters，SRCKF)，使用基于容积原则的数据积分方法直接计算非线性随机函数的均值和方差。该算法实现时只需要计算函数值，避免了求导运算，降低了计算复杂度，且该算法传播了状态协方差的平方根，确保了协方差矩阵的对称性和半正定性，改进了数据精度和稳定性。

CKF的平方根形式（SRCKF）是在CKF的基础上提高了滤波的稳定性。平方根CKF（SRCKF）在CKF的基础上，以Cholesky分解的形式直接传播和更新状态协方差矩阵的平方根，降低了计算负担，获得了更高的计算效率，同时能保证协方差矩阵的非负定性，避免了滤波器的发散，提高了滤波的收敛速度和数值的稳定性【[[6]](#footnote-6)】。





### 1.2 粒子滤波

该滤波方法是一种基于概率方法的滤波。

粒子滤波的缺点：PF实现时需要产生大量的粒子，计算量非常大，很难满足实时性的需求，且有粒子退化和贫化等问题。

## 2 时间序列异常检测

### 2.1 解决思路（寻找U0突变起点）

**1.** 【2018-12-26】找到前 3 CYCLE 中最大的 MAXscore； 这个思路中 【 2倍的比例关系】 仅是根据已有数据的试验分析。

**2.** 【2018-12-26】从第 (128\*3+1) 点开始与 MAXscore 比较。

2.1 如果 MAXscore < CurrentScore <= 2\*MAXscore，则用CurrentScore更新 MAXscore；

2.2 如果 CurrentScore > 2\*MAXscore，则认为找到了突变点。

**3.** 【2018-12-27】5+8=13 个故障波形，每个波形128点。

3.1 如果，在 13 个周波内，MAX-Score 不在第4~5周波内，且出现在第5个周波之后，则取前3个周波内最大的Score（记为Second-Score），然后从第4个周波开始判若有*连续3个Score*大于1.2\*Second-Score则认为是找到了故障起点，且该起点只可落入第5个周波之前。这里采用连续 3个Score而非 5 个点是因为导数可能会变小，但是曲线U0仍然会处在正半波或负半波。

3.2 如果，*连续3个Score不*大于1.2\*Second-Score，则取这连续3个Score的平均值大于1.3\*Second-Score，则作为后备满足条件。

3.3 如果，MAX-Score 存在在第4~5周波内，则该MAX-Score就是U0的故障起点。

3.4 为了避免找到的起点靠前，所以取起点开始的5个点，若其中至少有三个点符合暂态正负特性则判为区内或区外。

3.5 若前3个周波内最大的Score（记为Second-Score）为0，则从第4周波开始连续3个不为0的Score记为U0故障起点。

### 2.2 EGADS（雅虎开源代码）

EGADS (Extensible Generic Anomaly Detection System)是基于 Java 的雅虎开源代码（https://github.com/yahoo/egads），可以用于时间序列异常检测。

### 2.3 局部异常因子算法Local Outlier Factor（LOF）

LOF是一种基于密度的聚类算法。

### 2.4 肘部法则

肘部法则，可以用来确定K-Means方法中的K值。

### 2.5 数据源（异常点）

用于检验算法准确度的数据源 Numenta Anomaly Benchmark（NAB），在GitHub上的开源地址（https://github.com/numenta/NAB）。

### 2.6 Z-score（异常点）

Z-score的方法无法正常检测出这个波形【SAR-device.sdb.await\_\_DCU1923ZeroOrder(4\_8)2017-11-10 11\_51\_02\_831016.csv】；该波形的特点是故障前，有零序电压，故障时刻零序电压发生很小的波动。

### 2.7 智能运维中的异常检测（异常点）



## 3 代码工具库

### 3.1 Eigen

Eigen是一个高层次的C ++库，有效支持线性代数，矩阵和矢量运算，数值分析及其相关的算法。Eigen是一个开源库，从3.1.1版本开始遵从MPL2许可。除了C++标准库以外，不需要任何其他的依赖包。Eigen使用的CMake建立配置文件和单元测试，并自动安装。如果使用Eigen库，只需包特定模块的的头文件即可。

### 3.2 GSL

**GSL简介：**GNU Scientific Library（GSL）是一个开源的科学计算的函数库，里面有大量的数学计算函数，非常强大。

下载GSL安装文件：

<http://mirrors.ustc.edu.cn/gnu/gsl/gsl-2.4.tar.gz>

官方的网址是

http://www.gnu.org/software/gsl/

参考文章进行安装和使用：

https://blog.csdn.net/u012248802/article/details/80655902

### 3.3 luminol（时间序列异常检测）

Luminol 是用于时间序列数据分析的 Python 库，它支持的2个主要功能是异常检测和相关性；它可以对给定的一个时间序列，检测数据是否包含异常，并给出异常的时间戳；给出2个时间序列，找出这2个序列的相关系数。该库不依赖于时间序列值的任何预定义阈值。相反，它将每个数据点分配一个异常分数，并使用分数标识异常。

库包含两个类：AnomalyDetector 和 Correlator，有两组 api，每个类对应一个。

#### 3.3.1 Derivative Algorithm（导数算法）

该算法的思想是：

Instead of data point value, it uses the derivative of the data point.

## 4 术语

1. SVM 支持向量机。
2. 邻近算法：KNN、K-邻近、K-Means
3. 肘部法则
4. 轮廓系数（聚类评估效果）
5. 分段线性表示。

分段线性是一种典型的非线性；可以关注 Sigmoid函数，S型生长曲线。

1. ARIMA
2. RNN+LSTM
3. 趋势加速度
4. Mann-Kendail（突变点检测、趋势检测）

1. 【参考论文】Julier S J, Uhlmann J K. Unscented filtering and nonlinear estimation[J]. Proc of the IEEE, 2004, 92(3): 401-422. [↑](#footnote-ref-1)
2. 【参考论文】穆静, 蔡远利. 平方根容积卡尔曼滤波算法及其应用[J]. 兵工自动化. 2011-06:11 [↑](#footnote-ref-2)
3. 【参考论文】Cubature卡尔曼滤波及其在导航中的应用研究[D]. 哈尔滨工程大学 [↑](#footnote-ref-3)
4. 【参考论文】Arasaratnam I, Haykin S. A numerical-integration perspective on Gaussian filters [J]. IEEE Transactions on Automatic Control, 2009, 54(8): 1254-1269. [↑](#footnote-ref-4)
5. 【参考论文】李贤. 容积卡尔曼滤波方法及其应用研究[D]. 河南大学 [↑](#footnote-ref-5)
6. 【参考论文】基于SRCKF的自适应高斯和状态滤波算法[J].刘瑜, 董凯, 等. 控制与决策. [↑](#footnote-ref-6)