**TwinCAT 3无迹卡尔曼滤波的实现方法**

|  |  |
| --- | --- |
|  | 作者：袁英杰职务：华东区 技术支持工程师邮箱：yj.yuan@beckhoff.com.cn日期：2025-08-06 |
| **摘 要：**本文介绍ST语言下无迹卡尔曼滤波UKF的实现与应用。 |
| **附 件：**

|  |  |  |
| --- | --- | --- |
| 序 号 | 文件名 | 备注 |
| 1 | UKF\_test.tpzip | 样例程序 |
| 2 | FB\_UKF.library | 卡尔曼滤波封装库 |
| 3 | FunctionGenerator.library | 信号发生器封装库 |
|  |  |  |

 |
| **历史版本：**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

 |
| **免责声明：**我们已对本文档描述的内容做测试。但是差错在所难免，无法保证绝对正确并完全满足您的使用需求。本文档的内容可能随时更新，如有改动，恕不事先通知，也欢迎您提出改进建议。 |
| **参考信息：** |

目 录

[1. 无迹卡尔曼滤波 3](#_Toc205816239)

[2. 基本方法 4](#_Toc205816240)

[3. 线性模型 5](#_Toc205816241)

[4. 非线性模型 7](#_Toc205816242)

[5. 应用 9](#_Toc205816243)

[5.1. 线性模型 9](#_Toc205816244)

[5.2. 非线性模型 10](#_Toc205816245)

[6. 常见问题 11](#_Toc205816246)

# 无迹卡尔曼滤波

无迹卡尔曼滤波是一种处理​​非线性系统​​状态估计问题的卡尔曼滤波变体。它摈弃了扩展卡尔曼滤波对非线性函数进行雅可比矩阵线性化的方法，转而采用​​无迹变换​​这一核心思想。

第一步，Sigma点采样：在获得当前时刻的状态估计（均值）和协方差矩阵后，围绕该状态估计​​确定性​​地选取一组代表点，称为Sigma点。这些点精确地捕获状态分布的均值和协方差信息。

第二步，无迹变换 / Sigma点传递：将选取的Sigma点​​独立地​​通过非线性状态转移方程（用于预测）或非线性观测方程（用于更新），得到一组传递后的点。

第三步，加权重构：使用特定的权重，对传递后的Sigma点进行​​加权求和​​，计算得到预测状态（或更新后状态）的​​均值​​和​​协方差矩阵​​。

通过这种方式，UKF避免了线性化误差，直接近似了非线性变换后状态的统计特征（均值和协方差），并​​保持了原始卡尔曼滤波预测-更新（黄金五则）的数学框架​​。

对于UKF的公式推到和进一步的概念介绍，建议参考互联网提供的相关知识。

# 基本方法

全局可编辑变量如下：





超采样系数cOversamples，cMultDimension是观测维数，nVector是状态维数。

使用FB\_UKF作为基本功能块，包含了定义的基本变量，如先验、后验状态估计、各种状态矩阵、中间变量等。

功能块下定义了各种基本action和方法：



A\_Alarm定义了基本报警，避免因为矩阵数据发散导致的数据infinity；

A\_DynamicW可以实现动态的过程噪声的调整；

Init\_Matrix可以用作矩阵初始化；

LU\_Inv实现矩阵求逆；

M\_CalculateMatrixBound用作计算输入矩阵数据的边界；

M\_Cholesky用作cholesky分解；

M\_MatrixObservability实现可观性判据，同时可用作李导数观测性判别；

M\_Transpose实现矩阵转置；

MatrxiMultiply（1、2）用作不同维数的矩阵相乘；

Sigma\_Generate用作sigma点的生成。

# 线性模型

FB\_UKF\_Liner继承了基本功能块，实现了如下状态方程的UKF：

$$x\_{1}\left(k+1\right)=x\_{1}\left(k\right)+0.01\*x\_{2}\left(k\right)$$

$$x\_{2}\left(k+1\right)=x\_{2}\left(k\right)$$

$$z\_{1}\left(k+1\right)=x\_{1}\left(k\right)$$



A\_AdjustRW实现输入的测量噪声和过程噪声可调；

A\_Init为变量初始化；

A\_Process为UKF主体。

用户如果有线性模型，可以参照该结构，将模型替换到如下的位置，并且修改GVL中的维数。

首先，在A\_Init中修改状态维数和初始化参数：





其次，在A\_AdjustRW中修改噪声调整的维数：



最后，在A\_Process中修改模型：





# 非线性模型

$$F\left(t\right)=m\*\frac{d^{2}x}{dt^{2}}+c\*\frac{dx}{dt}+k\*x+a\*x^{3}$$

$$x\_{1}\left(k+1\right)=x\_{1}\left(k\right)+T\*x\_{2}(k)$$

$$x\_{2}\left(k+1\right)=x\_{2}\left(k\right)+T\*\frac{x\_{3}\left(k\right)-c\*x\_{2}\left(k\right)-k\*x\_{1}\left(k\right)-a\*x\_{1}(k)^{3}}{m}$$

$$x\_{3}\left(k+1\right)=x\_{3}\left(k\right)$$

$$z\_{1}\left(k+1\right)=x\_{1}\left(k\right)+0.1\*sin⁡(5\*x\_{1}\left(k\right))$$

$$z\_{2}\left(k+1\right)=x\_{2}\left(k\right)$$

其中，x1代表位置，x2代表速度，x3代表力，z1为实际测量值，模拟了传感器的非线性特性

m、c、k、a都是常数。

 由于PLC周期为10ms，故而采用0.01的欧拉离散化。类似的，可以使用离散双线性变换等其他方式进行离散化。



前三个Action和线性模型的FB\_UKF\_Liner类似。

M\_Jacobian计算该模型的雅各比矩阵，用作李导数观测性判别，该方法的修改较为繁琐，在用户自定义模型中可以被省略。该判别和可观性判别一样，虽然对于UKF本身是必要的，但是不可观的模型会体现在计算数据的异常中并由A\_Alarm抛出，故该判别在不严格要求的情况下可以不被使用。

在A\_Init中修改初始化参数：





修改A\_AdjustRW中的结构：



在A\_Process中修改模型结构：





若不修改M\_Jacobian，可以在FB中将其屏蔽：



# 应用

需要注意的是，以下两个实例的实现并不具备超采样功能（修改cOversamples没有用处）。

## 线性模型

线性模型完全可以使用标准卡尔曼滤波，使用UKF不会对运算结果产生影响只是会加大对PLC计算的负担，若该负担可以被承受（task较长，CPU性能够强或者模型维数较低）那使用UKF完全可行，若该负担难以被接受，可使用标准卡尔曼滤波。

使用FB\_UKF\_Liner，MAIN中做如下信号发生器，模拟一个传感器信号，数值为5，带有0.05的随机噪声，如下：



输入输出类型如下：



结果如下图所示，绿色是带有噪声的原始曲线，蓝色是经过UKF后的曲线。



## 非线性模型

使用FB\_UKF\_NLv1，MAIN中给出如下信号源aOut2 = aOut1 + sin(5\*aOut1)，模拟非线性模型，aOut1是为0.5的常数，添加一些0.001的随机噪声，和一个0.005的噪声，如下：



结果如下图所示，橙色是原始噪声信号，绿色UKF后的信号，能够观测到较好的aOut1的值。



# 常见问题

2维矩阵（5个sigma点）下的CPU负载率（10ms）：



3维矩阵（7个sigma点）下的CPU负载率（10ms）：



为避免较高CPU负载和潜在的task溢出风险，目前设定的运算矩阵最大支持6维，体现在GVL Constant和A\_Alarm的报警中。

需要注意的是，目前报警功能并不完善，有可能出现潜在的导致TwinCAT Exception/Page fault的问题，若在应用实现过程中有任何问题，欢迎联系作者。

若用户发现该算法的运行有任何的问题，或对该结构的实现有更好的建议，欢迎联系作者。

**上海（ 中国区总部）**

中国上海市静安区汶水路 299 弄 9号（市北智汇园）

电话: 021-66312666

**北京分公司**

北京市西城区新街口北大街 3 号新街高和大厦 407 室

电话: 010-82200036 邮箱: beijing@beckhoff.com.cn

**广州分公司**

广州市天河区珠江新城珠江东路32号利通广场1303室

电话: 020-38010300/1/2 邮箱: guangzhou@beckhoff.com.cn

**成都分公司**

成都市锦江区东御街18号 百扬大厦2305 室

电话: 028-86202581 邮箱: chengdu@beckhoff.com.cn

|  |  |
| --- | --- |
| 请用微信扫描二维码通过公众号与技术支持交流 | 倍福官方网站：https://www.beckhoff.com.cn在线帮助系统：https://infosys.beckhoff.com/index\_en.htm |
| 倍福虚拟学院：https://tr.beckhoff.com.cn/ |
| 招贤纳士：job@beckhoff.com.cn技术支持：support@beckhoff.com.cn产品维修：service@beckhoff.com.cn方案咨询：sales@beckhoff.com.cn |
|  |