

PX4 的 ECL EKF2 方程推导

Shuyong Chen

2022 年 11 月 28 日

1 简介

PX4 的 ECL (Estimation and Control Library, 估计与控制库) 使用 EKF 对机体状态进行估计, 是一个很成熟很优秀的状态估计模块。因此在很多项目里面得到了广泛应用。但是, 对于 ECL EKF 里面的算法, 却很少有文档进行系统的总结, 而是散落在各个角落。而总结 ECL EKF 的算法, 也相当困难。如果直接看 C++ 代码, 则会被里面复杂庞大的计算式吓到。因为它的算法, 以前采用 matlab 描述, 后来采用 python 描述, C++ 代码则是由符号推导系统优化后的展开式。但是, matlab / python 描述的只是算法的主要部分, 有很多细节的处理, 还需要回头看 C++ 代码的部分。

本文试图对 ECL EKF 其中的主要算法做一个梳理。经验和学识不足, 难免有错漏。欢迎读者讨论和指正。

2 EKF 简单回顾

2.1 标准 KF 系统

对于一个在状态空间描述的线性或者准线性系统, 状态 k 从 $k-1$ 的状态演变而来, 根据

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{G}_k \mathbf{u}_k + \mathbf{w}_k$$

在这里,

- \mathbf{F}_k 是状态转换矩阵。对于简单的问题, 这个矩阵可以是一个常数, 但是对于大多数实际应用程序, 转换矩阵依赖于状态向量的值并每次迭代改变。
- \mathbf{G}_k 是应用于控制向量 \mathbf{u} 的控制输入模型。这可用于为系统的已知控制输入建模, 例如应用于机器人电机的电流、汽车方向盘的位置等。
- \mathbf{w}_k 是过程噪声, 假定从零均值多元正态分布 \mathcal{N} 中提取, 利用协方差矩阵 \mathbf{Q}_k : $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ 。为这个矩阵确定一个合适的值是很棘手的, 并且在卡尔曼滤波器的文献中经常被忽视。

在时间 “ k ” 时刻, 一个真实状态的观测 (或测量) z_k 根据

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

其中

- \mathbf{H}_k 是将状态空间映射到观测空间的观测模型。
- \mathbf{v}_k 是假设为零均值高斯协方差的观测噪声 $\mathbf{R}_k : \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$

注意, \mathbf{H}_k 将状态向量映射到观测值, 而不是相反。这是因为 \mathbf{H}_k 通常是不可逆的, 也就是说, 它不提供对状态的直接可见性。

滤波器的状态由两个变量表示:

- $\hat{\mathbf{x}}_{k|k}$, 时间 k 的后验状态估计, 给出时间 k 之前 (包括该时间) 的观测值;
- $\mathbf{P}_{k|k}$, 后验误差协方差矩阵 (状态估计精度的度量)。

滤波器分两步工作:

- 一个时间更新 (预测) 步骤, 其中状态和协方差矩阵根据我们对系统动力学和误差特征的了解进行更新, 这些特征由 \mathbf{F} 和 \mathbf{Q} 矩阵建模。预测步骤不包括观察结果的影响。
- 一个测量更新 (校正) 步骤, 其中包括观测的影响, 以完善状态估计和协方差矩阵。这一步需要确定测量矩阵 \mathbf{H} 和 \mathbf{R} 矩阵。

预测和校正步骤的相应方程式如下: 其中符号 $\hat{\mathbf{x}}_{n|m}$ 表示在给定 n 时刻观测直至并包括 $m \leq n$ 时刻的 \mathbf{x} 的估计值。

预测		
1	预测 (先验) 状态估计	$\hat{\mathbf{x}}_{k k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1 k-1}$
2	预测 (先验) 估计协方差	$\mathbf{P}_{k k-1} = \mathbf{F}_k \mathbf{P}_{k-1 k-1} \mathbf{F}_k^T + \mathbf{Q}_k$

校正		
	新息或测量残差	$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k k-1}$
	新息 (或残差) 协方差	$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k k-1} \mathbf{H}_k^T + \mathbf{R}_k$
3	最佳卡尔曼增益	$\mathbf{K}_k = \mathbf{P}_{k k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$
4	更新 (后验) 的状态估计	$\hat{\mathbf{x}}_{k k} = \hat{\mathbf{x}}_{k k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$
5	更新 (后验) 的协方差	$\mathbf{P}_{k k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k k-1}$

卡尔曼滤波实现的主要任务是利用系统动力学模型和测量模型, 提出状态转换矩阵 \mathbf{F} 、测量矩阵 \mathbf{H} 和系统噪声特性, 设计过程和测量噪声协方差矩阵。

2.2 EKF 系统

在扩展卡尔曼滤波中，状态转移和观测模型不需要是状态的线性函数，而可以是可微函数。我们考虑非加性噪声的更一般形式的构想及方程

$$\begin{aligned}\mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{z}_k &= h(\mathbf{x}_k, \mathbf{v}_k)\end{aligned}$$

其中， \mathbf{w}_k 和 \mathbf{v}_k 是过程噪声和观测噪声，它们分别被假定为协方差为 \mathbf{Q}_k 和 \mathbf{R}_k 的零均值多元高斯噪声。 \mathbf{u}_k 是控制向量。

函数 $f()$ 可用于根据先前估计计算预测状态，类似地，函数 $h()$ 可用于根据预测状态计算预测测量。然而， $f()$ 和 $h()$ 不能直接应用于协方差，而是用其计算偏导数矩阵 (雅可比矩阵)。

在每个时间步骤，用当前预测状态来评估雅可比矩阵。这些雅可比矩阵可用于卡尔曼滤波方程。该过程基本上是将非线性函数在当前估计值周围使其线性化。

相比于标准 KF 系统，EKF 的方程则增加了本地线性化的处理，变化后的预测和校正步骤的相应方程式如下：

- 预测

- 预测 (先验) 状态估计

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k, \mathbf{w}_k) \quad (1)$$

- 预测 (先验) 估计协方差

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (2)$$

其中，状态转移矩阵 \mathbf{F}_k 为 Jacobian 矩阵

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k}$$

控制输入矩阵 \mathbf{G}_k 为 Jacobian 矩阵

$$\mathbf{G}_k = \left. \frac{\partial f}{\partial \mathbf{w}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k}$$

- 校正

- 新息或测量残差

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})$$

- 新息 (或残差) 协方差

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k$$

其中，观测矩阵 \mathbf{H}_k 为 Jacobian 矩阵

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}$$

- 最佳卡尔曼增益

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (3)$$

– 更新 (后验) 的状态估计

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (4)$$

– 更新 (后验) 的协方差

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (5)$$

对于 EKF, 新增的任务就是计算状态转换矩阵 \mathbf{F} 、控制输入矩阵 \mathbf{G} 和测量矩阵 \mathbf{H} 的 Jacobian 矩阵。

3 ECL EKF 所使用的传感器测量值

ECL EKF 具有不同的操作模式, 以允许不同的传感器测量组合。滤波器在启动时会检查传感器的最小可行组合, 并且在完成初始倾斜, 偏航和高度对准之后, 进入提供旋转, 垂直速度, 垂直位置, IMU 角度偏差和 IMU 速度偏差估计的模式。

此模式需要 IMU 数据, 偏航数据源 (磁力计或外部视觉) 和高度数据源。该数据集是所有 EKF 运行模式的最低需求数据。在此基础上可以使用其它传感器数据来估计额外的状态变量。

3.1 IMU

与机体固连的三轴惯性测量单元 (IMU), 以最小 100Hz 的频率获取增量角度和增量速度数据。EKF 仅将 IMU 数据用于状态预测。在 EKF 推导中, IMU 数据不作为观测值使用。

3.2 磁力计

需要以最小 5Hz 的速度的三轴机体固连磁力计数据 (或外部视觉系统姿势数据)。磁力计数据可以用于两种方式:

- 使用倾角估计和磁偏角将磁力计测量值转换为偏航角。然后将该偏航角用作 EKF 的观察值。该方法精度较低并且不允许学习机体坐标系场偏移, 但是它对于磁场异常和大的初置陀螺偏差更有鲁棒性。它是启动期间和在地面时的默认方法。
- XYZ 磁力计读数用作单独的观察值。该方法更精确并且允许学习机体坐标系场偏移, 但是它假设地球磁场环境只会缓慢变化, 并且当存在显着的外部磁场异常时表现较差。

3.3 高度

高度数据源 — 来自 GPS, 气压计, 测距仪或外部视觉设备, 需要最小频率为 5Hz。

如果不存在这些测量值, EKF 将无法启动。当检测到这些测量值时, EKF 将初始化状态并完成倾角和偏航对准。当倾角和偏航对准完成后, EKF 可以转换到其它操作模式, 从而可以使用其它传感器数据。

3.4 GPS

3.4.1 位置和速度测量

GPS 提供位置和速度测量。

3.4.2 偏航角测量

有一些全球定位系统接收器可用来提供一个偏航角测量，以取代磁力计数据的使用。在存在大型磁场异常的环境中工作时，或在高纬度地区，地球磁场具有很大的倾斜角时，这可能是一个重要的优势。

3.4.3 从 GPS 速度数据获取偏航角

EKF 在内部运行一个附加的多假设滤波器，它使用多个 3 参数状态 — 北/东向 (N/E) 的速度和偏航角 — 的扩展卡尔曼滤波器 (EKF)。然后使用高斯加和滤波器 (GSF) 合并这些偏航角的估计值。单个 3 参数状态的 EKF 使用了 IMU 和 GPS 水平速度数据 (加上可选的空速数据)，而不依赖于事先对偏航角或磁力计测量有任何了解。这里提供了一个对于主滤波器的偏航角备份，当起飞后导航丢失，表明磁力计的偏航估计值不好时，它被用于重置主 EKF 滤波器的 24 参数状态的中的偏航数据。

这也使得 ECL 能够在没有任何磁力计、或没有双天线 GPS 接收器的情况下运行，并提供偏航数据，只要起飞后能够进行某种水平的移动，偏航数据就变得可观测。一旦机体完成了足够的水平移动，使偏航角可观测，24 参数的主 EKF 将使其偏航角与 GSF 的估计值对准，并开始使用 GPS。

3.4.4 双 GPS 接收器

GPS 接收器提供的数据可以用基于所报告数据的精确度的加权算法混合 (如果两者都以相同的速度输出数据并使用相同的精确度，这样做效果最好)。如果来自接收器的数据丢失，该机制还提供了自动故障转移，(例如，它允许使用标准 GPS 作为更精确的 RTK 接收器的备份)。

3.5 测距仪

测距仪的对地距离被用于一个单状态滤波器以估计地形相对于高度基准的垂直位置。

如果在可用作零高度基准面的平面上操作，则 EKF 也可以直接使用测距仪数据估算高度。

3.6 空速

当存在空速传感器并且飞机类型不是旋翼时，将使用空速数据。

3.7 合成侧滑

固定翼平台可以利用假定的侧滑观测值为零来改进风速估计，也可以在没有空速传感器的情况下进行风速估计。

3.8 基于阻力比力的多旋翼风场估计

多旋翼平台可以利用沿 X 和 Y 机体轴的空速和阻力之间的关系来估计风速的北/东分量。

3.9 光流

光流传感器依赖测距仪，提供 xy 水平面的速度测量值。

3.10 外部视觉系统

外部视觉系统提供位置、速度和姿态测量。

3.11 测量值用途汇总

测量值	符号	单位	传感器	最小频率	用途
角速度	ω	rad/s	IMU	100Hz	预测
加速度	\mathbf{a}	m/s ²	IMU	100Hz	预测
地球磁场	\mathbf{M}_{meas}	gauss	磁力计	5Hz	校正
海拔高度	H_{meas}	m	GPS/气压计	5Hz	校正
距地高度	H_{meas}^{agl}	m	测距仪/外部视觉	5Hz	校正
速度测量	\mathbf{V}_{meas}	m/s	GPS/(光流 + 测距仪)/外部视觉	5Hz	校正
位置测量	\mathbf{P}_{meas}	m	GPS/外部视觉	5Hz	校正
偏航角	ψ_{meas}	rad	2-GPS/磁力计/外部视觉	5Hz	校正
磁偏角	$\psi^{DECLINATION}$	rad	GPS/用户设置	5Hz	校正
等效空速	V_{meas}^{TAS}	m/s	空速传感器	5Hz	校正
合成侧滑	beta	rad	虚拟		校正
阻力比力	drag		虚拟		校正

根据前面的描述可知：

- ECL EKF2 最小传感器集合为：IMU，偏航数据源和高度数据源。
- 机体的姿态变化由角速度驱动，速度变化由加速度驱动，这两者由 IMU 提供。
- 偏航测量值校正了在 xy 水平面上姿态，高度测量值校正了 z 轴上的位置。有了这两者才能执行最基本的自动导航功能。
- 磁力计是最适合校正姿态的传感器，也是最容易受到外界影响的传感器。气压计是最常见的测量高度的设备，因为成本低廉。
- GPS 是多面手，可以提供很多测量数据。所以如果不是成本受限，GPS 应该是无人机的标配设备。注意，GPS 的采样速率比较慢，在 ECL 系统中假设其最小频率为 2Hz。
- 光流设备和外部视觉设备是最时髦的设备。但是应用效果还有待观察。
- ECL EKF2 做了很多数据质量检测判断。因此外接传感器的种类越多，则容错率就越高。

4 ECL EKF 状态向量

4.1 状态向量

估计和控制库 (ECL) 使用扩展卡尔曼滤波算法 (EKF) 来处理传感器的测量信息，并提供如下状态向量的估计值：

- 四元数定义从北，东，地 (NED) 局部地球坐标系到 X, Y, Z 机体坐标系的旋转

- IMU 的速度 — 北, 东, 地 (NED)(m/s)
- IMU 的位置 — 北, 东, 地 (NED)(m)
- IMU 增量角度偏差估计 — X, Y, Z (rad)
- IMU 增量速度偏差估计 — X, Y, Z (m/s)
- 地球磁场分量 — 北, 东, 地 (NED)(gauss)
- 飞行器机体坐标系磁场偏差 — X, Y, Z (gauss)
- 风速 — 北, 东 (NE)(m/s)

24 个状态的状态向量定义为:

$$\mathbf{x} = \begin{bmatrix} q^T & \mathbf{V}_{NED}^T & \mathbf{P}_{NED}^T & \Delta\boldsymbol{\theta}_b^T & \Delta\mathbf{V}_b^T & \mathbf{M}_I^T & \mathbf{M}_B^T & \mathbf{V}_{wind}^T \end{bmatrix}^T$$

其中, 姿态四元数 q , 对外比较时称为 q_{ekf} , 用于定义 X, Y, Z 机体坐标系相对于 NED 导航坐标系的角位置, 我们用 \mathbf{q} 表示其向量部分,

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix}$$

\mathbf{V}_{NED} 表示机体在 NED 坐标系中的速度,

$$\mathbf{V}_{NED} = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}$$

\mathbf{P}_{NED} 表示机体在 NED 坐标系中的位置,

$$\mathbf{P}_{NED} = \begin{bmatrix} P_N \\ P_E \\ P_D \end{bmatrix}$$

$\Delta\boldsymbol{\theta}_b$ 表示在 X, Y, Z 机体坐标系的增量角度偏差 (bias),

$$\Delta\boldsymbol{\theta}_b = \begin{bmatrix} \Delta\theta_{b_X} \\ \Delta\theta_{b_Y} \\ \Delta\theta_{b_Z} \end{bmatrix}$$

$\Delta\mathbf{V}_b$ 表示在 X, Y, Z 机体坐标系的增量速度偏差 (bias),

$$\Delta\mathbf{V}_b = \begin{bmatrix} \Delta V_{b_X} \\ \Delta V_{b_Y} \\ \Delta V_{b_Z} \end{bmatrix}$$

\mathbf{M}_I 表示在 NED 坐标系的地球磁场向量, 简称地磁,

$$\mathbf{M}_I = \begin{bmatrix} M_N \\ M_E \\ M_D \end{bmatrix}$$

\mathbf{M}_B 表示在 X, Y, Z 机体坐标系的磁场偏差 (bias),

$$\mathbf{M}_B = \begin{bmatrix} M_X \\ M_Y \\ M_Z \end{bmatrix}$$

\mathbf{V}_{wind} 表示在 NED 坐标系的在北东 (NE) 方向上的风速,

$$\mathbf{V}_{wind} = \begin{bmatrix} V_{wind_N} \\ V_{wind_E} \end{bmatrix}$$

估计算法对外常用的数据输出有:

- 姿态输出。
- 速度输出。
- 位置输出。
- 风速输出。

注意: 输出的数据项与 EKF 状态向量中的数据不完全相同, 而是经过输出互补滤波器校正后的数据, 具体参见第 10 章的讨论。

4.2 旋转矩阵

从机体坐标系到导航坐标系的旋转矩阵根据姿态四元数 q 由以下方程给出:

$$\begin{aligned} [T]_B^N &= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 \cdot q_2 - q_0 \cdot q_3) & 2(q_1 \cdot q_3 + q_0 \cdot q_2) \\ 2(q_1 \cdot q_2 + q_0 \cdot q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 \cdot q_3 - q_0 \cdot q_1) \\ 2(q_1 \cdot q_3 - q_0 \cdot q_2) & 2(q_2 \cdot q_3 + q_0 \cdot q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1 \cdot q_2 - q_0 \cdot q_3) & 2(q_1 \cdot q_3 + q_0 \cdot q_2) \\ 2(q_1 \cdot q_2 + q_0 \cdot q_3) & 1 - 2q_1^2 - 2q_3^2 & 2(q_2 \cdot q_3 - q_0 \cdot q_1) \\ 2(q_1 \cdot q_3 - q_0 \cdot q_2) & 2(q_2 \cdot q_3 + q_0 \cdot q_1) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix} \end{aligned}$$

上面上下两个方程在数学上完全相等, 但是因为第二个方程在对角线元素上少两个变量, 因此在符号推导系统中产生更简单且更数值稳定的计算式, 因此 ECL EKF2 选择了第二个方程。

如果需要从导航坐标系到机体坐标系的旋转, 将使用转置, 并用 $[T]_N^B$ 表示:

$$[T]_N^B = ([T]_B^N)^T = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

5 ECL EKF 时间更新方程

IMU 传感器定时采样，采样间隔时间为 Δt ，获得角速度 ω 和加速度 \mathbf{a} 数据。所以我们得到增量角度的测量值为

$$\Delta \theta_{meas} = \omega \Delta t \quad (6)$$

还有增量速度的测量值为

$$\Delta \mathbf{V}_{meas} = \mathbf{a} \Delta t \quad (7)$$

已知陀螺仪的测量方差为 σ_{gyro}^2 ，加速计的测量方差为 σ_{accel}^2 。所以我们有增量角度的测量方差为

$$\sigma_{\Delta \theta}^2 = \sigma_{gyro}^2 \Delta t^2 \quad (8)$$

还有增量速度的测量方差为

$$\sigma_{\Delta \mathbf{V}}^2 = \sigma_{accel}^2 \Delta t^2 \quad (9)$$

根据 IMU 测量值 $\Delta \theta_{meas}$ 和增量角度偏差 $\Delta \theta_b$ 计算出真增量角度 $\Delta \theta_{truth}$ ：

$$\Delta \theta_{truth} = \Delta \theta_{meas} - \Delta \theta_b \quad (10)$$

增量四元数 Δq 定义了从第 k 时刻到第 $k+1$ 时刻四元数的旋转，使用小角度近似从真增量角度 $\Delta \theta_{truth}$ 计算 (惯性导航采用其它精确方法计算)：

$$\Delta q = \begin{bmatrix} 1 \\ \frac{\Delta \theta_{truth}}{2} \end{bmatrix} \quad (11)$$

从 k 时刻到 $k+1$ 时刻，四元数乘积规则用于将姿态四元数的状态向前旋转增量四元数 Δq ：

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_{k+1} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}_k \cdot \Delta q = \begin{bmatrix} q_0 \Delta q_0 - q_1 \Delta q_1 - q_2 \Delta q_2 - q_3 \Delta q_3 \\ q_1 \Delta q_0 + q_0 \Delta q_1 - q_3 \Delta q_2 + q_2 \Delta q_3 \\ q_2 \Delta q_0 + q_3 \Delta q_1 + q_0 \Delta q_2 - q_1 \Delta q_3 \\ q_3 \Delta q_0 - q_2 \Delta q_1 + q_1 \Delta q_2 + q_0 \Delta q_3 \end{bmatrix} \quad (12)$$

根据 IMU 测量值 $\Delta \mathbf{V}_{meas}$ 和增量速度偏差 $\Delta \mathbf{V}_b$ 计算出真增量速度 $\Delta \mathbf{V}_{truth}$ ：

$$\Delta \mathbf{V}_{truth} = \Delta \mathbf{V}_{meas} - \Delta \mathbf{V}_b \quad (13)$$

从 k 时刻到 $k+1$ 时刻，将真增量速度向量 $\Delta \mathbf{V}_{truth}$ 从机体坐标系旋转到地球坐标系，并减去重力，计算速度状态变化

$$\begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}_{k+1} = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}_k + [T]_B^N \cdot \Delta \mathbf{V}_{truth} + \mathbf{g}_D \cdot \Delta t \quad (14)$$

使用更精确的梯形积分方法更新位置状态

$$\begin{aligned}
(\mathbf{P}_{NED})_{k+1} &= (\mathbf{P}_{NED})_k + \frac{1}{2} ((\mathbf{V}_{NED})_{k+1} + (\mathbf{V}_{NED})_k) \cdot \Delta t \\
\begin{bmatrix} P_N \\ P_E \\ P_D \end{bmatrix}_{k+1} &= \begin{bmatrix} P_N \\ P_E \\ P_D \end{bmatrix}_k + \frac{1}{2} \left(\begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}_{k+1} + \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}_k \right) \cdot \Delta t
\end{aligned} \tag{15}$$

IMU 传感器的偏差、磁场和风速状态均采用静态过程模型：

$$\begin{bmatrix} \Delta \theta_b \\ \Delta V_b \\ M_I \\ M_B \\ V_{wind} \end{bmatrix}_{k+1} = \begin{bmatrix} \Delta \theta_b \\ \Delta V_b \\ M_I \\ M_B \\ V_{wind} \end{bmatrix}_k \tag{16}$$

自此，已经完成预测（先验）状态估计的计算。接下来要进行预测（先验）估计协方差的计算。方程 (1) 可表示为：

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k + \mathbf{w}_k$$

其中，控制向量 \mathbf{u} ，也称干扰 (disturbance) 向量，为

$$\mathbf{u} = \begin{bmatrix} \sigma_{\Delta \theta}^2 \\ \sigma_{\Delta V}^2 \end{bmatrix}$$

它是由惯性测量单元 (IMU) 的增量角度和增量速度的测量方差， $\sigma_{\Delta \theta}^2$ 和 $\sigma_{\Delta V}^2$ 所定义。在状态变量中消除偏差影响后，假设惯性解中的误差增长是由增量角度和增量速度中的“噪声”驱动的。这种方案是可行的，因为在状态方程中考虑了传感器偏差，将其消除后即为零偏差，并且因为该位姿估计系统是基于运动方程进行设计，所以惯性解中的误差增长，即状态方程中的控制项，则由 IMU 的“噪声”驱动。方程 (2) 中的矩阵 \mathbf{Q} 为测量方差元素构成的对角线矩阵：

$$\mathbf{Q}_{6 \times 6} = \text{diag} \left(\begin{bmatrix} \sigma_{\Delta \theta}^2 \\ \sigma_{\Delta V}^2 \end{bmatrix} \right)$$

并且 $\mathbf{G}\mathbf{Q}\mathbf{G}^T$ 又被称为状态误差矩阵。

根据前面推导的第 $k+1$ 时刻的状态向量方程 (12)、(14)、(15) 和 (16)，分别对第 k 时刻的状态 \mathbf{x}_k 求 Jacobian 可得状态转换矩阵 \mathbf{F} 、控制 (干扰) 影响矩阵 \mathbf{G} 。

状态转换矩阵 \mathbf{F} 是一个 24×24 的稀疏矩阵：

$$\begin{aligned}
F &= \frac{\partial \mathbf{x}_{k+1}}{\partial \mathbf{x}_k} \\
&= \begin{bmatrix}
\mathbf{F}_{q_k}^{q_{k+1}} & 0_{4 \times 3} & 0_{4 \times 3} & \mathbf{F}_{\Delta \theta_b^k}^{q_{k+1}} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 3} & 0_{4 \times 2} \\
\mathbf{F}_{q_k}^{V_{k+1}} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{F}_{\Delta V_b^k}^{V_{k+1}} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 2} \\
0_{3 \times 4} & \mathbf{F}_{V_k}^{P_{k+1}} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 2} \\
0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 2} \\
0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 2} \\
0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{I}_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 2} \\
0_{3 \times 4} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{I}_{3 \times 3} & 0_{3 \times 2} \\
0_{2 \times 4} & 0_{2 \times 3} & 0_{2 \times 3} & 0_{2 \times 3} & 0_{2 \times 3} & 0_{2 \times 3} & 0_{2 \times 3} & \mathbf{I}_{2 \times 2}
\end{bmatrix}
\end{aligned}$$

其中, $\mathbf{F}_{q_k}^{q_{k+1}}$ 、 $\mathbf{F}_{q_k}^{V_{k+1}}$ 、 $\mathbf{F}_{V_k}^{P_{k+1}}$ 、 $\mathbf{F}_{\Delta \theta_b^k}^{q_{k+1}}$ 、 $\mathbf{F}_{\Delta V_b^k}^{V_{k+1}}$ 这 5 个分块矩阵见下面的推导。

1. $\mathbf{F}_{q_k}^{q_{k+1}}$ 为关于四元数的状态转移矩阵, 即 $k+1$ 时刻的旋转相对于 k 时刻的 Jacobian, 大小为 4×4 , 根据方程 (11) 和 (12) 可得:

$$\begin{aligned}
\mathbf{F}_{q_k}^{q_{k+1}} &= \frac{\partial q_{k+1}}{\partial q_k} \\
&= \frac{\partial (q_k \cdot \Delta q)}{\partial q_k} \\
&= \frac{\partial \left(q_k + q_k \cdot \begin{bmatrix} 0 \\ \frac{\Delta \boldsymbol{\theta}_{truth}}{2} \end{bmatrix} \right)}{\partial q_k} \\
&= \mathbf{I}_{4 \times 4} + M \left(\frac{\Delta \boldsymbol{\theta}_{truth}}{2} \right)
\end{aligned}$$

其中, $M(\boldsymbol{\theta})$ 为 4×4 斜对称矩阵

$$M(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -\theta_x & -\theta_y & -\theta_z \\ \theta_x & 0 & \theta_z & -\theta_y \\ \theta_y & -\theta_z & 0 & \theta_x \\ \theta_z & \theta_y & -\theta_x & 0 \end{bmatrix}$$

该矩阵在四元数教材里又被称为 $\boldsymbol{\Omega}$ 矩阵, 出现在向量和四元数的乘积中

$$\boldsymbol{\Omega}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -\boldsymbol{\theta}^T \\ \boldsymbol{\theta} & -[\boldsymbol{\theta} \times] \end{bmatrix}$$

2. $\mathbf{F}_{q_k}^{V_{k+1}}$ 为速度关于四元数的状态转移矩阵, 即 $k+1$ 时刻的速度相对于 k 时刻的旋转的 Jaco-

bian , 大小为 3×4 , 根据方程 (14) 可得:

$$\begin{aligned}
\mathbf{F}_{q_k}^{V_{k+1}} &= \frac{\partial \left((\mathbf{V}_{NED})_k + [\mathbf{T}]_B^N \cdot \Delta \mathbf{V}_{truth} + \mathbf{g}_D \cdot \Delta t \right)}{\partial q_k} \\
&= \frac{\partial \left([\mathbf{T}]_B^N \cdot \Delta \mathbf{V}_{truth} \right)}{\partial q_k} \\
&= \frac{\partial \left(\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 \cdot q_2 - q_0 \cdot q_3) & 2(q_1 \cdot q_3 + q_0 \cdot q_2) \\ 2(q_1 \cdot q_2 + q_0 \cdot q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 \cdot q_3 - q_0 \cdot q_1) \\ 2(q_1 \cdot q_3 - q_0 \cdot q_2) & 2(q_2 \cdot q_3 + q_0 \cdot q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} \Delta V_X \\ \Delta V_Y \\ \Delta V_Z \end{bmatrix} \right)}{\partial q_k} \\
&= \frac{\partial \left(\begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) \Delta V_X + 2(q_1 \cdot q_2 - q_0 \cdot q_3) \Delta V_Y + 2(q_1 \cdot q_3 + q_0 \cdot q_2) \Delta V_Z \\ 2(q_1 \cdot q_2 + q_0 \cdot q_3) \Delta V_X + (q_0^2 - q_1^2 + q_2^2 - q_3^2) \Delta V_Y + 2(q_2 \cdot q_3 - q_0 \cdot q_1) \Delta V_Z \\ 2(q_1 \cdot q_3 - q_0 \cdot q_2) \Delta V_X + 2(q_2 \cdot q_3 + q_0 \cdot q_1) \Delta V_Y + (q_0^2 - q_1^2 - q_2^2 + q_3^2) \Delta V_Z \end{bmatrix} \right)}{\partial q_k} \\
&= 2 \begin{bmatrix} q_0 \Delta V_X - q_3 \Delta V_Y + q_2 \Delta V_Z & q_1 \Delta V_X + q_2 \Delta V_Y + q_3 \Delta V_Z & -q_2 \Delta V_X + q_1 \Delta V_Y + q_0 \Delta V_Z & -q_3 \Delta V_X - q_0 \Delta V_Y + q_1 \Delta V_Z \\ q_3 \Delta V_X + q_0 \Delta V_Y - q_1 \Delta V_Z & q_2 \Delta V_X - q_1 \Delta V_Y - q_0 \Delta V_Z & q_1 \Delta V_X + q_2 \Delta V_Y + q_3 \Delta V_Z & q_0 \Delta V_X - q_3 \Delta V_Y + q_2 \Delta V_Z \\ -q_2 \Delta V_X + q_1 \Delta V_Y + q_0 \Delta V_Z & q_3 \Delta V_X + q_0 \Delta V_Y - q_1 \Delta V_Z & -q_0 \Delta V_X + q_3 \Delta V_Y - q_2 \Delta V_Z & q_1 \Delta V_X + q_2 \Delta V_Y + q_3 \Delta V_Z \end{bmatrix} \\
&= 2 \begin{bmatrix} (\Phi_1 \cdot \Delta \mathbf{V}_{truth})^T \\ (\Phi_2 \cdot \Delta \mathbf{V}_{truth})^T \\ (\Phi_3 \cdot \Delta \mathbf{V}_{truth})^T \end{bmatrix}
\end{aligned}$$

其中,

$$\Phi_1 = \begin{bmatrix} q_0 & -q_3 & q_2 \\ q_1 & q_2 & q_3 \\ -q_2 & q_1 & q_0 \\ -q_3 & -q_0 & q_1 \end{bmatrix}, \quad \Phi_2 = \begin{bmatrix} q_3 & q_0 & -q_1 \\ q_2 & -q_1 & -q_0 \\ q_1 & q_2 & q_3 \\ q_0 & -q_3 & q_2 \end{bmatrix}, \quad \Phi_3 = \begin{bmatrix} -q_2 & q_1 & q_0 \\ q_3 & q_0 & -q_1 \\ -q_0 & q_3 & -q_2 \\ q_1 & q_2 & q_3 \end{bmatrix}$$

3. $\mathbf{F}_{V_k}^{P_{k+1}}$ 为位置关于速度的状态转移矩阵, 即 $k+1$ 时刻的位置相对于 k 时刻的速度的 Jacobian, 大小为 3×3 , 根据方程 (15) 可得:

$$\begin{aligned}
\mathbf{F}_{V_k}^{P_{k+1}} &= \frac{\partial ((\mathbf{P}_{NED})_k + (\mathbf{V}_{NED})_k \cdot \Delta t)}{\partial (\mathbf{V}_{NED})_k} \\
&= \mathbf{I}_{3 \times 3} \cdot \Delta t
\end{aligned}$$

4. $\mathbf{F}_{\Delta \theta_b^k}^{q_{k+1}}$ 为四元数关于增量角度偏差的状态转移矩阵, 即 $k+1$ 时刻的旋转相对于陀螺仪 k 时刻

增量角度偏差的 Jacobian , 大小为 4×3 , 根据方程 (10)、(11) 和 (12) 可得:

$$\begin{aligned}
 \mathbf{F}_{\Delta \theta_b^k}^{q_{k+1}} &= \frac{\partial (q_k \cdot \Delta q)}{\partial (\Delta \theta_b)_k} \\
 &= \frac{\partial \left(q_k + q_k \begin{bmatrix} 0 \\ \frac{\Delta \theta_{meas}}{2} \end{bmatrix} + q_k \begin{bmatrix} 0 \\ -\frac{(\Delta \theta_b)_k}{2} \end{bmatrix} \right)}{\partial (\Delta \theta_b)_k} \\
 &= \frac{\partial \left(-\frac{1}{2} \begin{bmatrix} -q_1 \Delta \theta_{bx} - q_2 \Delta \theta_{by} - q_3 \Delta \theta_{bz} \\ +q_0 \Delta \theta_{bx} - q_3 \Delta \theta_{by} + q_2 \Delta \theta_{bz} \\ +q_3 \Delta \theta_{bx} + q_0 \Delta \theta_{by} - q_1 \Delta \theta_{bz} \\ -q_2 \Delta \theta_{bx} + q_1 \Delta \theta_{by} + q_0 \Delta \theta_{bz} \end{bmatrix} \right)}{\partial (\Delta \theta_b)_k} \\
 &= -\frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ +q_0 & -q_3 & +q_2 \\ +q_3 & +q_0 & -q_1 \\ -q_2 & +q_1 & +q_0 \end{bmatrix} \\
 &= -\frac{1}{2} \Xi(q)
 \end{aligned}$$

矩阵 $\Xi(q)$ 出现在四元数与向量的乘法中

$$\Xi(q) = \begin{bmatrix} -\mathbf{q}^T \\ q_0 \mathbf{I}_{3 \times 3} + [\mathbf{q} \times] \end{bmatrix}$$

矩阵 Ξ 和 Ω 之间的关系为

$$\Omega(\theta) q = \Xi(q) \theta$$

5. $\mathbf{F}_{\Delta V_b^k}^{V_{k+1}}$ 为速度关于增量速度偏差的状态转移矩阵, 即 $k+1$ 时刻的速度相对于加速度计 k 时刻增量速度偏差的 Jacobian , 大小为 3×3 , 根据方程 (13) 和 (14) 可得:

$$\begin{aligned}
 \mathbf{F}_{\Delta V_b^k}^{V_{k+1}} &= \frac{\partial \left((\mathbf{V}_{NED})_k + [\mathbf{T}]_B^N \cdot \Delta \mathbf{V}_{truth} + \mathbf{g}_D \cdot \Delta t \right)}{\partial (\Delta \mathbf{V}_b)_k} \\
 &= \frac{\partial \left((\mathbf{V}_{NED})_k + [\mathbf{T}]_B^N \cdot (\Delta \mathbf{V}_{meas} - \Delta \mathbf{V}_b) + \mathbf{g}_D \cdot \Delta t \right)}{\partial (\Delta \mathbf{V}_b)_k} \\
 &= -[\mathbf{T}]_B^N
 \end{aligned}$$

控制 (干扰) 影响矩阵 \mathbf{G} 需要对增量角度和增量速度的测量值求偏导, 因为测量值中包含有噪声。它是一个 24×6 的稀疏矩阵:

$$\begin{aligned}
\mathbf{G} &= \frac{\partial \mathbf{x}_{k+1}}{\partial \mathbf{w}_k} \\
&= \frac{\partial \mathbf{x}_{k+1}}{\partial (\Delta \boldsymbol{\theta}_{meas}, \Delta \mathbf{V}_{meas})_k} \\
&= \begin{bmatrix} \mathbf{G}_{\Delta \theta_{meas}}^{q_{k+1}} & 0_{4 \times 3} \\ 0_{3 \times 3} & \mathbf{G}_{\Delta V_{meas}}^{V_{k+1}} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{2 \times 3} & 0_{2 \times 3} \end{bmatrix}
\end{aligned}$$

其中, $\mathbf{G}_{\Delta \theta_{meas}}^{q_{k+1}}$ 和 $\mathbf{G}_{\Delta V_{meas}}^{V_{k+1}}$ 这 2 个分块矩阵见下面的推导。

1. $\mathbf{G}_{\Delta \theta_{meas}}^{q_{k+1}}$ 为姿态四元数关于增量角度噪声的控制输入矩阵, 即 $k+1$ 时刻的旋转相对于陀螺仪 k 时刻增量角度噪声的协方差矩阵, 大小为 4×3 , 根据方程 (10)、(11) 和 (12) 可得:

$$\begin{aligned}
\mathbf{G}_{\Delta \theta_{meas}}^{q_{k+1}} &= \frac{\partial (q_k \cdot \Delta q)}{\partial (\Delta \boldsymbol{\theta}_{meas})_k} \\
&= \frac{\partial \left(q_k + q_k \begin{bmatrix} 0 \\ \frac{\Delta \boldsymbol{\theta}_{meas}}{2} \end{bmatrix} + q_k \begin{bmatrix} 0 \\ -\frac{(\Delta \boldsymbol{\theta}_b)_k}{2} \end{bmatrix} \right)}{\partial (\Delta \boldsymbol{\theta}_{meas})_k} \\
&= \frac{\partial \left(\frac{1}{2} \begin{bmatrix} -q_1 \Delta \theta_{meas_x} - q_2 \Delta \theta_{meas_y} - q_3 \Delta \theta_{meas_z} \\ +q_0 \Delta \theta_{meas_x} - q_3 \Delta \theta_{meas_y} + q_2 \Delta \theta_{meas_z} \\ +q_3 \Delta \theta_{meas_x} + q_0 \Delta \theta_{meas_y} - q_1 \Delta \theta_{meas_z} \\ -q_2 \Delta \theta_{meas_x} + q_1 \Delta \theta_{meas_y} + q_0 \Delta \theta_{meas_z} \end{bmatrix} \right)}{\partial (\Delta \boldsymbol{\theta}_{meas})_k} \\
&= \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ +q_0 & -q_3 & +q_2 \\ +q_3 & +q_0 & -q_1 \\ -q_2 & +q_1 & +q_0 \end{bmatrix} \\
&= \frac{1}{2} \boldsymbol{\Xi}(q)
\end{aligned}$$

2. $\mathbf{G}_{\Delta V_{meas}}^{V_{k+1}}$ 为速度关于增量速度噪声的控制输入矩阵, 即 $k+1$ 时刻的速度相对于加速度计 k 时刻增量速度噪声的协方差矩阵, 大小为 3×3 , 根据方程 (13) 和 (14) 可得:

$$\begin{aligned}
\mathbf{G}_{\Delta V_{meas}}^{V_{k+1}} &= \frac{\partial \left((\mathbf{V}_{NED})_k + [\mathbf{T}]_B^N \cdot \Delta \mathbf{V}_{truth} + \mathbf{g}_D \cdot \Delta t \right)}{\partial (\Delta \mathbf{V}_{meas})_k} \\
&= \frac{\partial \left((\mathbf{V}_{NED})_k + [\mathbf{T}]_B^N \cdot (\Delta \mathbf{V}_{meas} - \Delta \mathbf{V}_b) + \mathbf{g}_D \cdot \Delta t \right)}{\partial (\Delta \mathbf{V}_{meas})_k} \\
&= [\mathbf{T}]_B^N
\end{aligned}$$

最后, 为方程 (2) 中的协方差矩阵 \mathbf{P} 增加过程噪声 \mathbf{w} 。那些运动学的状态 (姿态 q 、速度 \mathbf{V}_{NED} 和位置 \mathbf{P}_{NED})，它们的误差增长分别由 IMU 噪声方差控制，已经在前面的方程中加入。对于其它静态过程模型的状态，则用静态过程模型构造这些状态的过程噪声方差对角线， 24×24 的过程噪声矩阵为

$$\mathbf{w} = \text{diag} \left(\begin{bmatrix} 0_{4 \times 1} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \\ \sigma_{\Delta\theta}^2 \\ \sigma_{\Delta V}^2 \\ \sigma_{M_I}^2 \\ \sigma_{M_B}^2 \\ \sigma_{V_{wind}}^2 \end{bmatrix} \right)$$

其中，增量角度偏差和增量速度偏差的过程噪声方差，同样采用增量角度的测量方差 $\sigma_{\Delta\theta}^2$ 和增量速度的测量方差 $\sigma_{\Delta V}^2$ 。地球磁场 M_I ，机体磁场偏差 M_B 和风速 V_{wind} 的过程噪声方差， $\sigma_{M_I}^2$ 、 $\sigma_{M_B}^2$ 和 $\sigma_{V_{wind}}^2$ ，可由用户自行设定。

至此，已经完成预测 (先验) 估计协方差的计算。

6 ECL EKF 测量更新方程

ECL EKF 算法最大的特点就是使用传感器组合进行状态估计。这样的好处是系统具有很高的容错性，不管发生什么异常，总是有方法进行数据校正。但是带来的问题就是算法复杂，代码分支繁多，很难把握，更难调试。

该算法的第二个特点是将 xy 水平面和 z 轴方向的高度分离估计。一个原因是因为 GPS 的高度数据误差太大，而高度数据源又很多，这样处理可以提升高度数据的精度，并且当 GPS 类的全局定位数据缺失时，高度数据是可信的，只是 xy 水平面定位进入惯性航位推算模式。

该算法的第三个特点是对所有的观测向量都是拆分为一维一维的数据进行校正。这是一种优化算法，通过使用更多的运算步骤换取使用更少的栈空间。这对于 MCU 这种内存紧张的环境有好处。

6.1 磁力计融合

磁力计更新分为两种情况：

1. 用三轴磁力计的数据作为三维观测 (3D Fusion)。
2. 直接将磁力计的数据转换为磁偏角作为一维观测 (Heading Fusion)。

第二种方法对于磁场异常和大的初置陀螺偏差更有鲁棒性。它是启动期间和在地面时的默认方法。第一种方法是通常使用的方法，它有更好的精度，但是更容易受到地球局部磁场畸变的影响，当存在显著的外部磁场异常时表现较差。

三轴磁场的更新根据是否有磁偏角的约束分为两个阶段。在初始化或差工况时，先进行无磁偏角约束的磁力计融合阶段，再进行有磁偏角约束的融合阶段；好工况时则相反。下面 2 个小节分别进行这两个阶段的方程推导。第 3 小节进行一维观测的方程推导。

6.1.1 磁力计测量融合方程

无磁偏角约束的三轴磁场的融合阶段，不考虑磁偏角 $\psi^{DECLINATION}$ ，也就是假设地球的地理南北极与地磁南北极重合。

由磁力计测量的机体磁场强度为 \mathbf{M}_{meas} 。已知磁力计的测量方差为 \mathbf{R}_{MAG} 。

预测的机体磁场强度为 \mathbf{M}_{pred} ，由状态向量中的地球磁场 \mathbf{M}_I 旋转到机体轴得到：

$$\mathbf{M}_{pred} = \left([T]_B^N\right)^T \mathbf{M}_I + \mathbf{M}_B$$

$$\begin{bmatrix} M_{pred_X} \\ M_{pred_Y} \\ M_{pred_Z} \end{bmatrix} = \left([T]_B^N\right)^T \begin{bmatrix} M_N \\ M_E \\ M_D \end{bmatrix} + \begin{bmatrix} M_X \\ M_Y \\ M_Z \end{bmatrix}$$

磁场观测矩阵 \mathbf{H}_{MAG} 是一个 3×24 的稀疏矩阵：

$$\mathbf{H}_{MAG} = \frac{\partial \mathbf{M}_{pred}}{\partial \mathbf{x}}$$

$$= \begin{bmatrix} \mathbf{H}_q & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{H}_{M_I} & \mathbf{H}_{M_B} & \mathbf{0}_{3 \times 2} \end{bmatrix}$$

其中， \mathbf{H}_q 、 \mathbf{H}_{M_I} 和 \mathbf{H}_{M_B} 这 3 个分块矩阵见下面的推导。

1. \mathbf{H}_q 为磁力计的观测相对于姿态四元数的观测矩阵，即磁场测量值相对于旋转的 Jacobian，大小为 3×4 ：

$$\mathbf{H}_q = \frac{\partial \mathbf{M}_{pred}}{\partial q}$$

$$= \frac{\partial \left(\left([T]_B^N\right)^T \mathbf{M}_I + \mathbf{M}_B \right)}{\partial q}$$

$$= \frac{\partial \left(\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} M_N \\ M_E \\ M_D \end{bmatrix} \right)}{\partial q}$$

$$= \frac{\partial \left(\begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) M_N + 2(q_1 q_2 + q_0 q_3) M_E + 2(q_1 q_3 - q_0 q_2) M_D \\ 2(q_1 q_2 - q_0 q_3) M_N + (q_0^2 - q_1^2 + q_2^2 - q_3^2) M_E + 2(q_2 q_3 + q_0 q_1) M_D \\ 2(q_1 q_3 + q_0 q_2) M_N + 2(q_2 q_3 - q_0 q_1) M_E + (q_0^2 - q_1^2 - q_2^2 + q_3^2) M_D \end{bmatrix} \right)}{\partial q}$$

$$= 2 \begin{bmatrix} q_0 M_N + q_3 M_E - q_2 M_D & q_1 M_N + q_2 M_E + q_3 M_D & -q_2 M_N + q_1 M_E - q_0 M_D & -q_3 M_N + q_0 M_E + q_1 M_D \\ -q_3 M_N + q_0 M_E + q_1 M_D & q_2 M_N - q_1 M_E + q_0 M_D & q_1 M_N + q_2 M_E + q_3 M_D & -q_0 M_N - q_3 M_E + q_2 M_D \\ q_2 M_N - q_1 M_E + q_0 M_D & q_3 M_N - q_0 M_E - q_1 M_D & q_0 M_N + q_3 M_E - q_2 M_D & q_1 M_N + q_2 M_E + q_3 M_D \end{bmatrix}$$

$$= 2 \begin{bmatrix} (\Phi_4 \cdot \mathbf{M}_I)^T \\ (\Phi_5 \cdot \mathbf{M}_I)^T \\ (\Phi_6 \cdot \mathbf{M}_I)^T \end{bmatrix}$$

其中，

$$\Phi_4 = \begin{bmatrix} q_0 & q_3 & -q_2 \\ q_1 & q_2 & q_3 \\ -q_2 & q_1 & -q_0 \\ -q_3 & q_0 & q_1 \end{bmatrix}, \quad \Phi_5 = \begin{bmatrix} -q_3 & q_0 & q_1 \\ q_2 & -q_1 & q_0 \\ q_1 & q_2 & q_3 \\ -q_0 & -q_3 & q_2 \end{bmatrix}, \quad \Phi_6 = \begin{bmatrix} q_2 & -q_1 & q_0 \\ q_3 & -q_0 & -q_1 \\ q_0 & q_3 & -q_2 \\ q_1 & q_2 & q_3 \end{bmatrix}$$

2. \mathbf{H}_{M_I} 为磁力计的观测相对于地球磁场向量的观测矩阵，即磁场测量值相对于地球磁场的 Jacobian，大小 3×3 为：

$$\begin{aligned}\mathbf{H}_{M_I} &= \frac{\partial \mathbf{M}_{pred}}{\partial \mathbf{M}_I} \\ &= \frac{\partial \left(\left([\mathbf{T}]_B^N \right)^T \mathbf{M}_I + \mathbf{M}_B \right)}{\partial \mathbf{M}_I} \\ &= \left([\mathbf{T}]_B^N \right)^T\end{aligned}$$

3. \mathbf{H}_{M_B} 为磁力计的观测相对于机体磁场偏差的观测矩阵，即磁场测量值相对于磁场偏差的 Jacobian，大小 3×3 为：

$$\begin{aligned}\mathbf{H}_{M_B} &= \frac{\partial \mathbf{M}_{pred}}{\partial \mathbf{M}_B} \\ &= \frac{\partial \left(\left([\mathbf{T}]_B^N \right)^T \mathbf{M}_I + \mathbf{M}_B \right)}{\partial \mathbf{M}_B} \\ &= \mathbf{I}_{3 \times 3}\end{aligned}$$

在实际的代码里，磁场观测矩阵 \mathbf{H}_{MAG} 实际上是分为 3 个一维观测矩阵进行观测：

$$\begin{aligned}\mathbf{H}_{MAG_x} &= \frac{\partial M_{pred_x}}{\partial \mathbf{x}} \\ \mathbf{H}_{MAG_y} &= \frac{\partial M_{pred_y}}{\partial \mathbf{x}} \\ \mathbf{H}_{MAG_z} &= \frac{\partial M_{pred_z}}{\partial \mathbf{x}}\end{aligned}$$

这是一种优化算法，使用更多的运算步骤换取使用更少的栈空间。这对于 MCU 这种内存紧张的环境有好处。

6.1.2 综合偏差测量的融合方程

有磁偏角约束的融合阶段，考虑事实上，地球的地理南北极与地磁南北极并不重合，也就是真北和磁北有一定夹角，这个夹角称为磁偏角 $\psi^{DECLINATION}$ 。因此如果增加了磁偏角的约束，则需要在融合完三轴磁力计测量值的基础上进行此部分的融合。这用于在没有绝对位置或速度测量的情况下保持正确的航向——例如使用光流时。这可以用来防止在航向观测不良期间，因地球磁场估计而产生不必要的偏航旋转。

磁偏角的测量值 $\psi_{meas}^{DECLINATION}$ ，有两个来源：一是由 GPS 提供（参见第 18.2 节的内容），二是由用户设定本地的磁偏角数据。磁偏角的噪声为 $R = \sigma_{decl}^2$ ，其中标准差 σ_{decl} 在代码中设定，好工况为 0.02，差工况为 0.5。

磁偏角的预测值 $\psi_{pred}^{DECLINATION}$ 为

$$\psi_{pred}^{DECLINATION} = \arctan \left(\frac{M_E}{M_N} \right)$$

磁偏角观测矩阵 $\mathbf{H}_{\psi_{pred}^{DECLINATION}}$ 是一个 1×24 的稀疏矩阵:

$$\begin{aligned}\mathbf{H}_{\psi_{pred}^{DECLINATION}} &= \frac{\partial \psi_{pred}^{DECLINATION}}{\partial \mathbf{x}} \\ &= \begin{bmatrix} 0_{1 \times 4} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \mathbf{H}_{M_I}^{\psi_{pred}^{DECLINATION}} & 0_{1 \times 3} & 0_{1 \times 2} \end{bmatrix}\end{aligned}$$

其中, $\mathbf{H}_{M_I}^{\psi_{pred}^{DECLINATION}}$ 是一个 1×3 的矩阵:

$$\begin{aligned}\mathbf{H}_{M_I}^{\psi_{pred}^{DECLINATION}} &= \frac{\partial \psi_{pred}^{DECLINATION}}{\partial \mathbf{M}_I} \\ &= \frac{\partial \left(\tan^{-1} \left(\frac{M_E}{M_N} \right) \right)}{\partial (M_N, M_E, M_D)} \\ &= \begin{bmatrix} \frac{\partial \left(\tan^{-1} \left(\frac{M_E}{M_N} \right) \right)}{\partial (M_N)} & \frac{\partial \left(\tan^{-1} \left(\frac{M_E}{M_N} \right) \right)}{\partial (M_E)} & 0 \end{bmatrix} \\ &= \begin{bmatrix} \frac{-M_E}{M_N^2 + M_E^2} & \frac{M_N}{M_N^2 + M_E^2} & 0 \end{bmatrix}\end{aligned}$$

在飞行期间磁力数据对准之后, 进行最优估计之后的磁偏角 $\psi^{DECLINATION}$ 成为磁偏角的三个数据源, 其它模块通过调用 `Ekf::getMagDeclination()` 函数获得当前最适合的磁偏角数据。

6.1.3 一维磁罗盘航向测量融合方程

航向角 (heading), 也称偏航角 (yaw) ψ , 其测量值可由磁力计测量值 \mathbf{M}_{meas} 计算出来。首先将机体磁场旋转到 NED 坐标系:

$$\begin{aligned}\mathbf{M}_{meas_I} &= [\mathbf{T}]_B^N (\mathbf{M}_{meas} - \mathbf{M}_B) \\ \begin{bmatrix} M_{meas_N} \\ M_{meas_E} \\ M_{meas_D} \end{bmatrix} &= [\mathbf{T}]_B^N \left(\begin{bmatrix} M_{meas_X} \\ M_{meas_Y} \\ M_{meas_Z} \end{bmatrix} - \begin{bmatrix} M_X \\ M_Y \\ M_Z \end{bmatrix} \right)\end{aligned}$$

则航向角的测量值 ψ_{meas} 为:

$$\psi_{meas} = -\arctan \left(\frac{M_{meas_E}}{M_{meas_N}} \right) + \psi^{DECLINATION}$$

其噪声 R_{yaw} 根据磁力计特性设定。

航向角的预测值 ψ_{pred} 可由旋转矩阵 $[\mathbf{T}]_B^N$, 也就是由状态向量 \mathbf{x} 提供的姿态四元数 q 计算出

来:

$$\begin{aligned}\psi_{pred} &= \begin{cases} \arctan \left(\frac{([T]_B^N)_{(1,0)}}{([T]_B^N)_{(0,0)}} \right) & \text{for yaw321} \\ \arctan \left(\frac{-([T]_B^N)_{(0,1)}}{([T]_B^N)_{(1,1)}} \right) & \text{for yaw312} \end{cases} \\ &= \begin{cases} \arctan \left(\frac{2(q_1 \cdot q_2 + q_0 \cdot q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2} \right) & \text{for yaw321} \\ \arctan \left(\frac{-2(q_1 \cdot q_2 - q_0 \cdot q_3)}{q_0^2 - q_1^2 + q_2^2 - q_3^2} \right) & \text{for yaw312} \end{cases}\end{aligned}$$

磁偏角的观测相对于状态向量 \mathbf{x} 的观测矩阵 \mathbf{H}_ψ ，即对应的观测矩阵 Jacobian，为 1×24 的稀疏矩阵:

$$\begin{aligned}\mathbf{H}_\psi &= \frac{\partial \psi_{pred}}{\partial \mathbf{x}} \\ &= \begin{bmatrix} \mathbf{H}_q^\psi & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 2} \end{bmatrix}\end{aligned}$$

其中， \mathbf{H}_q^ψ 为磁偏角的观测相对于姿态四元数 q 的观测矩阵，为 1×4 矩阵:

$$\mathbf{H}_q^\psi = \frac{\partial \psi_{pred}}{\partial q}$$

这个偏导展开太复杂，很难手工推导。使用符号推导系统生成的计算式如下:

$$\mathbf{H}_q^\psi = \begin{cases} \begin{bmatrix} \frac{2(-2q_0(q_0q_3+q_1q_2)+q_3(q_0^2+q_1^2-q_2^2-q_3^2))}{4(q_0q_3+q_1q_2)^2+(q_0^2+q_1^2-q_2^2-q_3^2)^2} \\ \frac{2(-2q_1(q_0q_3+q_1q_2)+q_2(q_0^2+q_1^2-q_2^2-q_3^2))}{4(q_0q_3+q_1q_2)^2+(q_0^2+q_1^2-q_2^2-q_3^2)^2} \\ \frac{2(q_1(q_0^2+q_1^2-q_2^2-q_3^2)+2q_2(q_0q_3+q_1q_2))}{4(q_0q_3+q_1q_2)^2+(q_0^2+q_1^2-q_2^2-q_3^2)^2} \\ \frac{2(q_0(q_0^2+q_1^2-q_2^2-q_3^2)+2q_3(q_0q_3+q_1q_2))}{4(q_0q_3+q_1q_2)^2+(q_0^2+q_1^2-q_2^2-q_3^2)^2} \end{bmatrix}^T & \text{for yaw321} \\ \begin{bmatrix} \frac{2(-2q_0(q_0q_3-q_1q_2)+q_3(q_0^2-q_1^2+q_2^2-q_3^2))}{4(q_0q_3-q_1q_2)^2+(q_0^2-q_1^2+q_2^2-q_3^2)^2} \\ \frac{2(2q_1(q_0q_3-q_1q_2)-q_2(q_0^2-q_1^2+q_2^2-q_3^2))}{4(q_0q_3-q_1q_2)^2+(q_0^2-q_1^2+q_2^2-q_3^2)^2} \\ \frac{2q_1(q_0^2-q_1^2+q_2^2-q_3^2)+4q_2(q_0q_3-q_1q_2)}{4(q_0q_3-q_1q_2)^2+(q_0^2-q_1^2+q_2^2-q_3^2)^2} \\ \frac{2(q_0(q_0^2-q_1^2+q_2^2-q_3^2)+2q_3(q_0q_3-q_1q_2))}{4(q_0q_3-q_1q_2)^2+(q_0^2-q_1^2+q_2^2-q_3^2)^2} \end{bmatrix}^T & \text{for yaw312} \end{cases}$$

简单的一维偏航角融合，虽然结果不太准确，但其受地球磁场畸变的影响较小。它的适用于俯仰角在 $-60^\circ < \theta < +60^\circ$ 之间的情况。

注意，当第 6.8 节的外部视觉复用该算法时，由其提供 R_{yaw} ，并且对于航向角的测量值 ψ_{meas} ，由其提供的姿态四元数 q_{ev} 计算出来，即此时的新息是两个姿态四元数， q_{ekf} 和 q_{ev} ，两者的偏航角差异值。

6.2 光流测量序列融合方程

光流传感器 (optical flow) 依赖于测距仪 (range finder), 它可以提供 xy 水平面的速度测量值, 并且已知其测量噪声 R_{LOS} 。光流传感器的观测量有:

- 视轴 (line of sight, LOS) 角速度测量 (相对于传感器坐标系), 从向下看的光流传感器测量, 以 rad/sec 为单位, 围绕 X 和 Y 传感器坐标系轴。这些速度是运动补偿的。
- 正视轴 X 速度是图像围绕 X 传感器轴的相对旋转, 在正的 Y 轴方向上产生地面相对速度。
- 正视轴 Y 速度是图像围绕 Y 传感器轴的相对旋转, 在负的 X 轴方向上产生地面相对速度。
- 距离 (range) 的测量与机体 Z 轴对准 (假设为平地模型)。

距离 (range) 为从摄像机焦点到传感器视场 (field of view, fov) 中心的距离, 由地形高差 (terrain height offset) 估计算法计算, 并且为描述方便, 测量视轴角速度 \mathbf{V}_{meas}^{LOS} 的计算都在第 7.2 节中描述。

为预测视轴角速度, 首先计算传感器坐标系中的相对速度 \mathbf{V}^B (更精确的算法见第 7.2 节的描述):

$$\mathbf{V}^B = \left([T]_B^N\right)^T \mathbf{V}_{NED}$$

$$\begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} = \left([T]_B^N\right)^T \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix}$$

将相对速度除以距离 (range) 得到相对于 X 和 Y 轴的预测视轴角速度 \mathbf{V}_{pred}^{LOS} (注: 这些是已经过机体角速度运动补偿的光流速度):

$$V_X^{LOS} = +V_Y/\text{range}$$

$$V_Y^{LOS} = -V_X/\text{range}$$

$$\mathbf{V}_{pred}^{LOS} = \begin{bmatrix} V_X^{LOS} \\ V_Y^{LOS} \end{bmatrix}$$

$$= \frac{1}{\text{range}} \begin{bmatrix} 2(q_1 q_2 - q_0 q_3) V_N + (q_0^2 - q_1^2 + q_2^2 - q_3^2) V_E + 2(q_2 q_3 + q_0 q_1) V_D \\ -((q_0^2 + q_1^2 - q_2^2 - q_3^2) V_N + 2(q_1 q_2 + q_0 q_3) V_E + 2(q_1 q_3 - q_0 q_2) V_D) \end{bmatrix}$$

视轴角速度的观测相对于状态向量 \mathbf{x} 的观测矩阵 \mathbf{H}_{LOS} , 即对应的观测矩阵 Jacobian, 为 2×24 的稀疏矩阵:

$$\mathbf{H}_{LOS} = \frac{\partial \mathbf{V}_{pred}^{LOS}}{\partial \mathbf{x}}$$

$$= \begin{bmatrix} \mathbf{H}_q^{V^{LOS}} & \mathbf{H}_V^{V^{LOS}} & 0_{2 \times 3} & 0_{2 \times 3} & 0_{2 \times 3} & 0_{2 \times 3} & 0_{2 \times 3} & 0_{2 \times 2} \end{bmatrix}$$

其中, $\mathbf{H}_q^{V^{LOS}}$ 和 $\mathbf{H}_V^{V^{LOS}}$ 这 2 个分块矩阵见下面的推导。

1. $\mathbf{H}_q^{V^{LOS}}$ 为视轴角速度的观测相对于姿态四元数的观测矩阵，即视轴角速度相对于旋转的 Jacobian，大小为 2×4 ：

$$\begin{aligned}\mathbf{H}_q^{V^{LOS}} &= \frac{\partial \mathbf{V}_{pred}^{LOS}}{\partial q} \\ &= \frac{\partial \left(\frac{1}{\text{range}} \begin{bmatrix} 2(q_1 q_2 - q_0 q_3) V_N + (q_0^2 - q_1^2 + q_2^2 - q_3^2) V_E + 2(q_2 q_3 + q_0 q_1) V_D \\ -((q_0^2 + q_1^2 - q_2^2 - q_3^2) V_N + 2(q_1 q_2 + q_0 q_3) V_E + 2(q_1 q_3 - q_0 q_2) V_D) \end{bmatrix} \right)}{\partial q} \\ &= \frac{2}{\text{range}} \begin{bmatrix} -q_3 V_N + q_0 V_E + q_1 V_D & q_2 V_N - q_1 V_E + q_0 V_D & q_1 V_N + q_2 V_E + q_3 V_D & -q_0 V_N - q_3 V_E + q_2 V_D \\ -(q_0 V_N + q_3 V_E - q_2 V_D) & -(q_1 V_N + q_2 V_E + q_3 V_D) & -(-q_2 V_N + q_1 V_E - q_0 V_D) & -(-q_3 V_N + q_0 V_E + q_1 V_D) \end{bmatrix} \\ &= \frac{2}{\text{range}} \begin{bmatrix} (\Phi_5 \cdot \mathbf{V}_{NED})^T \\ (-\Phi_4 \cdot \mathbf{V}_{NED})^T \end{bmatrix}\end{aligned}$$

其中，

$$\Phi_4 = \begin{bmatrix} q_0 & q_3 & -q_2 \\ q_1 & q_2 & q_3 \\ -q_2 & q_1 & -q_0 \\ -q_3 & q_0 & q_1 \end{bmatrix}, \quad \Phi_5 = \begin{bmatrix} -q_3 & q_0 & q_1 \\ q_2 & -q_1 & q_0 \\ q_1 & q_2 & q_3 \\ -q_0 & -q_3 & q_2 \end{bmatrix}$$

2. $\mathbf{H}_V^{V^{LOS}}$ 为视轴角速度的观测相对于 NED 坐标系下机体速度的观测矩阵，即视轴角速度相对于 NED 坐标系下机体速度的 Jacobian，大小为 2×3 ：

$$\begin{aligned}\mathbf{H}_V^{V^{LOS}} &= \frac{\partial \mathbf{V}_{pred}^{LOS}}{\partial \mathbf{V}_{NED}} \\ &= \frac{\partial \left(\frac{1}{\text{range}} \begin{bmatrix} 2(q_1 q_2 - q_0 q_3) V_N + (q_0^2 - q_1^2 + q_2^2 - q_3^2) V_E + 2(q_2 q_3 + q_0 q_1) V_D \\ -((q_0^2 + q_1^2 - q_2^2 - q_3^2) V_N + 2(q_1 q_2 + q_0 q_3) V_E + 2(q_1 q_3 - q_0 q_2) V_D) \end{bmatrix} \right)}{\partial \mathbf{V}_{NED}} \\ &= \frac{1}{\text{range}} \begin{bmatrix} 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ -(q_0^2 + q_1^2 - q_2^2 - q_3^2) & -2(q_1 q_2 + q_0 q_3) & -2(q_1 q_3 - q_0 q_2) \end{bmatrix}\end{aligned}$$

在实际的代码里，磁场观测矩阵 \mathbf{H}_{LOS} 实际上是分为 2 个一维观测矩阵进行观测：

$$\begin{aligned}H_{LOS_x} &= \frac{\partial V_X^{LOS}}{\partial \mathbf{x}} \\ H_{LOS_y} &= \frac{\partial V_Y^{LOS}}{\partial \mathbf{x}}\end{aligned}$$

这是一种优化算法，使用更多的运算步骤换取使用更少的栈空间。这对于 MCU 这种内存紧张的环境有好处。

6.3 GPS 融合方程

6.3.1 GPS 的偏航角融合方程

某些 GPS 设备设计有双天线，或者两个特殊 GPS 设备组成双天线，在这种条件下，GPS 设备可以提供偏航角测量值，因此可以进行双天线航向融合。该方法的一个重要的概念是移动基线 (Moving Baseline)，即在两个 GPS 天线之间的线。其可以提供两个测量值：基线偏航角 $\psi_{baseline}$

和基线相对于机体 x 轴的偏航偏移角 $\psi_{antenna}$ 。该融合方程与第 6.1.3 节对磁偏角观测的原理类似，是对偏航偏移角 $\psi_{antenna}$ 的观测融合。以下推导双天线偏航测量融合方程。

设置测量噪声方差 R_{yaw} ，目前复用磁力计的数值。

计算天线阵列的观测偏航角，将其从机体变换为天线的偏航测量值 ψ_{meas}

$$\psi_{meas} = \psi_{baseline} + \psi_{antenna}$$

在机体坐标系中定义天线向量 \mathbf{v}_B

$$\mathbf{v}_B = \begin{bmatrix} \cos(\psi_{antenna}) \\ \sin(\psi_{antenna}) \\ 0 \end{bmatrix}$$

将其旋转到地球坐标系

$$\begin{aligned} \mathbf{v}_I &= [\mathbf{T}]_B^N \cdot \mathbf{v}_B \\ \begin{bmatrix} v_N \\ v_E \\ v_D \end{bmatrix} &= [\mathbf{T}]_B^N \cdot \begin{bmatrix} \cos(\psi_{antenna}) \\ \sin(\psi_{antenna}) \\ 0 \end{bmatrix} \end{aligned}$$

根据投影计算预测天线偏航角 ψ_{pred}

$$\psi_{pred} = \arctan\left(\frac{v_E}{v_N}\right)$$

偏航偏移角的观测相对于状态向量 \mathbf{x} 的观测矩阵 \mathbf{H}_ψ ，即对应的观测矩阵 Jacobian，为 1×24 的稀疏矩阵：

$$\begin{aligned} \mathbf{H}_\psi &= \frac{\partial \psi_{pred}}{\partial \mathbf{x}} \\ &= \begin{bmatrix} \mathbf{H}_q^\psi & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 2} \end{bmatrix} \end{aligned}$$

其中， \mathbf{H}_q^ψ 为磁偏角的观测相对于姿态四元数 q 的观测矩阵，为 1×4 矩阵：

$$\mathbf{H}_q^\psi = \frac{\partial \psi_{pred}}{\partial q}$$

这个偏导展开太复杂，无法手工推导。具体可见代码中用符号推导系统生成的计算式。

6.3.2 GPS 速度和水平位置融合方程

GPS 的速度和位置都涉及状态的直接观测，因此观测模型比较简单。因为 GPS 在 z 轴方向上的数据误差比较大，而高度数据源比较多，所以对于位置向量只进行水平位置更新。

GPS 的观测相对于状态向量 \mathbf{x} 的观测矩阵 \mathbf{H}_{GPS} ，即对应的观测矩阵 Jacobian 比较简单，为 5×24 稀疏矩阵：

$$\mathbf{H}_{GPS} = \begin{bmatrix} \mathbf{0}_{3 \times 4} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{2 \times 4} & \mathbf{0}_{2 \times 3} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 2} \end{bmatrix}$$

并且已知 GPS 的测量噪声 \mathbf{R}_{GPS} ，由此可对状态向量 \mathbf{x} 中的速度和水平位置进行校正。

6.4 空速融合方程

对于固定翼平台，空速观测方程假设传感器测量相对于风场的速度 \mathbf{V}_{EAS} 为：

$$\mathbf{V}_{EAS} = \mathbf{V}_{NED} - \mathbf{V}_{wind}$$

$$\begin{bmatrix} V_{EAS_N} \\ V_{EAS_E} \\ V_{EAS_D} \end{bmatrix} = \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} - \begin{bmatrix} V_{wind_N} \\ V_{wind_E} \\ 0 \end{bmatrix}$$

则真实风速的预测值 V_{pred}^{TAS} 为：

$$V_{pred}^{TAS} = \sqrt{(V_{EAS_N}^2 + V_{EAS_E}^2 + V_{EAS_D}^2)}$$

当只使用风速估计而不使用合成侧滑时，真实风速的观测相对于状态向量 \mathbf{x} 的观测矩阵 \mathbf{H}_{TAS} ，即对应的观测矩阵 Jacobian，为 1×24 稀疏矩阵：

$$\begin{aligned} \mathbf{H}_{TAS} &= \frac{\partial V_{pred}^{TAS}}{\partial \mathbf{x}} \\ &= \frac{\partial \left((V_N - V_{wind_N})^2 + (V_E - V_{wind_E})^2 + V_D^2 \right)^{\frac{1}{2}}}{\partial \mathbf{x}} \\ &= \begin{bmatrix} 0_{1 \times 4} & \mathbf{H}_{V_{NED}}^{V_{pred}^{TAS}} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \mathbf{H}_{V_{wind}}^{V_{pred}^{TAS}} \end{bmatrix} \end{aligned}$$

其中， $\mathbf{H}_{V_{NED}}^{V_{pred}^{TAS}}$ 为真实风速相对于速度的 1×3 观测矩阵：

$$\mathbf{H}_{V_{NED}}^{V_{pred}^{TAS}} = \begin{bmatrix} \frac{V_N - V_{wind_N}}{\sqrt{(V_N - V_{wind_N})^2 + (V_E - V_{wind_E})^2 + V_D^2}} \\ \frac{V_E - V_{wind_E}}{\sqrt{(V_N - V_{wind_N})^2 + (V_E - V_{wind_E})^2 + V_D^2}} \\ \frac{V_D}{\sqrt{(V_N - V_{wind_N})^2 + (V_E - V_{wind_E})^2 + V_D^2}} \end{bmatrix}^T$$

$\mathbf{H}_{V_{wind}}^{V_{pred}^{TAS}}$ 为真实风速相对于风速的 1×2 观测矩阵：

$$\mathbf{H}_{V_{wind}}^{V_{pred}^{TAS}} = - \begin{bmatrix} \frac{V_N - V_{wind_N}}{\sqrt{(V_N - V_{wind_N})^2 + (V_E - V_{wind_E})^2 + V_D^2}} \\ \frac{V_E - V_{wind_E}}{\sqrt{(V_N - V_{wind_N})^2 + (V_E - V_{wind_E})^2 + V_D^2}} \end{bmatrix}^T$$

并且已知风速传感器的测量值 V_{meas}^{TAS} 及其测量噪声 R_{TAS} ，由此可对状态向量 \mathbf{x} 中的速度和风速进行校正。本节的方程可对比第 11.4.1 节“真实空速融合”的内容。

注意，因为风速的误差相当大，为减小其对状态向量 \mathbf{x} 中的速度的影响，正常情况下都将 $\mathbf{H}_{V_{NED}}^{V_{pred}^{TAS}} = \mathbf{0}$ ，直到 ECL 估计系统进入风场航位推算模式模式后，才用 $\mathbf{H}_{V_{NED}}^{V_{pred}^{TAS}}$ 对状态向量 \mathbf{x} 中的速度进行校正。具体见第 19 章“惯性航位推算”中的讨论。

6.5 合成侧滑融合方程

非固定翼平台可以利用假定的侧滑观测值为零来改进风速估计，也可以在没有空速传感器的情况下进行风速估计。本节的方程可对比第 11.4.2 节“合成侧滑融合”的内容。

首先将等效风速 \mathbf{V}_{EAS} 旋转到机体坐标系中得到机体的风速 \mathbf{V}_{wind_B} ：

$$\begin{aligned}\mathbf{V}_{wind_B} &= \left([T]_B^N\right)^T \mathbf{V}_{EAS} \\ &= \left([T]_B^N\right)^T (\mathbf{V}_{NED} - \mathbf{V}_{wind}) \\ \begin{bmatrix} V_{wind_X} \\ V_{wind_Y} \\ V_{wind_Z} \end{bmatrix} &= \left([T]_B^N\right)^T \left(\begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} - \begin{bmatrix} V_{wind_N} \\ V_{wind_E} \\ 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} V_N - V_{wind_N} \\ V_E - V_{wind_E} \\ V_D \end{bmatrix}\end{aligned}$$

用小角近似法计算预测侧滑角度，也是侧滑新息，因为假设观测值为零：

$$\text{beta} = \frac{V_{wind_Y}}{V_{wind_X}}$$

合成侧滑的噪声 R_{beta} 由用户设定。合成侧滑的观测相对于状态向量 \mathbf{x} 的观测矩阵 \mathbf{H}_{beta} ，即对应的观测矩阵 Jacobian，为 1×24 稀疏矩阵：

$$\begin{aligned}\mathbf{H}_{\text{beta}} &= \frac{\partial \left(\frac{V_{wind_Y}}{V_{wind_X}} \right)}{\partial \mathbf{x}} \\ &= \frac{\partial \left(\frac{2(q_1q_2 - q_0q_3)(V_N - V_{wind_N}) + (q_0^2 - q_1^2 + q_2^2 - q_3^2)(V_E - V_{wind_E}) + 2(q_2q_3 + q_0q_1)V_D}{(q_0^2 + q_1^2 - q_2^2 - q_3^2)(V_N - V_{wind_N}) + 2(q_1q_2 + q_0q_3)(V_E - V_{wind_E}) + 2(q_1q_3 - q_0q_2)V_D} \right)}{\partial \mathbf{x}} \\ &= \begin{bmatrix} \mathbf{H}_q^{\text{beta}} & \mathbf{H}_V^{\text{beta}} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & \mathbf{H}_{V_{wind}}^{\text{beta}} \end{bmatrix}\end{aligned}$$

其中， $\mathbf{H}_q^{\text{beta}}$ 为合成侧滑的观测相对于姿态四元数的观测矩阵，即合成侧滑相对于旋转的 Jacobian，大小为 1×4 ， $\mathbf{H}_V^{\text{beta}}$ 为合成侧滑的观测相对于 NED 坐标系下机体速度的观测矩阵，即合成侧滑相对于 NED 坐标系下机体速度的 Jacobian，大小为 1×3 ， $\mathbf{H}_{V_{wind}}^{\text{beta}}$ 为合成侧滑的观测相对于风速观测矩阵，大小为 1×3 。这 3 个偏导展开太复杂，就不在这里推导。具体可见代码中用符号推导系统生成的计算式。

注意，因为合成侧滑的误差相当大，为减小其对状态向量 \mathbf{x} 中的姿态和速度的影响，正常情况下都将 $\mathbf{H}_q^{\text{beta}} = \mathbf{0}$ ， $\mathbf{H}_V^{\text{beta}} = \mathbf{0}$ ，直到 ECL 估计系统进入风场航位推算模式模式后，才用 $\mathbf{H}_q^{\text{beta}}$ 和 $\mathbf{H}_V^{\text{beta}}$ 对状态向量 \mathbf{x} 中的姿态和速度进行校正。具体见第 19 章“惯性航位推算”的讨论。

6.6 阻力比力融合方程

与合成侧滑类似，首先将等效风速 \mathbf{V}_{EAS} 旋转到机体坐标系中得到机体的风速 \mathbf{V}_{wind_B} ：

$$\begin{aligned}\mathbf{V}_{wind_B} &= \left([T]_B^N\right)^T \mathbf{V}_{EAS} \\ &= \left([T]_B^N\right)^T (\mathbf{V}_{NED} - \mathbf{V}_{wind}) \\ \begin{bmatrix} V_{wind_x} \\ V_{wind_y} \\ V_{wind_z} \end{bmatrix} &= \left([T]_B^N\right)^T \left(\begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} - \begin{bmatrix} V_{wind_N} \\ V_{wind_E} \\ 0 \end{bmatrix} \right) \\ &= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} V_N - V_{wind_N} \\ V_E - V_{wind_E} \\ V_D \end{bmatrix}\end{aligned}$$

根据 IMU 测得的加速度 \mathbf{a}_{meas} 计算阻力比力 drag：

$$\text{drag} = \mathbf{a}_{meas} - \Delta \mathbf{V}_b / \Delta t$$

并用空气密度 ρ 和弹道系数 coef 估算空速 airspeed：

$$\text{airspeed} = \sqrt{(2|\text{drag}|) / (\text{coef} \cdot \rho)}$$

并计算比力系数 \mathbf{K}_{acc} ：

$$\mathbf{K}_{acc} = \text{coef} \cdot \text{airspeed}$$

预测的加速度 \mathbf{a}_{pred} 为：

$$\mathbf{a}_{pred} = -\frac{1}{2} \text{sign} \left(\begin{bmatrix} V_{wind_x} \\ V_{wind_y} \\ V_{wind_z} \end{bmatrix} \right) (\text{coef} \cdot \rho) \begin{bmatrix} V_{wind_x}^2 \\ V_{wind_y}^2 \\ V_{wind_z}^2 \end{bmatrix}$$

则阻力比力新息 drag_innov 为：

$$\text{drag_innov} = \mathbf{a}_{pred} - \text{drag}$$

阻力比力的观测相对于状态向量 \mathbf{x} 的观测矩阵 \mathbf{H}_{drag} ，即对应的观测矩阵 Jacobian，为 3×24 稀疏矩阵：

$$\begin{aligned}\mathbf{H}_{drag} &= \frac{\partial \mathbf{a}_{pred}}{\partial \mathbf{x}} \\ &= \begin{bmatrix} \mathbf{H}_q^{drag} & \mathbf{H}_V^{drag} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & \mathbf{H}_{V_{wind}}^{drag} \end{bmatrix}\end{aligned}$$

其中， \mathbf{H}_q^{drag} 为阻力比力的观测相对于姿态四元数的观测矩阵，即阻力比力相对于旋转的 Jacobian，大小为 3×4 ， \mathbf{H}_V^{drag} 为阻力比力的观测相对于 NED 坐标系下机体速度的观测矩阵，即阻力比力相对于 NED 坐标系下机体速度的 Jacobian，大小为 3×3 ， $\mathbf{H}_{V_{wind}}^{drag}$ 为阻力比力的观测相对于风速观测矩阵，大小为 3×2 。这 3 个偏导展开太复杂，就不在这里推导。具体可见代码中用符号推导系统生成的计算式。

6.7 高度融合方程

高度涉及状态的直接观测，因此观测模型比较简单。

高度数据源比较多，有 4 种：

1. 气压计 (baro)
2. GPS
3. 外部视觉 (EV)
4. 测距仪 (range)

同一时刻只能使用其中一种。软件内部会根据数据源的质量状况进行切换。

高度的观测相对于状态向量 \mathbf{x} 的观测矩阵 \mathbf{H}_{hgt} ，即对应的观测 Jacobian 矩阵，为 1×24 稀疏矩阵：

$$\mathbf{H}_{hgt} = \begin{bmatrix} 0_{1 \times 4} & 0_{1 \times 3} & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 2} \end{bmatrix}$$

根据各自传感器 z 轴的噪声 R_{hgt} ，就可以对高度进行校正。

高度融合方程虽然简单，但是因为涉及到 NED 坐标系和切换高度数据源的算法，实际的算法还是有些复杂。具体算法将在第 16 章讨论。

6.8 外部视觉融合方程

外部视觉系统可以提供速度 \mathbf{V}_{NED} 、位置 \mathbf{P}_{NED} 和姿态四元数的测量值 q_{ev} ，因此也可以提供航向角的测量值 ψ_{meas} 。需要注意的是，外部视觉的坐标系不一定是 NED，可能需要进行坐标系变换。速度与水平位置的融合方程与第 6.3.2 节的融合方程类似，高度则提供给第 6.7 节的融合方程，航向角则提供给第 6.1.3 节的融合方程，只是由其提供其自身的测量噪声 R_{yaw} 和姿态四元数 q_{ev} 。

6.9 辅助速度融合方程

某些飞行器具有红外锁定 (IR-Lock) 和 Pixy 视觉传感器，在这些飞行器着陆阶段，PX4 系统可以提供着陆目标的二维速度估计，该速度为着陆目标相对于机体的 x/y 方向速度。当着陆目标固定时，根据机体着陆姿态近水平面的假设，该速度向量取负值就是机体相对于大地的北/东方向速度 (V_N/V_E)。相应的融合方程与第 6.3.2 节的 GPS 速度融合方程类似。

6.10 虚拟位置融合方程

当没有一种水平面的位置和速度的测量值以校正用 IMU 数据进行的水平面位置推算时，则需要使用合成位置观测来防止惯性导航系统 (Inertial Navigation System, INS) 状态的无约束漂移。其方法是在最后一个已知位置进行虚拟位置测量，并且设置特别的位置方差，以限制漂移。为提高效率，虚拟位置测量最小融合频率为 5Hz。

相关的信息见第 19 章“惯性航位推算”中的讨论。

6.11 虚拟高度融合方程

与上一节类似，当没有一种高度测量值以校正用 IMU 数据进行的高度推算时，用于限制漂移的虚拟高度测量。如果我们没有任何辅助校正，则需要使用最后已知的垂直位置进行虚拟高度测量以限制漂移。相关的信息见第 19 章“惯性航位推算”中的讨论。

6.12 IMU 偏差的自校正

检查以上融合方程可以发现，没有一种传感器测量值直接校正增量角度偏差 $\Delta\theta_b$ 和增量速度偏差 ΔV_b 。这两者是通过校正后的姿态四元数 q 和速度 V_{NED} ，以及各传感器的噪声一起影响在 EKF 中计算出来的，这种方法被称为传感器的自校正。但是因为没有测量值对其进行直接校正，这两个偏差有可能发散，所以需要对其进行限制， $|\Delta\theta_b| < 20^\circ/s \cdot \Delta t$, $|\Delta V_b| < 0.4m/s^2 \cdot \Delta t$ 。

7 地形高差估计的 EKF 方程

因为光流的算法依赖测距仪提供的高度数据，所以 ECL 模块在原先维护的 24 元状态向量 EKF(24-EKF) 之外，还单独维护一个地形估计器，其目的是融合测距仪的测量值以估计地形垂直位置。地形高差 (terrain height offset) 估计的 EKF 方程在本章描述，注意其数值的正负符号，在 NED 坐标系中 z 轴向下为正，参见第 16 章的描述。

7.1 距地高度的估计

在地形估计器内部用一个标准 KF 维护一个地形高度 (terrain height) 状态值，又称地形垂直位置 (terrain vertical position, terrain_{vpos})，也称在向下轴的地形位置 (position of terrain in down axis, ptd)。相对应的，在 ECL/EKF2 中维护的垂直高度 P_D 为海平面高差 (altitude)，又称在向下轴飞行器的位置 (position of vehicle in down axis, pd)。

假设距地高度 (height above ground level, H^{agl}) 是一个固定模型，它的变化由噪声驱动，所以 $F = 1$ 。距地高度的测量值 H_{meas}^{agl} 由测距仪提供。另外我们有地形测量标准差 $\sigma_{terrain}$ ，并有地形梯度 $\nabla_{terrain}$ ，用于当位置变化时估计过程噪声。

初始化时设置

$$\text{terrain}_{vpos} = P_D + H_{meas}^{agl}$$

时间更新时，因为是固定模型，所以只需要计算状态方差 P

$$Q = (\sigma_{terrain} \cdot \Delta t)^2 + (\nabla_{terrain} \cdot \Delta t)^2 (V_N^2 + V_E^2)$$

$$P = P + Q$$

测量更新时，有两种数据融合，一是测距仪距地高度融合，二是光流速度融合。本节讨论前者，在下一节讨论后者。

测距仪距地高度融合。距地高度预测值 H_{pred}^{agl} 由以下方程计算 (注意在 NED 坐标系中 z 轴的正偏移将导致更低的距地高度)：

$$H_{pred}^{agl} = \text{terrain}_{vpos} - P_D$$

计算距地高度新息：

$$\tilde{y} = H_{pred}^{agl} - H_{meas}^{agl}$$

计算观测噪声有些复杂。我们从协方差矩阵 \mathbf{P} 的对角线相应位置得到 P_D 的状态方差 P_{P_D} ，设置其增益系数 G_{P_D} 。我们有测距仪测距仪噪声方差 σ_{range}^2 ，及其测量噪声增益 G_{range} ，以及测距仪测量值 $range_{meas}$ ，由此计算观测噪声 R

$$R = P_{P_D} \cdot G_{P_D} + \sigma_{range}^2 + G_{range} \cdot range_{meas}$$

计算新息方差

$$S = P + R$$

计算距地高度新息测试率

$$hagl_test_ratio = \tilde{y}^2 / (\text{gate}^2 \cdot S)$$

其中新息门限 $\text{gate} = 5$ 。当距地高度新息测试率 $hagl_test_ratio < 1$ 时，继续执行卡尔曼滤波的后面两个步骤，否则将该离群的测量值剔除。计算卡尔曼增益

$$K = P/S$$

更新状态值为

$$\text{terrain}_{vpos} = \text{terrain}_{vpos} - K \cdot \tilde{y}$$

更新状态不确定性方差

$$P = P - K \cdot P$$

由此可以得出最优估计的地形高度 terrain_{vpos} 。

7.2 光流速度融合

我们有从机体坐标系导航坐标系到的旋转矩阵 $[T]_B^N$ 及其逆矩阵 $[T]_N^B$ ，其第 3 行第 3 列数值用姿态四元数计算为 $R_{3,3} = q_0^2 - q_1^2 - q_2^2 + q_3^2$ 。根据工况好坏用互补滤波计算得出光流观测方差 R_{LOS} 。注意光流测量使用的符号约定：LOS 速率是在场景中围绕该轴的右手旋转为正。

光流传感器直接提供的测量向量有：

- $\Delta\theta_{xy}^{flow}$ ，围绕 X 和 Y 机体轴的图像的测量增量角度 (rad)，右手旋转为正。
- $\Delta\theta_{xyz}$ ，从速率陀螺测量中获得的惯性系统机体轴的测量增量角度 (rad)，右手旋转为正。其 xy 平面的数据为 $\Delta\theta_{xy}$ 。

计算用测量增量角度 $\Delta\theta_{meas}^{flow}$ 需补偿机体运动以计算 LOS 速率测量值

$$\Delta\theta_{meas}^{flow} = \Delta\theta_{xy}^{flow} - \Delta\theta_{xy}$$

首先计算 LOS 速率测量值。先计算运动补偿的陀螺偏差误差 $\Delta\omega_{bias}^{flow}$ 。我们有机体参考增量角 $\Delta\theta_{true}$ 和光流测量增量角 $\Delta\theta_{xyz}$ ，可计算出光流陀螺偏差误差 ω_{bias}^{flow}

$$\omega_{true} = \Delta\theta_{true} / \Delta t$$

$$\omega_{flow} = \Delta\theta_{xyz} / \Delta t$$

$$\omega_{bias}^{flow} = 0.99 \cdot \omega_{bias}^{flow} + 0.01 \cdot (\omega_{true} - \omega_{flow})$$

光流测量值去除机体旋转后, 图像围绕 X 和 Y 轴的测量增量角为 $\Delta\theta_{meas}^{flow}$, 即光流已在 xy 平面做补偿, 将其除以时间 Δt 可得光流速率。使用光流速率, 再去除机体角速率提供的, 用于校正进行运动补偿的陀螺偏差误差 ω_{bias}^{flow} , 可计算出光学 LOS 速率测量值 V_{meas}^{LOS} ,

$$V_{meas}^{LOS} = \Delta\theta_{meas}^{flow} / \Delta t + \omega_{bias}^{flow}$$

其次计算 LOS 速率预测值。先计算光流传感器在机体坐标系中的速度。我们有光流传感器从速率陀螺仪测量获得的惯性系统机体轴的测量增量角 $\Delta\theta_{xyz}$, 根据设置有光流传感器相对于 IMU 的偏移 ΔP^B , 由此计算传感器在机体坐标系中相对于 IMU 的速度, 注意, 惯性系的增量角 $\Delta\theta_{xyz}$ 在机体坐标系中需加负号

$$V_{flow \rightarrow IMU}^B = -\Delta\theta_{xyz} / \Delta t \otimes \Delta P^B$$

计算光流传感器在地球坐标系中的速度

$$V_{flow}^I = V_{NED} + [T]_B^N \cdot V_{flow \rightarrow IMU}^B$$

计算光流传感器在机体坐标系中的相对速度

$$V_{flow}^B = \begin{bmatrix} V_X \\ V_Y \\ V_Z \end{bmatrix} = [T]_N^B \cdot V_{flow}^I$$

距离 (range) 为从摄像机焦点到传感器视场 (field of view, fov) 中心的距离, 由以下方程计算:

$$\begin{aligned} \text{range} &= H^{agl} / R_{3,3} \\ &= (\text{terrain}_{vpos} - P_D) / (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{aligned}$$

将相对速度除以距离 (range) 得到相对于 X 和 Y 轴的预测视轴角速率 V_{pred}^{LOS} (注: 这些是已经过机体角速度运动补偿的光流速度):

$$\begin{aligned} V_X^{LOS} &= +V_Y / \text{range} \\ V_Y^{LOS} &= -V_X / \text{range} \\ V_{pred}^{LOS} &= \begin{bmatrix} V_X^{LOS} \\ V_Y^{LOS} \end{bmatrix} \end{aligned}$$

分别计算 X 和 Y 轴的观测 Jacobian 矩阵:

$$\begin{aligned} H_{flow}^{X,Y} &= \frac{\partial (V_X^{LOS}, V_Y^{LOS})}{\partial (\text{terrain}_{vpos})} \\ &= \frac{\partial \left(\frac{+V_Y (q_0^2 - q_1^2 - q_2^2 + q_3^2)}{(\text{terrain}_{vpos} - P_D)}, \frac{-V_X (q_0^2 - q_1^2 - q_2^2 + q_3^2)}{(\text{terrain}_{vpos} - P_D)} \right)}{\partial (\text{terrain}_{vpos})} \\ &= \left(\frac{+V_Y (q_0^2 - q_1^2 - q_2^2 + q_3^2)}{(\text{terrain}_{vpos} - P_D)^2}, \frac{-V_X (q_0^2 - q_1^2 - q_2^2 + q_3^2)}{(\text{terrain}_{vpos} - P_D)^2} \right) \end{aligned}$$

分别计算 X 和 Y 轴的新息及其协方差和卡尔曼增益, 两次校正地形高度 terrain_{vpos} , 剩下的步骤和上一节类似。由此可以得出最优估计的地形高度 terrain_{vpos} 。

8 使用 IMU 和 GPS 进行偏航对准

8.1 EKF-GSF 偏航估计器描述

该算法能在没有磁力计或外部偏航传感器的情况下运行，其目的是自动校正偏航误差引起的起飞后导航损失。它可以从不良磁偏航引起的起飞后导航故障中快速自动恢复。

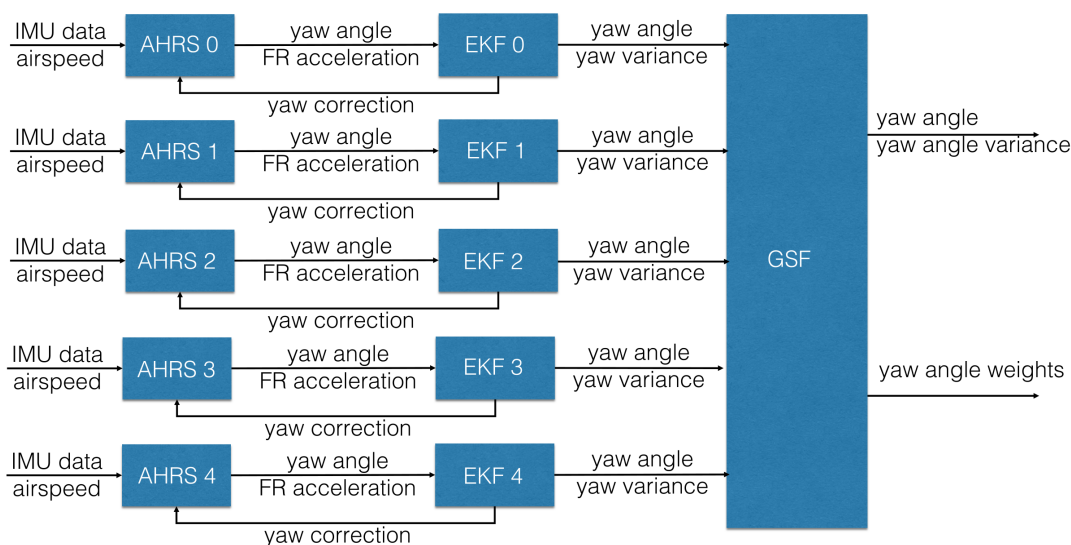
ECL 项目自 2018 年起进行的初步算法工作，以确定是否有可能仅使用 IMU 和 GPS 速度数据快速确定偏航角 (yaw)，而不依赖机体动力学的假设。最后确定使用多个 EKF 状态的高斯和滤波器 (Gaussian Sum Filter, GSF) 算法，即 EKF-GSF，因其为应用提供了最佳的性能/计算成本权衡。

EKF-GSF 算法也是在 24-EKF 之外又一个 EKF 实现，包括以下内容：

- 使用互补滤波器的 5 组 AHRS 的解
 - 这些计算预测偏航角和向前、向右加速度。
 - 空速 (测量或估计) 用于固定翼飞机飞行期间的向心加速度校正。
- 5 组 3 参数状态扩展卡尔曼滤波器
 - 状态为向北 (N)、向东 (E) 速度和偏航角。
 - 偏航角估算开始时的角度间隔相等， $\left[-4/5\pi \quad -2/5\pi \quad 0 \quad 2/5\pi \quad 4/5\pi \right]$ 。
 - GPS 的向北 (N)、向东 (E) 速度作为观测值。
- 高斯和滤波器 (Gaussian Sum Filter)
 - 根据标准化 GPS 的向北 (N)、向东 (E) 速度新息数值级别计算每个 EKF 的权重。总权重为 1。
 - 输出偏航角 ψ_{GSF} 估计值，这是单个 EKF 估计值的加权平均值。

GSF-EKF 过程框图如下：

EKF-GSF Yaw Estimator Diagram



该算法运行要求无人机在起飞和初始水平运动时必须处于不需要位置固定模式 (Position Control Mode)，让无人机水平移动一段距离。根据实验，无人机大约在 1s 左右移动 4m 左右的距离，EKF 将自动对准偏航并开始使用 GPS，从而启用位置固定模式。测试结果表明：

- GSF 权重在单个 EKF 滤波器之前收敛。
- 要求水平速度变化大于 GPS 速度不确定性。
- 速度变化越大，收敛速度越快。
- 由于失去导航，速度变化较大，导致在 1s 内收敛。

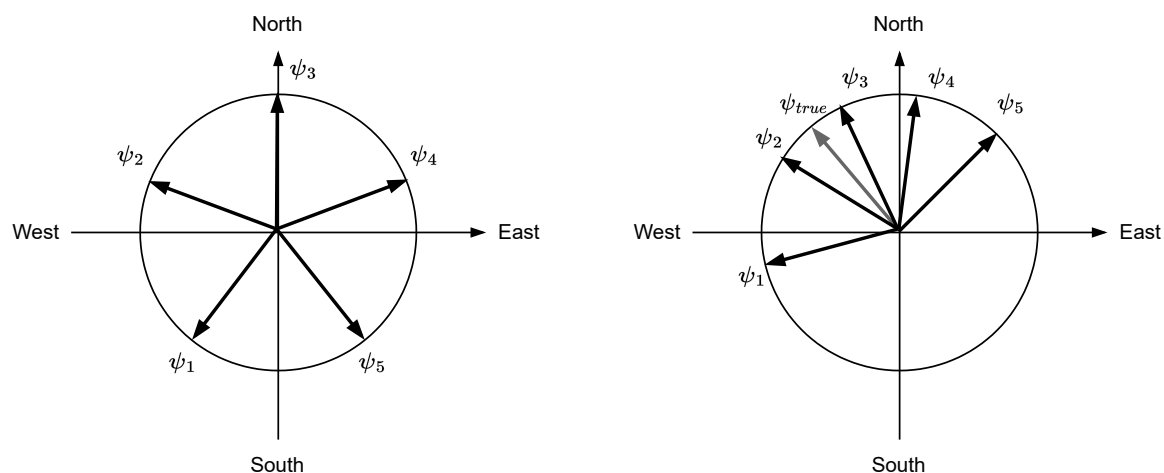
8.2 偏航估计器的算法直觉

EKF-GSF 算法是一个很新奇也很有创意的算法。不论是在 KF 方面还是在传感器方面，以及无人机的运动特性方面，设计者在技术方面和工程方面都已经炉火纯青、游刃有余。虽然该算法在数学方面还有说不清的地方，但是 KF 同样擅长在相关性领域工作，建模不一定要因果关系。

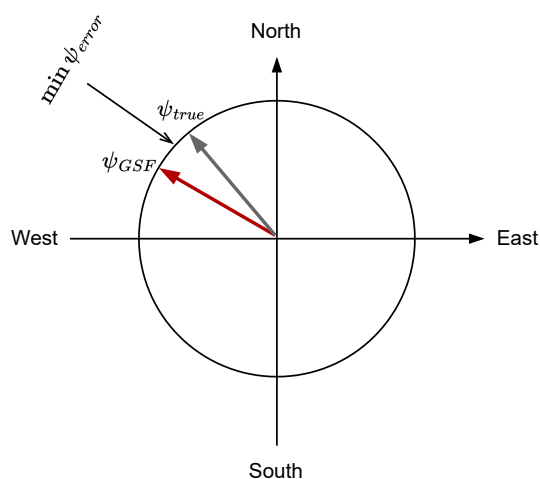
首先该算法一个假设就是偏航角由 xy 水平面的向北 (N)、向东 (E) 的增量速度和绕 z 轴旋转的增量角度所驱动，所以偏航角的测量误差也和这 3 个值的测量噪声相关

$$\psi_{error} \propto (\sigma_{\Delta V_x}^2, \sigma_{\Delta V_y}^2, \sigma_{\Delta \psi}^2)$$

这 3 个值和 3 个噪声由 IMU 提供。于是我们可以在 xy 水平面上预置 5 个偏航角 $\psi_i, i \in 1 \sim 5$ ，每个偏航角由这 3 个值和 3 个噪声驱动演化，独立进行时间更新。在测量更新阶段，我们观测 GPS 提供的 xy 水平面的向北 (N)、向东 (E) 的速度，因为残差的存在，这 5 个偏航角 ψ_i 会向真实的偏航角 ψ_{true} 聚拢，但是真实的偏航角 ψ_{true} 不可见。



不过我们可以通过高斯和的算法，计算这 5 个偏航角 ψ_i 的权重并相加，得到距离真实的偏航角 ψ_{true} 最近的复合偏航角 ψ_{GSF} 。

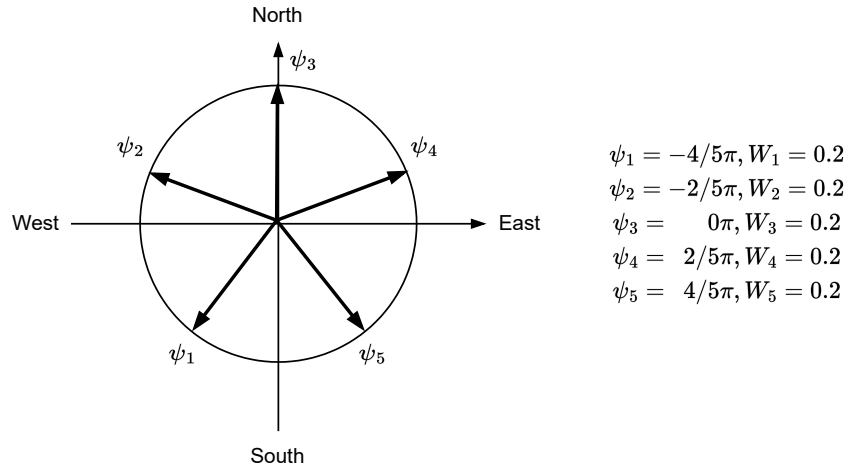


因此该算法就可以在没有磁力计的情况下估计偏航角，并且是水平速度变化越大，收敛越快。

8.3 偏航估计器的初始化与对准

8.3.1 偏航估计器的初始化

初始化的关键在于在 xy 平面上预制 5 个偏航角，平均间隔，平均权重。其它初始化都是常规操作。



8.3.2 AHRS 倾斜对准

就是如何从增量速度计算旋转矩阵。旋转矩阵直接从加速度测量中构造，对于所有模型都是相同的，因此只需计算一次。其假设是：

1. 偏航角为零 — 当速度融合开始时，每个模型的偏航角稍后对准。
2. 机体没有加速，因此所有测量的加速度都是由重力引起的。

用 Δt 表示传感器事件间隔时间。从主模块得到的加速度 \mathbf{a}^I (参见第 10.3.2 节) 计算得到增量速度 $\Delta \mathbf{V}^I$

$$\Delta \mathbf{V}^I = \mathbf{a}^I \cdot \Delta t$$

这是地球坐标系中的数据，需要转换到机体坐标系中并归一得到重力 D 轴方向的单位向量

$$\mathbf{D}^B = -\Delta \mathbf{V}^I / \|\Delta \mathbf{V}^I\|$$

计算地球坐标系向北 N 轴的单位向量，旋转为机体坐标系，与 \mathbf{D}^B 正交

$$\mathbf{N}^B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \mathbf{D}^B \cdot \left(\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{D}^B \right)$$

$$\mathbf{N}^B = \mathbf{N}^B / \|\mathbf{N}^B\|$$

计算地球坐标系向东 E 轴的单位向量，旋转为机体坐标系，与 \mathbf{D}^B 和 \mathbf{N}^B 正交

$$\mathbf{E}^B = \mathbf{D}^B \times \mathbf{N}^B$$

从地球坐标系到机体坐标系的旋转矩阵中的每一列表示旋转到机体坐标系的相应地球坐标系单位向量的投影，例如 \mathbf{N}^B 将是第一列。我们需要从机体坐标系到地球坐标系的旋转矩阵，因此旋转到机体坐标系的地球坐标系单位向量被复制到相应的行中

$$[T]_B^N = \begin{bmatrix} \mathbf{N}^B & \mathbf{E}^B & \mathbf{D}^B \end{bmatrix}^T$$

倾斜对准完成后即可对每一个模型进行时间更新。

8.3.3 AHRS 偏航角对准

其算法是根据最新的偏航角 ψ 更新欧拉角向量，进而生成新的姿态矩阵 $[T]_B^N$ 。在 24-EKF 算法之外，在 EKF-GSF 算法中为每一个模型用增益系数法维护一个独立的姿态矩阵 $[T]_B^N$ ，用于后面的算法中。

偏航角对准完成后可对每一个模型进行测量更新。

8.3.4 AHRS 预测

在每次时间更新前先从 IMU 数据生成姿态基准，即每一个模型的姿态矩阵 $[T]_B^N$ 被 IMU 数据驱动往前旋转了一个角度。所用数据为 IMU 数据及真实空速 (固定翼飞机)，并用加速度融合增益系数和陀螺仪偏差增益系数进行计算。最后计算得到一个校正后的增量角度，将其应用到姿态矩阵 $[T]_B^N$ 就得到新的姿态矩阵。

8.3.5 AHRS 更新

同样的，在每次测量更新之后，需要对姿态矩阵 $[T]_B^N$ 进行更新。因为增量偏航角 $\Delta\psi$ 发生在 xy 平面上，所以利用偏航旋转矩阵的稀疏性可以对姿态矩阵 $[T]_B^N$ 做优化更新。

8.4 偏航估计器的预测方程

令 ψ 表示机体坐标系相对于地球坐标系的偏航角 (yaw)。

令 \mathbf{V}_{NE} 表示机体在世界坐标系中的向北 (N) 和向东 (E) 的速度，

$$\mathbf{V}_{NE} = \begin{bmatrix} V_N \\ V_E \end{bmatrix}$$

令 $\Delta\psi$ 表示 IMU 的 z 轴在机体轴上的增量角度测量值，即增量偏航角的测量值。

令 $\sigma_{\Delta\psi}^2$ 表示 IMU 的 z 轴增量角度测量方差值。

令 $\Delta\mathbf{V}_{xy}$ 表示 IMU 的 x 轴和 y 轴在机体轴上的增量速度测量值

$$\Delta\mathbf{V}_{xy} = \begin{bmatrix} \Delta V_x \\ \Delta V_y \end{bmatrix}$$

令 $\sigma_{\Delta\mathbf{V}_{xy}}^2$ 表示 IMU 的 x 轴和 y 轴增量速度测量方差

$$\sigma_{\Delta\mathbf{V}_{xy}}^2 = \begin{bmatrix} \sigma_{\Delta V_x}^2 \\ \sigma_{\Delta V_y}^2 \end{bmatrix}$$

推导机体到导航方向的变换矩阵 (2D)

$$[T]_B^N = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}$$

姿态更新方程

$$\psi' = \psi + \Delta\psi$$

速度更新方程

$$\mathbf{V}_{NE}' = \mathbf{V}_{NE} + [T]_B^N \Delta\mathbf{V}_{xy}$$

定义状态向量

$$\hat{\mathbf{x}}_{k-1|k-1} = \begin{bmatrix} \mathbf{V}_{NE} \\ \psi \end{bmatrix}$$

定义过程方程的向量

$$\hat{\mathbf{x}}_{k|k-1} = \begin{bmatrix} \mathbf{V}'_{NE} \\ \psi' \end{bmatrix}$$

此即方程 (1)，预测 (先验) 状态估计。

计算状态转移矩阵

$$\begin{aligned} \mathbf{F} &= \frac{\partial \hat{\mathbf{x}}_{k|k-1}}{\partial \hat{\mathbf{x}}_{k-1|k-1}} \\ &= \frac{\partial \left(\begin{bmatrix} V_N + \cos(\psi) \Delta V_x - \sin(\psi) \Delta V_y \\ V_E + \sin(\psi) \Delta V_x + \cos(\psi) \Delta V_y \\ \psi + \Delta\psi \end{bmatrix} \right)}{\partial \left(\begin{bmatrix} V_N \\ V_E \\ \psi \end{bmatrix} \right)} \\ &= \begin{bmatrix} 1 & 0 & -\sin(\psi) \Delta V_x - \cos(\psi) \Delta V_y \\ 0 & 1 & \cos(\psi) \Delta V_x - \sin(\psi) \Delta V_y \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

定义控制 (干扰) 向量

$$\mathbf{u} = \begin{bmatrix} \Delta \mathbf{V}_{xy} \\ \Delta\psi \end{bmatrix}$$

推导协方差预测方程。在消除偏差效应后，假设惯性解中的误差增长由增量角度和增量速度中的“噪声”驱动。推导 IMU 噪声到状态噪声的 3×3 控制 (干扰) 影响矩阵

$$\begin{aligned} \mathbf{G} &= \frac{\partial \hat{\mathbf{x}}_{k|k-1}}{\partial \mathbf{u}} \\ &= \frac{\partial \left(\begin{bmatrix} V_N + \cos(\psi) \Delta V_x - \sin(\psi) \Delta V_y \\ V_E + \sin(\psi) \Delta V_x + \cos(\psi) \Delta V_y \\ \psi + \Delta\psi \end{bmatrix} \right)}{\partial \left(\begin{bmatrix} \Delta V_x \\ \Delta V_y \\ \Delta\psi \end{bmatrix} \right)} \\ &= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

这实际上是在 xy 平面上的旋转矩阵。

定义干扰 (disturbance) 矩阵，即 3×3 过程噪声矩阵

$$\mathbf{D} = \text{diag} \left(\begin{bmatrix} \sigma_{\Delta \mathbf{V}_{xy}}^2 \\ \sigma_{\Delta\psi}^2 \end{bmatrix} \right)$$

推导状态误差矩阵，即过程噪声协方差矩阵

$$\mathbf{Q} = \mathbf{G}\mathbf{D}\mathbf{G}^T$$

传播协方差矩阵

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q}$$

此即方程 (2)，预测 (先验) 估计协方差。

8.5 偏航估计器的观测方程

由 GPS 提供的向北 (N)、向东 (E) 速度观测值为

$$\mathbf{V}_{GPS} = \begin{bmatrix} V_{GPS_N} \\ V_{GPS_E} \end{bmatrix}$$

其方差为 $\sigma_{\Delta V_{GPS}}^2$ ，由此构建 2×2 测量噪声矩阵

$$\mathbf{R} = \text{diag} \left(\begin{bmatrix} \sigma_{\Delta V_{GPS}}^2 \\ \sigma_{\Delta V_{GPS}}^2 \end{bmatrix} \right)$$

因为是对速度的直接观测，所以测量矩阵比较简单

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

推导向北 (N)、向东 (E) 速度观测的协方差更新方程，即方程 (3)、方程 (4) 和方程 (5)

$$\begin{aligned} \tilde{\mathbf{y}} &= \mathbf{H}\hat{\mathbf{x}}_{k|k-1} - \mathbf{V}_{GPS} \\ \mathbf{S} &= \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R} \\ \mathbf{K} &= \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{S}^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} - \mathbf{K}\tilde{\mathbf{y}} \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}\mathbf{S}\mathbf{K}^T \end{aligned}$$

8.6 偏航估计器的 GSF 方程

对每个模型的状态计算高斯密度。首先计算马氏距离 (Mahalanobis distance)

$$D_M^2 = \tilde{\mathbf{y}}^T \mathbf{S}^{-1} \tilde{\mathbf{y}}$$

再计算 2D 正态分布 (Multivariate Normal Distribution) 的密度

$$\text{Density} = \left(\frac{1}{2} \right)^{p/2} \sqrt{\det(\mathbf{S}^{-1})} \exp \left(-\frac{D_M^2}{2} \right)$$

其中我们观测的是 2×1 速度向量，所以 $p = 2$ 。

计算 5 组 AHRS 的解的权重并归一

$$\begin{aligned} W'_i &= \text{Density}_i \cdot W_i \\ \mathbf{W} &= \mathbf{W}' / \|\mathbf{W}'\| \end{aligned}$$

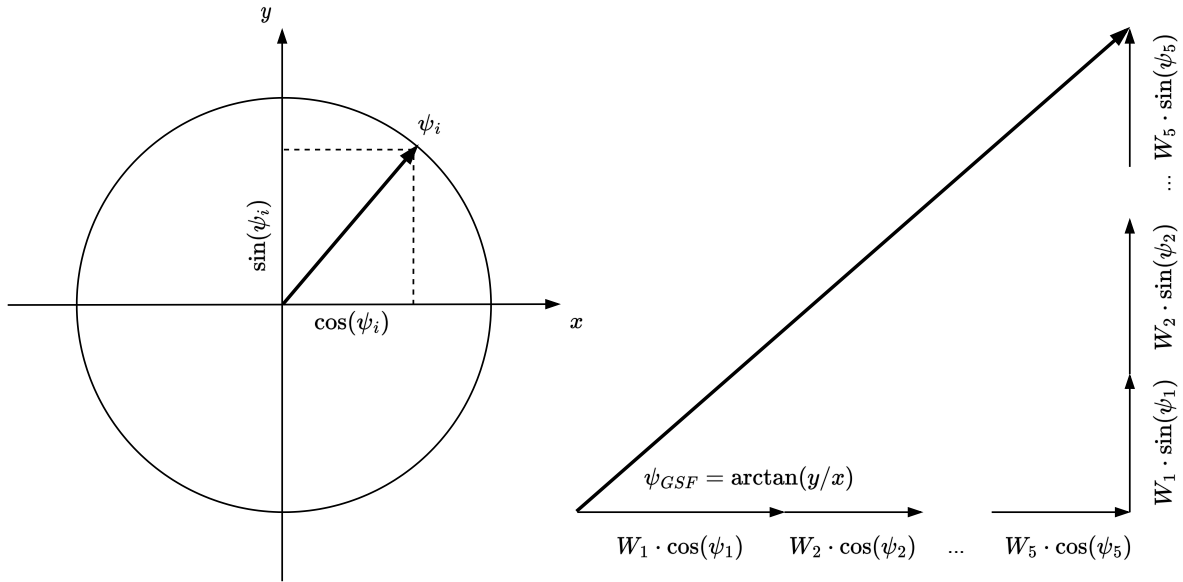
用每个模型的偏航状态的加权平均值以计算复合偏航向量。为避免角度回绕问题，在求和之前，将偏航状态转换为长度等于权重值为 1 的向量

$$x = \sum_{i=1}^5 W_i \cdot \cos(\psi_i)$$

$$y = \sum_{i=1}^5 W_i \cdot \sin(\psi_i)$$

$$\psi_{GSF} = \text{atan2}(y, x)$$

该算法原理在于将每一个偏航状态 ψ_i 看成是一个从原点出发的向量，因而在 x/y 轴上的投影通过权重缩放后相加，就得到单位圆上的一个向量，因而求出加权复合偏航角 ψ_{GSF} ，该值将在第 9 章“姿态的初始化值与对准”中使用。



根据每个模型方差的加权平均值计算偏航状态的复合方差。具有较大新息的模型的权重较小

$$\Delta\psi_i = \psi_i - \psi_{GSF}$$

$$\sigma_{\psi_{GSF}}^2 = \sum_{i=1}^5 W_i \cdot (P_i(3,3) + \Delta\psi_i^2)$$

其中 $P_i(3,3)$ 为第 i 组的协方差矩阵 P_i 中偏航角 ψ_i 的方差。

9 姿态的初始化值与对准

对于卡尔曼滤波器的状态向量的初始值，可以将其设置为任意值，经过多轮迭代，最终都将收敛到距离真实值最近的数值。但是，如果在初始化时将其设置为较为接近真实值的数值，则可以有效地减少迭代时间。这种设置，称之为对准。并且，当传感器受到外界的干扰，很容易造成估计值的无限制漂移。因此在 ECL 估计系统运行过程中，会使用几种不同原理的估计器相互校验，对于误差较大的估计器，将其重置（重新对准），以约束估计值的漂移。

姿态的初始化分为倾斜对准 (tilt alignment) 和偏航角对准 (yaw alignment) 两个阶段。其中偏航角对准的运行流程最为复杂，因为偏航角 ψ 的主要由磁力计和 GPS 进行校正，而这两种传感器很容易受到外界的干扰，磁力计的问题主要是地磁本身太弱、很容易被环境干扰，GPS 的问题主要是其信号很容易被山势、植被和建筑物阻挡和干扰，因此在干扰大的环境中，偏航角 ψ 经常需要重置 (重新对准)。

9.1 倾斜对准

因为能够校正 xy 水平面的传感器数据比较多，干扰也较少，所以当姿态四元数 q 设置好初值后，很快就可以迭代出较为精确的横滚角 ϕ 和俯仰角 θ ，因此倾斜对准只需要判断用三维旋转向量表示的姿态的 xy 水平面的方差和小于阈值，即可认为估计系统完成倾斜对准。其算法比较简单。

我们有表示姿态的单位四元数 q ，用 \mathbf{q} 表示其向量部分，

$$\begin{aligned} q &= \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} \\ &= \begin{bmatrix} \cos(\theta/2) \\ \mathbf{u} \sin(\theta/2) \end{bmatrix} \end{aligned}$$

相应有一个旋转向量 $\boldsymbol{\theta} = \mathbf{u}\theta$ ，其中 \mathbf{u} 为表示旋转轴的单位向量， θ 为旋转角度。

将姿态四元数 q 变换为旋转向量 $\boldsymbol{\theta}$ 的方程为

$$\begin{aligned} \theta &= 2 \arccos(q_0) \\ \boldsymbol{\theta} &= \frac{\theta}{\sin(\theta/2)} \mathbf{q} \end{aligned}$$

计算四元数到旋转向量的转移矩阵，为 3×4 矩阵

$$\begin{aligned} \mathbf{G} &= \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{q}} \\ &= \frac{\partial \left(\frac{\theta}{\sin(\theta/2)} \mathbf{q} \right)}{\partial \left(\begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} \right)} \end{aligned}$$

姿态四元数 q 的协方差矩阵 \mathbf{P}_q 为 24-EKF 的 \mathbf{P} 矩阵的左上角 4×4 部分。将协方差矩阵从四元数旋转到旋转向量为

$$\mathbf{P}_{rv} = \mathbf{G} \mathbf{P}_q \mathbf{G}^T$$

取出协方差矩阵 \mathbf{P}_{rv} 的对角线元素即为旋转向量 $x/y/z$ 三轴的方差向量 $\begin{bmatrix} \sigma_{rv_x}^2 & \sigma_{rv_y}^2 & \sigma_{rv_z}^2 \end{bmatrix}^T$ 。我们设置倾斜对准的标准差为 3° ，方差和小于此阈值即认为倾斜对准，判断方程为

$$\sigma_{rv_x}^2 + \sigma_{rv_y}^2 < \left(\frac{3^\circ}{180^\circ} \pi \right)^2$$

相对应的，我们也可以预置旋转向量的方差向量 $\begin{bmatrix} \sigma_{rv_x}^2 & \sigma_{rv_y}^2 & \sigma_{rv_z}^2 \end{bmatrix}^T$ ，并将其变换为姿态四元数 q 的协方差矩阵 P_q 的初始化值。将旋转向量 θ 变换为姿态四元数 q 的方程为

$$\theta = \|\theta\|$$

$$q = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \frac{\theta}{\|\theta\|} \sin\left(\frac{\theta}{2}\right) \end{bmatrix}$$

计算旋转向量到四元数的转移矩阵，为 4×3 矩阵

$$G = \frac{\partial q}{\partial \theta}$$

$$= \frac{\partial \left(\begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \frac{\theta}{\|\theta\|} \sin\left(\frac{\theta}{2}\right) \end{bmatrix} \right)}{\partial \theta}$$

旋转向量 θ 的协方差矩阵 P_{rv} 为

$$P_{rv} = \text{diag} \left(\begin{bmatrix} \sigma_{rv_x}^2 \\ \sigma_{rv_y}^2 \\ \sigma_{rv_z}^2 \end{bmatrix} \right)$$

将协方差矩阵旋转为四元数坐标

$$P_q = G P_{rv} G^T$$

设置姿态四元数 q 的算法比较简单。我们先设置姿态四元数 $q = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$ ，并假定飞行器开机时处于相对静止状态，判断为

$$0.8g < \|\mathbf{a}\| < 1.2g$$

$$15^\circ/s < \|\boldsymbol{\omega}\|$$

此时加速度 \mathbf{a} 的数值大多为重力加速度，将其分解就得到欧拉角中的横滚角 ϕ 和俯仰角 θ ，

$$\phi = \text{atan2}(-a_y, -a_z)$$

$$\theta = \text{asin}(a_x)$$

并假定偏航角 $\psi = 0$ ，将该欧拉角变换为四元数即可得到姿态初值。

9.2 应用 GPS 的偏航角对准

采用 GPS 强制重新校准偏航角，以与 GPS 的水平速度向量对准。它仅用于固定翼飞行器在发射或起飞后对准偏航角。它的使用条件比较严格，需要至少 5 m/s 的 GPS 水平速度，并且速度误差与速度之比小于 0.15 才能进行可靠对准。算法假设机体为近水平姿态。

计算 GPS 航线过地 (course over ground, COG) 角度

$$COG_{GPS} = \text{atan2}(V_{GPS_E}, V_{GPS_N})$$

计算 EKF 航线偏航角

$$COG_{EKF} = \text{atan2}(V_E, V_N)$$

检查 EKF 和 GPS 的航线过地角度是否一致

$$\psi_{Error} = COG_{GPS} - COG_{EKF}$$

如果角度不一致 ($\psi_{Error} > 0.5 \text{ rad}$), 并且 GPS 水平速度新息较大 (由第 6.3.2 节方程计算), 或者之前没有进行偏航对准, 则使用 GPS 地面航道校正偏航角。

计算新的偏航角估计有三种情况:

1. 当我们第一次飞行对准时, 我们可以假设最近的速度变化是由于向前起飞或发射造成的, 因此在姿态四元数中的惯性偏航角 ψ_q 和 GPS 地面航线的差异是由于偏航误差造成的。
2. 我们之前已经在飞行中对准了偏航, 并对风进行了估算, 因此设置了偏航值, 以使飞行器机头与风场相对 GPS 速度向量对准。
3. 我们没有风场的估计, 所以将偏航与 GPS 速度向量对准。

$$\psi = \begin{cases} \psi_q + \psi_{Error} & \text{first time} \\ \text{atan2}(V_{GPS_E} - V_{wind_E}, V_{GPS_N} - V_{wind_N}) & \text{have wind estimates} \\ \text{atan2}(V_{GPS_E}, V_{GPS_N}) & \text{no wind estimates} \end{cases}$$

使用组合的 EKF 和 GPS 的速度方差来计算对准后偏航误差的粗略估计

$$\begin{aligned} \sigma_{V_{Error}}^2 &= \text{accuracy}_{GPS_speed}^2 + \sigma_{V_N}^2 + \sigma_{V_E}^2 \\ V_{GPS} &= \sqrt{V_{GPS_E}^2 + V_{GPS_N}^2} \\ \sin(\psi_{Error}) &= \frac{\sqrt{\sigma_{V_{Error}}^2}}{V_{GPS}} \\ \sigma_{\psi}^2 &= (\arcsin(\sin(\psi_{Error})))^2 \end{aligned}$$

由此将新的偏航角 ψ 和偏航角方差 σ_{ψ}^2 应用于姿态四元数 q 及其协方差。接着使用最新的磁力计测量值 (机体坐标系中的向量) 重置磁场状态

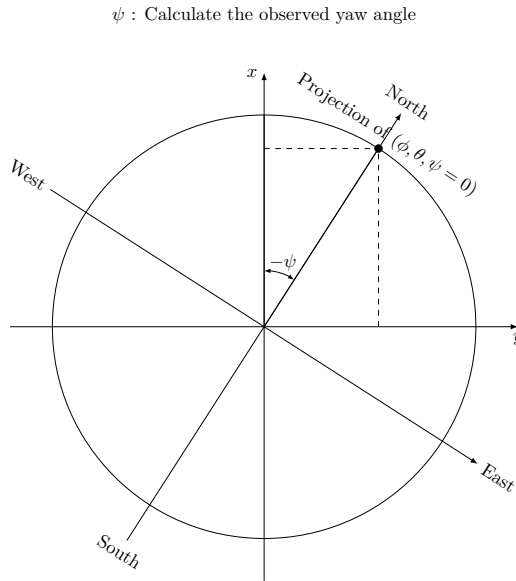
$$\begin{aligned} \mathbf{M}_I &= \mathbf{R}(q) \cdot \mathbf{M}_{meas} \\ \mathbf{M}_B &= \mathbf{0} \end{aligned}$$

9.3 应用磁力计的偏航角对准

在大多数室外情况下, 如果外部环境没有强磁干扰, 则应用磁力计的偏航角对准。室内 (IN-DOOR) 情况不适用该方法。

首先计算可观测的偏航角和偏航方差。因为已经倾斜对准, 姿态四元数 q 中的横滚角 ϕ 和俯仰角 θ 可信, 使用零偏航角 $\psi = 0$ 将磁力计测量值旋转到地球坐标系中, 投影到水平面上的角度给出了偏航角

$$\begin{aligned} \mathbf{M}_I &= \mathbf{R}(\phi, \theta, \psi = 0) \cdot \mathbf{M}_{meas} \\ \psi &= -\text{atan2}(M_{I_E}, M_{I_N}) + \psi^{DECLINATION} \end{aligned}$$



将新的偏航角 ψ 更新到姿态四元数 q 中即完成对准。接着使用最新的磁力计测量值 (机体坐标系中的向量) 重置磁场状态

$$\mathbf{M}_I = \mathbf{R}(q) \cdot \mathbf{M}_{meas}$$

最后用预置的地磁向量测量方差重置协方差矩阵中的相应行和列。

9.4 应用 EKF-GSF 的偏航角对准

EKF-GSF 算法 (参见第 8 章的内容) 可以在没有磁力计的情况下估计偏航角, 因此可以发现强磁干扰情况下的偏航角漂移。

判断条件很简单。当飞行器位于空中并且 GPS 信号无丢失时, 假定 EKF-GSF 偏航角 ψ_{GSF} 虽然不如 24-EKF 偏航角 ψ 精确, 但是距离不可见的真实偏航角 ψ_{true} 不会太远, 在第 6.3.2 节计算时检查两个偏航角的一致性, 当 $|\psi_{GSF} - \psi| > 25^\circ$ 时, 则认为 24-EKF 偏航角 ψ 已经被外界强磁场干扰发生漂移。这时则用 ψ_{GSF} 和 $\sigma_{\psi_{GSF}}^2$ 重置 24-EKF 的偏航角 ψ 及其协方差以限制其漂移。

10 输出互补滤波器

该算法使用当前时间范围内的最新 IMU 数据实现捷联惯导算法 (strapdown INS algorithm)。该算法缓冲惯性导航状态, 并计算延迟融合时间范围内 (delayed fusion time horizon) 与 EKF 状态的差异。根据差值计算增量角度、增量速度和速度校正, 并在当前时间范围内应用它们, 以便惯性导航状态在延迟融合时间范围内跟踪 EKF 状态。使用互补滤波器校正 EKF 中的时间延迟的灵感来源于参考文档 [12] 的工作。

从软件的角度看, 因为现在的系统越来越复杂, 原本在 RTOS 时代所做的假设越来越不能得到满足。现在传感器采样, 滤波器计算位姿, 用户获取位姿, 都分别处于不同的线程中, 有不同的工作频率。众多传感器各自工作在不同的频率, 异步采样, 数据异步到达。滤波器有自己的工作频率, 上层软件也有另外的工作频率。所有这些, 都会有一个时间差, 因此在用户获取当前位姿信息的瞬间, 滤波器需要对当前所维护的信息做一个补偿, 这就是输出互补滤波器。它的工作原理就是

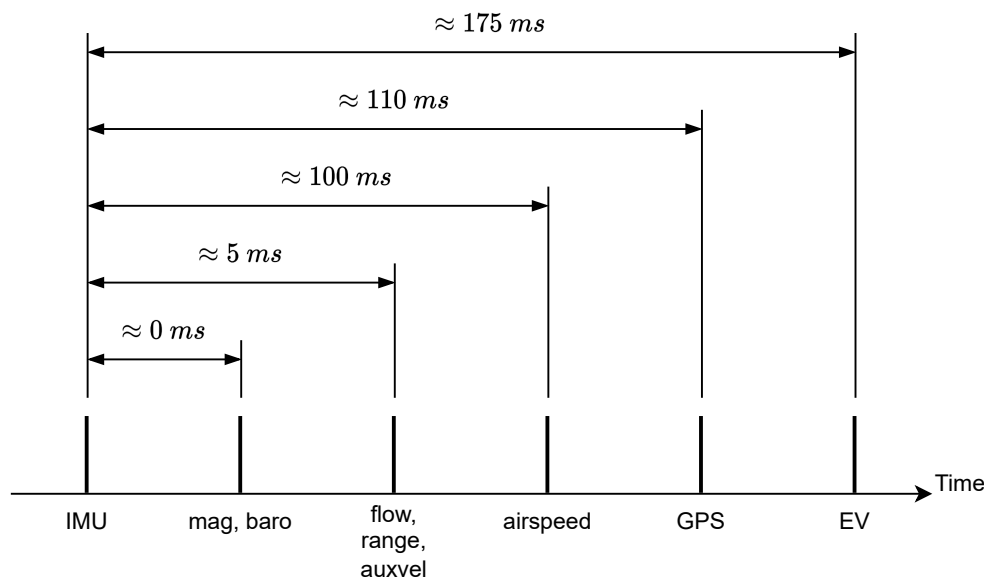
维护一个历史窗口，从历史数据中计算互补参数，然后对输出信息进行校正，具体的原理请看参考文档 [12]。在这里没有什么复杂的计算公式，一行的互补公式较多且大多简单细碎，经验参数众多，有很多工程细节，依赖专业背景和工程经验方可调参。下面简单描述该算法的实现。

10.1 融合时间范围

在传统的姿态估计的应用场景中，为简化计算并提高估计精度，有一个并行采样的假设。即所有的传感器数据都是同时采样。在高端设备上，并行采样是通过设计硬件电路来保证的。在低成本方案里，实际上执行的是串行采样，陀螺仪/加速计/地磁计这三者串行采样，有可能前后采样的时间差有 1 ms 左右。如果对估计精度不讲究，把它们当成并行采样处理也是可以的。那些低成本的无人机的姿态估计模块也都是这么做的。

此外，还有一个数据采样周期不一致的问题会影响估计精度。在低成本方案里，陀螺仪/加速计可以达到 $100 \sim 200\text{ Hz}$ 的采样率，而地磁计的采样率可能只有 10 Hz ，而 GPS 可能只有 2 Hz 。但是嵌入式软件系统为了简化设计，往往只围绕一种采样率进行设计。如果陀螺仪/加速计/地磁计这三者都以 100 Hz 进行采样，因地磁计来不及锁定数据，可能连续 10 次上报同一个数据，直到寄存器锁定后才上报新的数据。这样的数据组合，既会影响估计精度，也会增加系统负荷。

因此，为提高估计精度，必须处理多速率多延迟的向量测量这种情况。多延迟如下图所示，如果同时对所有传感器发出采样指令，IMU 是反应最灵敏的，可以认为是立即返回，其它传感器相对于 IMU 都有不同程度的延迟。



由此可以看出，像 GPS 或外部视觉这类复杂传感器反应都很慢。另外，从传感器与测量值关系列表中也可以看出，每种传感器的采样速率也不一样：

传感器	测量值	调用方法	最小频率	用途
IMU	ω, \mathbf{a}	setIMUData	100Hz	预测
磁力计	\mathbf{M}_{meas}	setMagData	5Hz	校正
GPS	$\mathbf{P}_{meas}, \mathbf{V}_{meas}$ $\psi^{DECLINATION}$	setGpsData	2Hz	校正
2-GPS	ψ_{meas}			
气压计	H_{meas}	setBaroData	5Hz	校正
空速传感器	V_{meas}^{TAS}	setAirspeedData	5Hz	校正
测距仪	H_{meas}^{agl}	setRangeData	5Hz	校正
光流	$\Delta\theta_{xy}^{flow}, \Delta\theta_{xyz}$	setOpticalFlowData	5Hz	校正
外部视觉	$\mathbf{P}_{meas}, \mathbf{V}_{meas}, q_{ev}$	setExtVisionData	5Hz	校正
着陆速度	\mathbf{V}_{xy}	setAuxVelData	5Hz	校正
虚拟	beta	内部调用		校正
虚拟	drag	内部调用		校正

甚至在当前的复杂系统中，采样时间点也不可能同步，而 ECL EKF 估计系统的运行频率与这些传感器不一样也不同步。ECL EKF 估计系统的设计主要考虑两种频率：IMU 的采样频率和 EKF 的运行频率。其中，IMU 的常见采样频率为 100 ~ 250 Hz，而 EKF 的运行频率在理想情况下应该是 IMU 采样频率的整数倍，为 50 ~ 200 Hz。我们将其称之为延迟融合时间范围 (delayed fusion time horizon)。为简化设计，在实现中两者都采用 100 Hz 的设计。

因为增量角度和增量速度依赖时间间隔 dt 的积分，所以 IMU 的 dt_{imu} 的精度影响滤波器的估计精度。如果 IMU 的名义采样率为 100 Hz，因为系统的各种原因，实际采样率可能在 98 ~ 102 Hz 之间抖动。所以在代码中 dt_{imu} 的数值不能写死，而是采用采样事件之间的时间戳差值，并使用低通滤波计算：

$$dt_{imu_avg} = \alpha \cdot dt_{imu_avg} + (1 - \alpha) dt_{imu_diff}$$

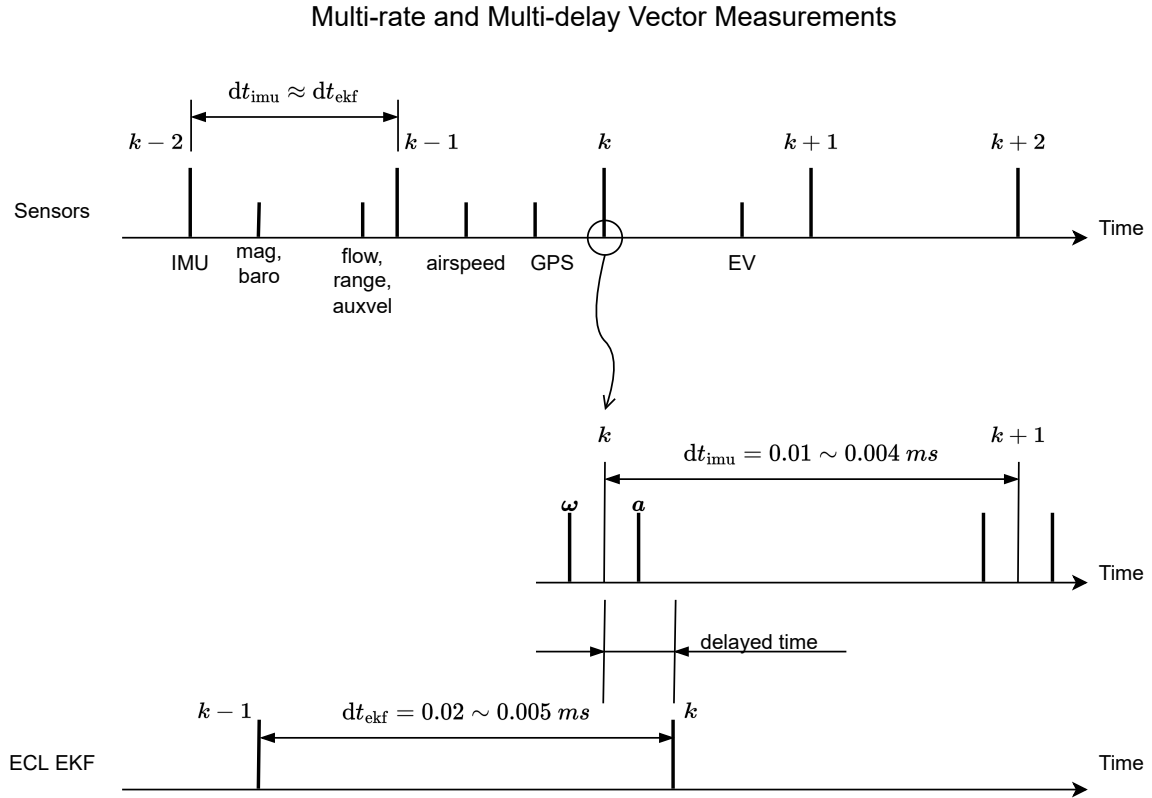
其中， $\alpha = 0.8$ ，这意味着我们假定采样时钟比较稳定，偶尔才会出现抖动。

同样的，ECL 的运行频率， dt_{ekf} ，也需要使用低通滤波计算：

$$dt_{ekf_avg} = \alpha \cdot dt_{ekf_avg} + (1 - \alpha) dt_{ekf_diff}$$

其中， $\alpha = 0.99$ ，这意味着我们假定系统时钟很稳定，偶尔才会出现抖动。

各传感器采样率与 EKF 的运行频率的可能的组合如下图所示：



在实现中，ECL EKF 估计系统的运行线程会被各种传感器多次唤醒。如果我们对输入系统以 IMU 周期为栅格进行考察，其它传感器测量值将随机落到不同的栅格中。因为各种传感器采样速率差别巨大，实际上有很多栅格里并没有校正传感器测量值到达，也就是在这段时间内，位姿只有预测而没有校正。如果滤波器以预测-校正为一个完整迭代周期，则因为采样率与采样时间点的差异，以及 ECL EKF 估计系统运行的延迟，最终所实现的滤波系统为在一段时间内估计精度可接受的近似最优估计系统。

10.2 互补滤波基本原理

因为多速率多延迟的问题，24-EKF 所估计的姿态四元数 q_{ekf} 、速度向量 V_{NED} 和位置向量 P_{NED} 并不直接对外输出，所以 ECL EKF 估计系统采用一个时间滑窗系统跟踪 24-EKF 估计的状态向量，然后动态估计增益系数，再用互补滤波对输出数据进行校正。

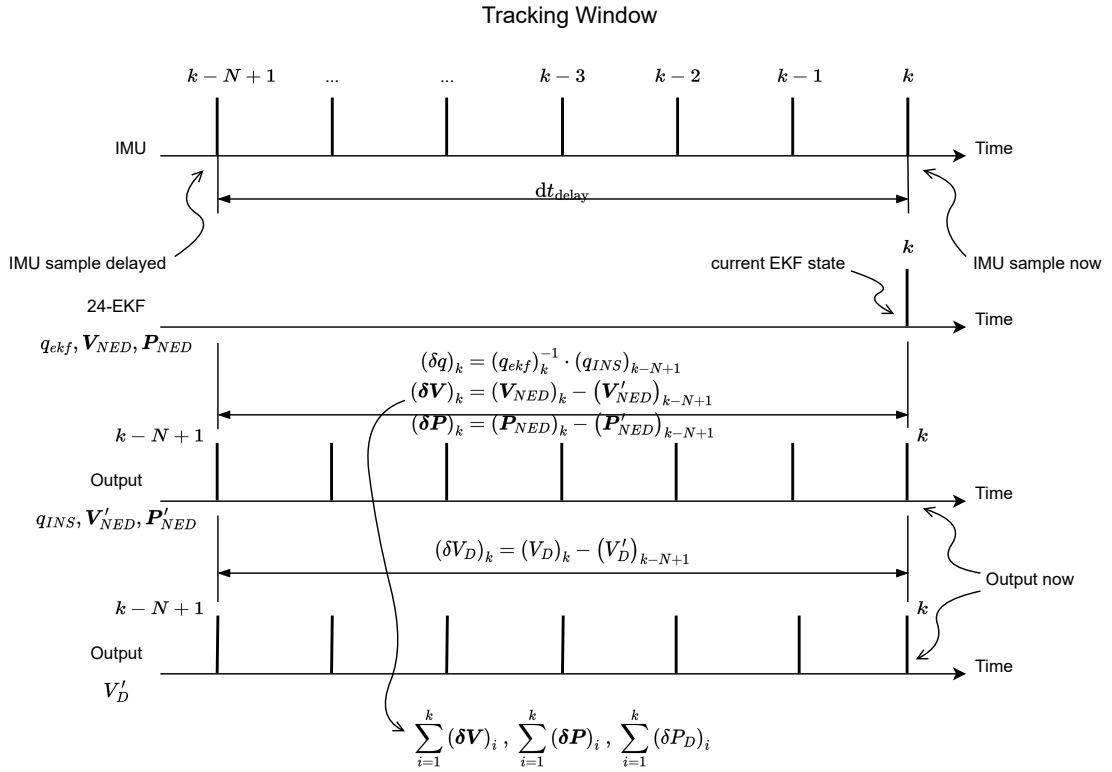
其算法是，程序维护 3 个有 N 个元素的时间滑窗跟踪 24-EKF 估计的状态向量：

1. IMU 采样输入环形缓冲区
2. 三维姿态、速度和位置输出环形缓冲区，输出姿态四元数 q_{INS} 、速度向量 V'_{NED} 和位置向量 P'_{NED} 。
3. z 轴速度输出环形缓冲区，输出 z 轴速度 V'_D 。

其中之所以单独估计 z 轴速度，是因为在 z 轴方向上的速度误差过大，所以需要更精细的估计。该算法基本原理是，假设互补滤波输出的是无延迟的理想值，则延迟估计的 24-EKF 的状态与之就会有误差，该时间滑窗系统就是为了捕捉当前 k 时刻 24-EKF 的状态与输出互补滤波的状态之间的误差值：

1. 时间的误差值, dt_{delay} 。
2. 姿态的误差值, 用四元数表示为 $(\delta q)_k$, 用旋转向量表示为 $(\delta \theta)_k$ 。
3. 三维速度向量和位置向量的误差值, $(\delta \mathbf{V})_k$ 和 $(\delta \mathbf{P})_k$, 及其误差的积分值, $\sum_{i=1}^k (\delta \mathbf{V})_i$ 和 $\sum_{i=1}^k (\delta \mathbf{P})_i$ 。
4. z 轴速度的误差值, $(\delta V_D)_k$, 以及 z 轴位置误差值的积分值 $\sum_{i=1}^k (\delta P_D)_i$ 。

其中, 使用有 N 个元素的时间滑窗就是放大这些误差值, 并且假设 EKF 估计器会收敛到稳态, 因此所观测的误差值也会收敛到稳态。于是在近一段时间内, 这些误差值会趋向某些稳定的局部特征, 并且从长时间看, 这些误差值的分布符合高斯分布, 因此这些误差值的积分趋向于靠近零点的某些特征值。滑窗与输入输出之间的关系如图:



因此我们用这些误差值或特征值计算以下增量增益系数:

1. 增量姿态增益系数, G_θ 。
2. 增量速度增益系数, G_V 。
3. 增量位置增益系数, G_P 。

并且所有的增量值、误差值及其积分在初始化时都设置为零。

10.3 计算过程

10.3.1 计算姿态补偿

使用频率差异计算增量角度的偏差缩放系数

$$S_{dt} = dt_{imu}/dt_{ekf}$$

为在 EKF 融合时间范围内跟踪的四元数状态所需的增量角进行校正

$$\Delta\theta_{truth} = \Delta\theta_{meas} - \Delta\theta_b \cdot S_{dt} + \Delta\theta_{correct}$$

其中, $\Delta\theta_{correct}$ 为上一次迭代时留下的增量角度校正向量, 本次迭代中将在后面使用增量姿态增益系数 G_θ 进行计算, 参见第 10.3.3 节, 并且 $\Delta\theta_{truth}$ 为在机体坐标系中校正后的增量角度。

计算绕地球坐标系垂直方向的偏航角变化

$$\begin{bmatrix} \Delta\phi_I \\ \Delta\theta_I \\ \Delta\psi_I \end{bmatrix} = \Delta\theta_{truth}^T \cdot \mathbf{R}((q_{INS})_{k-1})$$

其中, $\Delta\psi_I$ 为最近绕地球坐标系 z 轴测得的偏航角变化 (rad), \mathbf{R} 为从 $k-1$ 时刻的输出姿态四元数 $(q_{INS})_{k-1}$ 变换来的从机体坐标系到地球坐标系的旋转矩阵。

使用低通滤波计算偏航变化率 (偏航角导数)

$$\dot{\psi} = (1 - \alpha) \dot{\psi} + \alpha \cdot \Delta\psi_I / dt_{imu}$$

其中, $\alpha = 0.05$, 假设偏航变化率短时间内没有大的变化。注意, 固定系数用于保存计算。确切的时间常数并不重要。

用增量四元数旋 δq 转先前的惯性导航系统 (INS) 的姿态四元数 q_{INS} , 并用此更新计算旋转矩阵 \mathbf{R} ,

$$(q_{INS})_k = (q_{INS})_{k-1} \cdot \delta q(\Delta\theta_{truth})$$

修改后的四元数必须进行归一化

$$(q_{INS})_k = (q_{INS})_k / \|(q_{INS})_k\|$$

10.3.2 计算当前加速度、速度和位置

用速度偏差校正机体坐标系的增量速度

$$\Delta\mathbf{V}_{truth}^B = \Delta\mathbf{V}_{meas} - \Delta\mathbf{V}_b \cdot S_{dt}$$

将校正后的增量速度旋转到地球坐标系

$$\Delta\mathbf{V}_{truth}^I = \begin{bmatrix} \Delta V_N \\ \Delta V_E \\ \Delta V_D \end{bmatrix} = \mathbf{R}((q_{INS})_k) \Delta\mathbf{V}_{truth}^B$$

叠加重力引起的测量加速度

$$\Delta\mathbf{V}_{truth}^I = \Delta\mathbf{V}_{truth}^I + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \cdot dt_{imu}$$

计算机体在地球坐标系中的速度导数 (加速度)

$$\mathbf{a}^I = \begin{bmatrix} a_N \\ a_E \\ a_D \end{bmatrix} = \Delta \mathbf{V}_{truth}^I / dt_{imu}$$

通过测量加上校正, 增加惯性导航系统的速度状态

$$(\mathbf{V}_{NED}')_k = (\mathbf{V}_{NED}')_{k-1} + \Delta \mathbf{V}_{truth}^I$$

对于替代校正算法所使用的垂直速度状态也同样如此

$$(V_D')_k = (V_D')_{k-1} + \Delta V_D$$

使用先前时刻和当前时刻的速度, 使用梯形积分计算惯性导航系统的位置状态

$$\Delta \mathbf{P} = \begin{bmatrix} \Delta P_N \\ \Delta P_E \\ \Delta P_D \end{bmatrix} = \frac{1}{2} \left((\mathbf{V}_{NED}')_{k-1} + (\mathbf{V}_{NED}')_k \right) \cdot dt_{imu}$$

$$(\mathbf{P}_{NED}')_k = (\mathbf{P}_{NED}')_{k-1} + \Delta \mathbf{P}$$

对于替代校正算法所使用的垂直位置状态 (垂直速度积分) 也同样如此

$$\sum_{i=1}^k (\delta P_D)_i = \sum_{i=1}^{k-1} (\delta P_D)_i + \Delta P_D$$

用 IMU 偏移校正原点速度。计算上一次 IMU 更新的平均角速度变化率 (角加速度)

$$\boldsymbol{\alpha} = \Delta \boldsymbol{\theta}_{meas} / dt_{imu}$$

我们有 IMU 位于机体坐标系的位置 \mathbf{P}_{imu}^B , 用来计算 IMU 点在机体坐标系中相对于机体原点的速度

$$\mathbf{V}_{imu}^B = \boldsymbol{\alpha} \% \mathbf{P}_{imu}^B$$

将相对速度旋转到地球坐标系中, 就得到 NED 地球坐标系中 IMU 点相对于机体原点的速度

$$(\mathbf{V}_{imu}^I)_k = \begin{bmatrix} V_{imu_N} \\ V_{imu_E} \\ V_{imu_D} \end{bmatrix} = \mathbf{R}((q_{INS})_k) \mathbf{V}_{imu}^B$$

10.3.3 计算增量增益系数

INS 状态存储环形缓冲区与 IMU 数据缓冲区具有相同长度 N 和时间坐标。从 INS 环形缓冲区获取最旧的 INS 状态数据, 为 $k - N + 1$ 时刻的数据, 这些数据将与 24-EKF 当前 k 时刻的融合时间范围状态相比较, 其中两者的时间误差为 dt_{delay} 。该算法主要是为了放大误差值。

在 EKF 融合时间范围内计算 INS 四元数 $(q_{INS})_{k-N+1}$ 和 EKF 四元数 $(q_{ekf})_k$ 之间的四元数增量

$$(\delta q)_k = \begin{bmatrix} \delta q_0 \\ \delta q_1 \\ \delta q_2 \\ \delta q_3 \end{bmatrix} = (q_{ekf})_k^{-1} \cdot (q_{INS})_{k-N+1}$$

将四元数增量转换为增量角度 (小角度近似)

$$\delta\theta = 2 \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \end{bmatrix}$$

计算一个姿态增益, 提供对 24-EKF 估计器姿态状态的严密跟踪, 并调整时间延迟的变化, 以保持约 0.7 的一致阻尼比

$$G_\theta = \frac{1}{2} dt_{imu} / dt_{delay}$$

用其计算增量角度的校正向量, 这将导致 INS 跟踪 24-EKF 姿态四元数, 并且该增量角度校正向量将在下一代被使用到, 参见第 10.3.1 节

$$\Delta\theta_{correct} = G_\theta \cdot \delta\theta$$

计算速度和位置的互补滤波器增益。用户设置的增益系数, $dt_{ekf} < \tau_V, \tau_P < 10$, 由此计算速度和位置的增益系数

$$G_V = dt_{ekf} / \tau_V$$

$$G_P = dt_{ekf} / \tau_P$$

后面将遍历输出滤波器状态历史, $i \in 1 \sim N$, 并将校正应用于速度和位置状态。在这里使用与姿态校正不一样的方法。因为上述用于校正姿态状态的方法需要四元数运算, 对于速度和位置来说太昂贵了。这里的方法由于它允许使用更高的跟踪增益, 因此消除了“校正回路”中的时间延迟, 并减少了相对于 EKF 状态的跟踪误差。

10.3.4 计算垂直速度补偿

计算 z 轴的速度和位置跟踪增量

$$(\delta V_D)_k = (V_D)_k - (V'_D)_{k-N+1}$$

$$(\delta P_D)_k = (P_D)_k - (P'_D)_{k-N+1}$$

计算将应用于输出状态历史的速度校正, 使用一个 PD 反馈调参到 5% 超调

$$\Delta V_{correct} = G_P \cdot \delta P_D + 1.1 \cdot G_V \cdot \delta V_D$$

计算要应用于 V'_D 的校正, 该校正导致垂直位置增量 $\sum_{i=1}^k (\delta P_D)_i$ 使用用于速度和位置状态跟踪的替代算法, 以在融合时间范围内跟踪 EKF 的 z 轴位置状态。该算法对 V'_D 状态历史应用校正, 并使用当前校正后的 V'_D 向前传播 $\sum_{i=1}^k (\delta P_D)_i$ 。这提供了一种替代的垂直速度输出, 该输出更接近于位置的一阶导数, 但相对于 24-EKF 状态确实降低了跟踪成本。

从最早的垂直输出滤波器状态历史开始循环, $i \in 1 \sim N$, 并将校正应用于 V'_D 状态, 并使用校正后的 V'_D 向前传播 $\sum_{i=1}^k (\delta P_D)_i$

- 校正当前的垂直速度

$$(V'_D)_i = (V_D)_i + \Delta V_{correct}$$

- 使用校正后的速度和梯形积分器向前传播位置积分

$$\sum_{i=1}^k (\delta P_D)_i = \sum_{i=1}^{k-1} (\delta P_D)_i + \frac{1}{2} ((V'_D)_k + (V'_D)_{k-1}) \cdot dt_{imu}$$

最后将当前垂直速度的输出状态更新为 INS 输出缓冲区中最新的校正值。

10.3.5 计算三维速度和位置补偿

计算水平速度和位置跟踪误差

$$\begin{aligned} (\delta \mathbf{V})_k &= (\mathbf{V}_{NED})_k - (\mathbf{V}'_{NED})_{k-N+1} \\ (\delta \mathbf{P})_k &= (\mathbf{P}_{NED})_k - (\mathbf{P}'_{NED})_{k-N+1} \end{aligned}$$

计算将应用于输出状态历史的速度校正

$$\Delta \mathbf{V}_{correct} = G_V \cdot (\delta \mathbf{V})_k + 0.1 \cdot G_V^2 \cdot \sum_{i=1}^k (\delta \mathbf{V})_i$$

计算将应用于输出状态历史的位置校正

$$\Delta \mathbf{P}_{correct} = G_P \cdot (\delta \mathbf{P})_k + 0.1 \cdot G_P^2 \cdot \sum_{i=1}^k (\delta \mathbf{P})_i$$

计算要应用于速度和位置输出状态历史的校正。速度和位置状态历史被分别校正，以便它们在融合时间范围内跟踪 EKF 状态。此选项提供最准确的 EKF 状态跟踪。

遍历输出滤波器状态历史， $i \in 1 \sim N$ ，并将校正应用于速度和位置状态

- 应用常量速度校正

$$(\mathbf{V}'_{NED})_i = (\mathbf{V}_{NED})_i + \Delta \mathbf{V}_{correct}$$

- 应用常量位置校正

$$(\mathbf{P}'_{NED})_i = (\mathbf{P}_{NED})_i + \Delta \mathbf{P}_{correct}$$

最后将当前三维速度和位置的输出状态更新为 INS 输出缓冲区中最新的校正值。

10.4 输出状态

在经过互补滤波校正后，ECL EKF 输出以下常用数据：

- 输出姿态四元数 $(q_{INS})_k$ ，用户可用来转换成易读的欧拉角。
- 输出速度，即在局部 NED 地球坐标系中机体坐标系原点的速度

$$\mathbf{V}_{NED}^O = (\mathbf{V}'_{NED})_k - (\mathbf{V}_{imu}^I)_k$$

- 输出加速度，即机体在地球坐标系中的速度导数 \mathbf{a}^I 。
- 输出垂直方向速度，即局部 NED 地球坐标系中机体坐标系原点垂直位置的导数

$$V_D^O = (V'_D)_k - V_{imu_D}$$

- 输出位置，即在局部地球坐标系中机体坐标系原点的位置。将 IMU 相对于机体原点的位置 \mathbf{P}_{imu}^B 旋转到地球坐标系中

$$\mathbf{P}_{imu}^I = \mathbf{R}((q_{INS})_k) \mathbf{P}_{imu}^B$$

将其从 EKF 位置 (即 IMU 位置) 减去，即得到机体原点的位置

$$\mathbf{P}_{NED}^O = (\mathbf{P}_{NED}')_k - \mathbf{P}_{imu}^I$$

- 输出预测姿态四元数。即使用最新的 IMU 增量角数据向前预测先前的四元数输出状态。使用 EKF 的偏差状态估计值校正偏差误差的增量角数据，并应用跟踪 EKF 四元数状态所需的校正

$$\delta\theta = \Delta\theta_{meas} - \Delta\theta_b \cdot S_{dt} + \Delta\theta_{correct}$$

使用校正后的增量角向量校正姿态四元数

$$q_{pred} = q_{INS} \cdot \delta q(\delta\theta)$$

修改后的姿态四元数必须进行归一化

$$q_{pred} = q_{pred} / \|q_{pred}\|$$

在 PX4 项目中组合这些输出数据，对外发布如下信息：

- 发布里程计信息
 - 位置 \mathbf{P}_{NED}^O
 - 姿态四元数 q_{INS}
 - 速度 \mathbf{V}_{NED}^O
 - 角速度 $\omega = (\Delta\theta_{meas} - \Delta\theta_b) / dt_{imu}$
 - 速度方差，24-EKF 的 \mathbf{P} 矩阵速度方阵的对角线向量。
 - 位置方差，24-EKF 的 \mathbf{P} 矩阵位置方阵的对角线向量。
 - 姿态方差，24-EKF 的 \mathbf{P} 矩阵姿态方阵的对角线向量。
- 发布局部位置信息
 - 位置 \mathbf{P}_{NED}^O
 - 速度 \mathbf{V}_{NED}^O
 - 垂直方向速度 V_D^O
 - 加速度 \mathbf{a}^I
 - 经纬度与海拔高度
- 发布 (预测) 姿态 q_{pred} 信息

11 风速估计

因为风速对评估飞行环境十分重要，并且为了容错，所以除了在主要的 24-EKF 估计器中估计风速之外，ECL 还采用另外不同模型的 3-EKF 算法来估计风速。ECL 估计系统对外发布的风速数据由该估计器发布，24-EKF 估计器中估计的风速只在其内部使用。

11.1 假设

假设飞行器为近水平面姿态。飞行器有 z 轴方向上的运动，但是飞行器以近水平面姿态运动。对于旋翼飞机这意味着零侧滑 (Zero sideslip) 假设。

假设风速为近水平面风场，因此 z 轴方向风速为 0。用 V_{wind_N}/V_{wind_E} 表示在 NED 坐标系中在北/东 (N/E) 方向上的风速，则风速向量 \mathbf{V}_{wind} 表示为

$$\mathbf{V}_{wind} = \begin{bmatrix} V_{wind_N} \\ V_{wind_E} \\ 0 \end{bmatrix}$$

真实空速 (True Air Speed, TAS), 又称真空速, 是表示飞行器飞行时相对于周围空气运动的速度。真实空速的方程与温度和空气密度相关。在这里做一个简化, 假设其为飞行器相对于地球固定坐标系的速度向量 \mathbf{V}_I 和风速向量 \mathbf{V}_{wind} 的差值, 并乘上真实空速缩放因子 k_{TAS} 。

所以, 最终的要估计的状态向量 \mathbf{x} 为

$$\mathbf{x} = \begin{bmatrix} V_{wind_N} \\ V_{wind_E} \\ k_{TAS} \end{bmatrix}$$

并且假设风场在短时间内不会有大的变化, 所以采用固定过程模型。

外部系统输入航向角 ψ , 以及飞行器相对于地表的速度向量 $\mathbf{V}_I = \begin{bmatrix} V_N & V_E & V_D \end{bmatrix}^T$, 我们关注 N/E 方向的地面速度方差 $\sigma^2 = \begin{bmatrix} \sigma_{V_N}^2 & \sigma_{V_E}^2 \end{bmatrix}^T$ 。对于风速的过程噪声分量为 Q_w , 对于真实空速过程噪声分量为 $Q_{k_{TAS}}$ 。对于固定翼飞机, 由空速传感器提供真实空速测量值 V_{meas}^{TAS} , 其测量噪声为 R_{TAS} ; 对于旋翼飞机, 由合成侧滑 beta 计算真实空速测量值, 其测量噪声为 R_{beta} 。

11.2 初始化

初始化阶段的主要目标是通过地面速度初始化风速协方差。这是一种对准算法, 与时间更新阶段的方程不一样。

由外部输入航向角 ψ 。根据近水平面的假设, 根据飞行器地面速度、航向和空速测量值计算风速估计值 (状态向量)

$$\text{wind}_{est} = \begin{bmatrix} V_{wind_N} \\ V_{wind_E} \end{bmatrix} = \begin{bmatrix} V_N \\ V_E \end{bmatrix} - \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix} \cdot V_{meas}^{TAS}$$

风速向量的估计与水平速度 (V_N/V_E), 航向角 ψ , 合成侧滑 beta, 空速测量值 V_{meas}^{TAS} 这 5 个量相关。因为是零侧滑, 所以使用航向角 ψ 的偏导数传播合成侧滑方差。因此, 风速的状态转移矩

阵 \mathbf{F} ，其为相对于这 5 个量的 Jacobian 矩阵，

$$\mathbf{F} = \frac{\partial (\text{wind}_{est})}{\partial \begin{bmatrix} V_N \\ V_E \\ \psi \\ \psi \\ V_{meas}^{TAS} \end{bmatrix}}$$

设置初始状态协方差矩阵的对角线为前面 5 个量的方差

$$\mathbf{P}_0 = \text{diag} \begin{bmatrix} \sigma_{V_N}^2 \\ \sigma_{V_E}^2 \\ \sigma_{\psi}^2 \\ \sigma_{beta}^2 \\ \sigma_{TAS}^2 \end{bmatrix}$$

根据状态转移矩阵 \mathbf{F} ，获得初始化时的状态协方差矩阵 \mathbf{P} 的估计值

$$\mathbf{P} = \mathbf{F} \mathbf{P}_0 \mathbf{F}^T$$

设置状态向量 \mathbf{x} 初始值为

$$\mathbf{x} = \begin{bmatrix} V_{wind_N} \\ V_{wind_E} \\ 1 \end{bmatrix}$$

11.3 时间更新

因为算法采用的是固定过程模型， $\mathbf{x}_{k+1} = \mathbf{x}_k$ ，所以我们只需要提供协方差矩阵 \mathbf{P} 的预测方程

$$\mathbf{P}_{k|k-1} = \mathbf{P}_{k-1|k-1} + \begin{bmatrix} Q_w \\ Q_w \\ Q_{kTAS} \end{bmatrix} \cdot dt$$

11.4 测量更新

11.4.1 真实空速融合

对于带有空速传感器的非旋翼无人机，采用这个分支。本节的方程可对比第 6.4 节“空速融合方程”的内容。

由外界输入真实空速测量值 V_{meas}^{TAS} 和飞行器相对于地表的速度 $\mathbf{V}^N = \begin{bmatrix} V_N & V_E & V_D \end{bmatrix}^T$ 。

计算真实空速预测值 V_{pred}^{TAS}

$$\begin{aligned} V_{pred}^{TAS} &= \|\mathbf{V}^N - \mathbf{V}_{wind}\|_2 \cdot k_{TAS} \\ &= \sqrt{(V_N - V_{wind_N})^2 + (V_E - V_{wind_E})^2 + V_D^2} \cdot k_{TAS} \end{aligned}$$

真实空速的观测矩阵 \mathbf{H}_{TAS}

$$\begin{aligned}\mathbf{H}_{TAS} &= \frac{\partial (V_{pred}^{TAS})}{\partial \mathbf{x}} \\ &= \frac{\partial \left(\sqrt{(V_N - V_{wind_N})^2 + (V_E - V_{wind_E})^2 + V_D^2 \cdot k_{TAS}} \right)}{\partial \left(\begin{bmatrix} V_{wind_N} \\ V_{wind_E} \\ k_{TAS} \end{bmatrix} \right)}\end{aligned}$$

计算新息 (或残差) 协方差 S

$$S = \mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + R_{TAS}$$

最佳卡尔曼增益 $\mathbf{K}_{3 \times 1}$

$$\mathbf{K} = \mathbf{P}_{k|k-1} \mathbf{H}^T S^{-1}$$

更新 (后验) 的协方差 $\mathbf{P}_{3 \times 3}$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K} \mathbf{H} \mathbf{P}_{k|k-1}$$

最后注意, 风速融合频率不超过 10 Hz。

11.4.2 合成侧滑融合

对于提供虚拟合成侧滑数值的旋翼无人机, 采用这个分支。本节的方程可对比第 6.5 节“合成侧滑融合方程”的内容。

由外界输入姿态四元数 q 和飞行器相对于地表的速度 $\mathbf{V}^N = \begin{bmatrix} V_N & V_E & V_D \end{bmatrix}^T$ 。

计算在飞行器机体坐标系中的相对于地球坐标系的风速向量

$$\begin{aligned}\mathbf{V}_{wind}^N &= \mathbf{V}^N - \mathbf{V}_{wind} \\ &= \begin{bmatrix} V_N - V_{wind_N} \\ V_E - V_{wind_E} \\ V_D \end{bmatrix}\end{aligned}$$

根据外部输入的姿态四元数 q 得到从机体坐标系到导航坐标系的旋转矩阵 $[T]_B^N$

$$[T]_B^N = \mathbf{R}(q)$$

计算相对于机体坐标系的风速

$$\begin{aligned}\mathbf{V}_{wind}^B &= \left([T]_B^N \right)^T \cdot \mathbf{V}_{wind}^N \\ &= \begin{bmatrix} V_{wind_x} \\ V_{wind_y} \\ V_{wind_z} \end{bmatrix}\end{aligned}$$

合成侧滑模型的小角度近似方程为

$$\text{beta} = \frac{V_{wind_Y}}{V_{wind_X}}$$

并且根据零侧滑假设, $-\text{beta}$ 即为新息。

计算合成侧滑的观测矩阵 \mathbf{H}_{beta}

$$\begin{aligned} \mathbf{H}_{\text{beta}} &= \frac{\partial(\text{beta})}{\partial \mathbf{x}} \\ &= \frac{\partial \left(\frac{V_{wind_Y}}{V_{wind_X}} \right)}{\partial \left(\begin{bmatrix} V_{wind_N} \\ V_{wind_E} \\ k_{TAS} \end{bmatrix} \right)} \end{aligned}$$

后面的计算如上一小节一样。

12 ECL EKF 的架构设计

PX4 项目主要使用 C++ 实现。但是其中的代码并没有使用 C++ 的高级功能, 称之为带类的 C 也不为过。在总体架构上, PX4 项目采用微内核的设计思想, 即各个功能模块都运行在各自的独立空间, 由 RTOS 提供消息机制, 各个模块相互之间不能直接访问, 而只能通过消息交换数据。这种方法有效地避免了因共享变量的并发访问而带来的复杂的同步与互斥机制, 从而使得软件模型变得简单。但是这也会带来其它一些问题, 例如, 消息海洋的问题, 有时会让程序难以定位问题并且难以调试, 同时整个系统的效率严重依赖 OS 提供的消息机制的效率。

ECL EKF 估计系统主要有三类主动对象 (Active Object)

1. 传感器驱动。一种传感器封装成一个主动对象。
2. 估计器。有 24-EKF 位姿估计器和 3-EKF 风速估计器两种, 其中前者可能有多个实例同时运行, 参见第 17 章, 后者参见第 11 章。
3. 使用估计数据的模块。有多个, 靠订阅消息获取估计信息, 参见第 10.4 节。

每个主动对象都绑定在一个线程 (Thread) 上, 用生产-消费模式异步运行, 这也造成了多速率多延迟的问题, 参见第 10 章的描述。

ECL EKF 的核心算法部分为了可移植性, 没有采用 C++ 语言的高级功能, 基本上是 C 语言的过程模型, 并且它们也没有调用 OS 或外部模块的特殊函数 (注意, 这个原则已经被打破, 最新的版本已经依赖 uORB 消息接口), 基本上都是数学公式的代码, 所以可以比较方便地移植到其它项目中。后面简单讨论核心算法的流程以及传感器数据的流向。

12.1 算法流程

ECL EKF 的核心算法流程是一个事件处理的大循环, 其线程被唤醒后就进行事件处理, 其方法名为 `bool Ekf::update()`。唤醒事件有各种传感器数据输入, 还有一个 10 Hz 的定时事件。当一段时间内没有任何传感器数据输入时, 系统以 10 Hz 频率进行惯性航位推算, 并对外报告异常。算法的处理流程如下 (在不同的版本之间会有一些差异):

1. 运行主要的 24-EKF 估计器 (100 Hz)

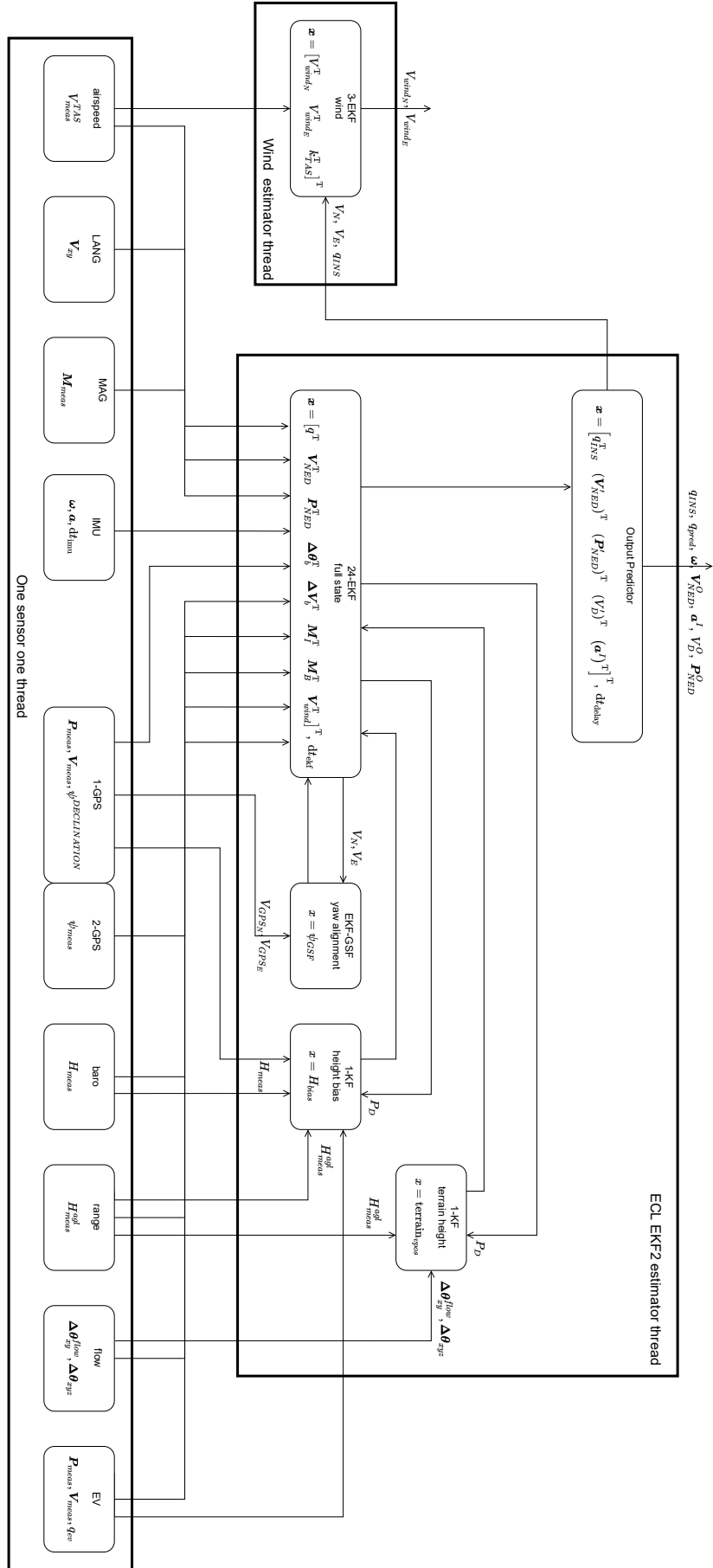
- (a) 预测新的状态 (24-EKF 方程 1), 参见第 5 章。
- (b) 预测新的协方差矩阵 (24-EKF 方程 2), 参见第 5 章。
- (c) 受控制的观测数据的融合 (24-EKF 方程 3, 4, 5), 参见第 6 章。
 - i. 检查倾斜对准并在对准后选择高度数据源, 参见第 9 章。
 - ii. 检查 baro/GPS/MAG/range/flow/EV/airspeed 等传感器的可用情况, 参见第 13 章。
 - iii. 检查并在必要时切换高度数据源 (baro/EV/GPS/range), 参见第 16 章。
 - iv. 运行验证偏航角 ψ 估计器 (EKF-GSF), 参见第 8 章。
 - v. 磁力融合, 参见第 6.1 节。
 - vi. 光流融合, 参见第 6.2 节。
 - vii. GPS 融合, 参见第 6.3 节。
 - viii. 空速融合, 参见第 6.4 节。
 - ix. beta 融合, 参见第 6.5 节。
 - x. drag 融合, 参见第 6.6 节。
 - xi. 高度融合, 参见第 6.7 节。
 - xii. 外部视觉融合, 参见第 6.8 节。
 - xiii. 辅助速度融合, 参见第 6.9 节。
 - xiv. 零新息航向角更新。当飞行器处于静止状态时, 航向角的新息很小, 需要设置特别参数, 以便估计器能正常运行。
 - xv. 零速度更新。当飞行器处于静止状态时, 速度的新息很小, 需要设置特别参数, 以便估计器能正常运行。
 - xvi. 虚拟位置融合, 参见第 6.10 节。
 - xvii. 虚拟高度融合, 参见第 6.11 节。
 - xviii. 更新航位推算状态, 检查我们是否不再融合直接约束速度漂移的测量。参见第 19 章。

2. 运行地形高差估计器 (1-KF), 参见第 7 章。

3. 运行输出观测器, 计算当前时间范围的输出补偿 (100 Hz), 参见第 10 章。

12.2 数据流向

下面简单总结传感器测量值和估计值的输入输出流向, 注意, 图中没有包含噪声与方差的传播路径。



12.3 算法优化

在 PX4 项目中的算法，以前采用 matlab 描述，后来采用 python/SymPy 描述，C++ 代码则是由符号推导系统优化后的展开式。这种 C++ 计算式很难看，基本无法理解。如今情况有些改观，PX4 项目中的算法开始转向用 python/SymForce 描述。SymForce 可译为“符号动力”。虽然 SymForce 也是基于 SymPy 构建，但是其输出更为易读一些，因为 SymForce 号称其代码生成器针对任何语言都使用一个模板系统将符号表达式转换为快速的、无分支的代码，并且有着干净的 API 和最小的依赖项。

ECL 估计器采用的大多是 EKF 算法，其中的 Jacobian 矩阵十分重要，而且大多很复杂，基本上无法手写推导。采用符号推导系统的优点，一是可以针对复杂的原函数生成 Jacobian 矩阵，设计人员只需要考虑信息的关联关系而不需要关心如何生成复杂的偏导方程，这样就释放了设计人员的想象力，二是开发人员只需要维护原函数，如何快速计算 Jacobian 矩阵，由已经长久验证的成熟模板生成优化代码，比手写的快且不容易出错。但是缺点也有，就是生成的代码丧失了可读性，使得函数代码又长又难看。

现在更换成 SymForce 以后，对比以前老版本的 SymPy 的输出代码，SymForce 的输出代码的格式要漂亮一些，至少能读个半懂，从输出的注释中能了解一些其中的步骤和意图。并且有了 SymForce 提供的封装函数，应用程序总体上也会简洁易读一些，不会被原先那些奇奇怪怪代码打断。至于 SymForce 的输出代码能否运行得更快一些，因为 ECL 估计算法的规模太小，所以无法判断。

12.4 面向问题方面的分析

以上章节的方程基于面向流程的分析方式进行组织。但是在实际工程中，问题十分复杂，所以 ECL/EKF2 的代码流程也十分复杂，我们还需要基于另一个维度的分析，即面向问题方面的分析。

面向问题方面有如下话题：

- 异常判断算法，参见第 13 章。
- 偏航角数据的管理，参见第 14 章。
- 水平面数据的管理，参见第 15 章。
- 高度数据的管理，参见第 16 章。
- 估计器多实例的管理，参见第 17 章。
- 惯性航位推算，参见第 19 章。

13 异常判断算法

对于卡尔曼滤波系统的运行情况，通常采用马氏距离进行判断。马氏距离 (Mahalanobis distance) 是一种衡量点 P 和概率分布 D 之间距离的方法，它是对衡量 P 距离 D 的均值有多少标准差的概念的多维概括。 P 在 D 的均值处的距离为零，并随着 P 沿每个主分量轴远离均值而增长。如果这些轴中的每一个都被重新标定为具有单位方差，那么马氏距离就对应于变换空间中的标准欧几里德距离。因此，马氏距离是无单位的、尺度不变的，并考虑到了数据集的相关性。

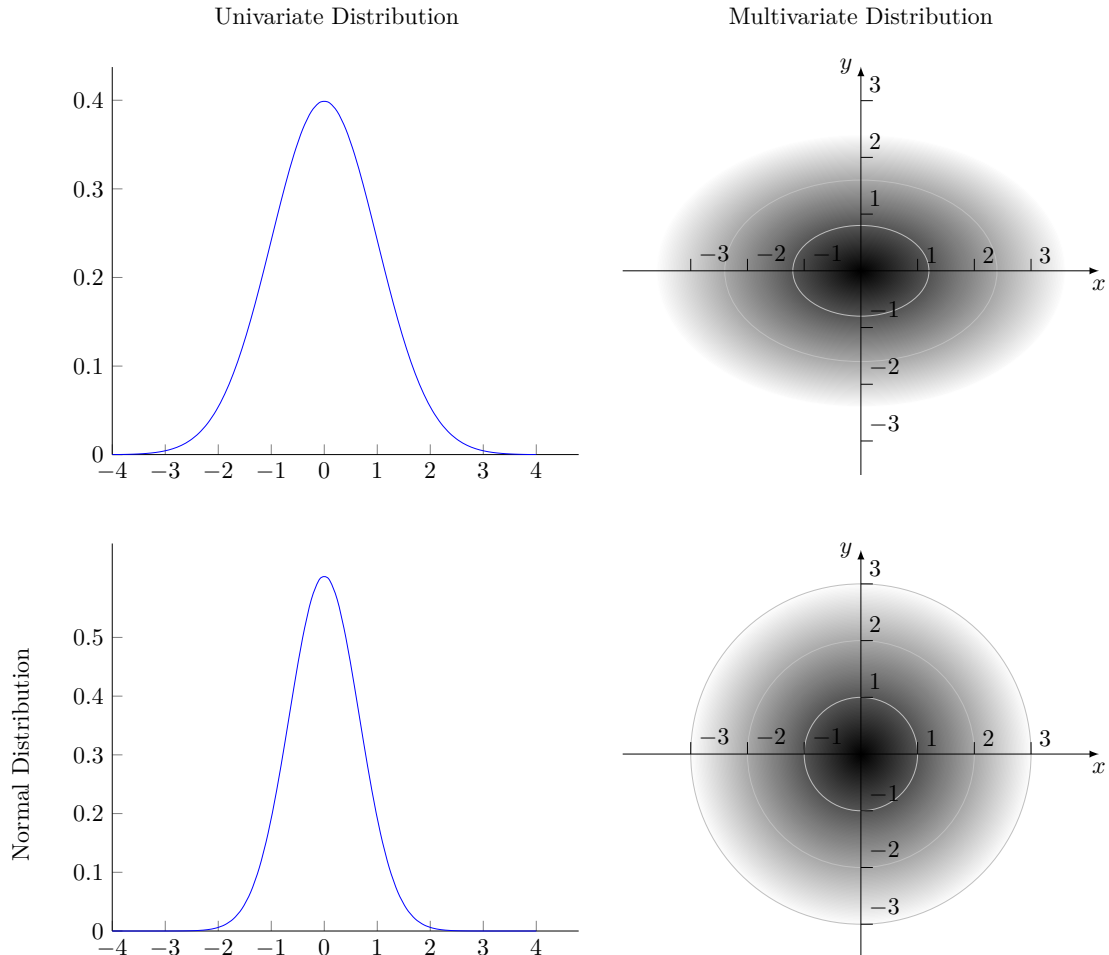
设一组观测值为 $\mathbf{x} = (x_1, x_2, x_3, \dots, x_N)^T$ 与一组具有均值 $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \dots, \mu_N)^T$ 及其协方差矩阵 \mathbf{S} 的观测值的马氏距离定义为

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

在卡尔曼滤波的应用中，我们以状态的当前 k 时刻的预测值 $\hat{\mathbf{x}}_{k|k-1}$ 为均值，其位于新息协方差矩阵 \mathbf{S} 表示的概率中心，则马氏距离 (或其平方值即为“广义的点间距离的平方”) 也可以定义为在用新息协方差矩阵 \mathbf{S} 表示的相同概率分布下，测量值 z_k 与均值 $h(\hat{\mathbf{x}}_{k|k-1})$ 之间的距离，即测量值到达概率中心的距离：

$$D_M^2(\mathbf{z}, \mathbf{x}) = (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}))^T \mathbf{S}^{-1} (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}))$$

该方法的直觉是，对于单维状态，输入数据的方差将标准正态分布扩散为非标正态分布，对于多维状态，各维数据的方差将圆球状的白化数据扩散为椭球状，而马氏距离则将这些被方差驱动扩散的数据重新变换为标准正态分布或圆球状的白化数据。因此这种方法又称为标准化新息平方 (Normalized Innovation Squared, NIS)。NIS 是一个标量数据，它的分布符合卡方分布。



卡尔曼滤波属于动态递归的最优化算法，协方差矩阵在算法里面起着核心作用。所以对于卡尔曼滤波器当前的质量状态，可以通过对当前的协方差矩阵状态进行判断。具体就是使用新息向量和新息协方差计算马氏距离进行判断。我们可以设置一个门限 (gate)，比如 5，代表着 ± 5 倍的标准

差, 根据常识, ± 3 倍的标准差已经包含 99.7% 的数据, 因此如果新息的马氏距离超过门限则说明该测量值已不可信, 我们就称其为离群值 (Outlier)。

因为在卡尔曼滤波器中, 协方差矩阵是由新息驱动而动态变化的。因为统计学上杠杆效应问题, 距离概率中心越远的离群值, 对协方差矩阵的影响越大。如果频繁发生马氏距离超过门限的事件, 则说明或者是测量值或者是协方差矩阵质量出现了问题。

PX4/ECL 项目采用一种更直观的度量, 即测试率 (test ratio), 以表示各个输入测量值的质量状况。测试率将测量向量的各维数据拆开单独计算 NIS, 用比率直观判断该维数据的质量, 其中新息协方差矩阵 \mathbf{S} 中的相应对角线元素被称为该维数据的新息方差:

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \\ \sigma_i^2 &= \mathbf{S}(i, i) \\ \text{test_ratio}_i &= \tilde{y}_i^2 / (\text{gate} \cdot \sigma_i)^2\end{aligned}$$

当 $\text{test_ratio}_i > 1$, 则我们判断该维的数据已经超过标准差的门限倍数, 比如我们设置 $\text{gate} = 3$, 则我们认为该维数据已经超出 ± 3 倍的标准差的范围。

传感器测量值新息的门限大小列表如下:

- `baro_innov_gate = 5.0`: 气压和 GPS 高度新息一致性门限大小 (STD)
- `gps_pos_innov_gate = 5.0`: GPS 水平位置新息一致性门限大小 (STD)
- `gps_vel_innov_gate = 5.0`: GPS 速度新息一致性门限大小 (STD)
- `heading_innov_gate = 2.6`: 航向融合新息一致性门限大小 (STD)
- `mag_innov_gate = 3.0`: 磁力计融合新息一致性门限大小 (STD)
- `mag_acc_gate = 0.5`: 当处于自动选择模式时, 当机动加速度低于此值 (m/s^2) 时, 将使用航向融合
- `mag_yaw_rate_gate = 0.25`: 模式选择逻辑使用的偏航率阈值 (rad/s)
- `tas_innov_gate = 5.0`: 真实空速新息一致性门限大小 (STD)
- `beta_innov_gate = 5.0`: 标准偏差下的合成侧滑新息一致性门限大小 (STD)
- `range_innov_gate = 5.0`: 测距仪融合新息一致性门限大小 (STD)
- `range_aid_innov_gate = 1.0`: 用于距离辅助融合新息一致性检查的门限大小
- `range_kin_consistency_gate = 1.0`: 测距仪运动一致性检查使用的门限大小
- `ev_vel_innov_gate = 3.0`: 视觉速度融合新息一致性门限大小 (STD)
- `ev_pos_innov_gate = 5.0`: 视觉位置融合新息一致性门限大小 (STD)
- `flow_innov_gate = 3.0`: 光流融合新息一致性门限大小 (STD)
- `auxvel_gate = 5.0`: 辅助速度融合新息一致性门限大小 (STD)

在 ECL 估计系统中的测试率列表如下：

- `yaw_test_ratio`：偏航角新息与新息测试极限的比率。
- `mag_test_ratio`：最大磁力计新息分量与新息测试极限的比率。
- `vel_test_ratio`：最大速度新息分量与新息测试极限的比率。
- `pos_test_ratio`：最大水平位置新息分量与新息测试极限的比率。
- `hgt_test_ratio`：垂直位置新息与新息测试极限的比率。
- `tas_test_ratio`：真实空速新息与新息测试极限的比率。
- `hagl_test_ratio`：距地高度新息与新息测试极限的比率。
- `beta_test_ratio`：合成侧滑新息与新息测试极限的比率。
- `innov_test_ratio`：标准化新息平方 (NIS) 测试率。
- `combined_test_ratio`：组合测试率，是速度和位置平均测试率和高度测试率的最大值，
`max((vel_test_ratio + pos_test_ratio) / 2, hgt_test_ratio)`
- `relative_test_ratio`：相对测试率，在多实例 EKF 中与当前实例的组合测试率有最大差值的测试率
`combined_test_ratio[i] - combined_test_ratio[selected]`

14 偏航角数据的管理

在 ECL 估计系统中，偏航角的数据源有 3 种：

1. 具有双天线的 GPS，算法参见第 6.3.1 节。
2. 外部视觉 (EV)，算法参见第 6.8 节。
3. 磁力计，算法参见第 6.1 节。

在同一时刻只能使用其中一种，这个顺序也是内部选择的优先级。软件根据传感器的数据质量、数据连续性，以及偏航新息测试率 (`yaw_test_ratio`) 进行掉落选择。其中对于 GPS 的数据质量检查得最多，因为其最重要。但是因为成本的原因，在大多数情况下，飞行器只有磁力计这一个易受外界干扰的偏航角数据源。例如，地面磁场异常足以影响磁力计，当飞行器飞离地面一定高度后 (1.5m)，需要重置偏航角，以便从地面磁干扰中恢复。

15 水平面数据的管理

在 ECL 估计系统中，水平面的数据源有 4 种：

1. GPS 提供 \mathbf{P}_{NED} 和 \mathbf{V}_{NED} ，算法参见第 6.3.2 节。

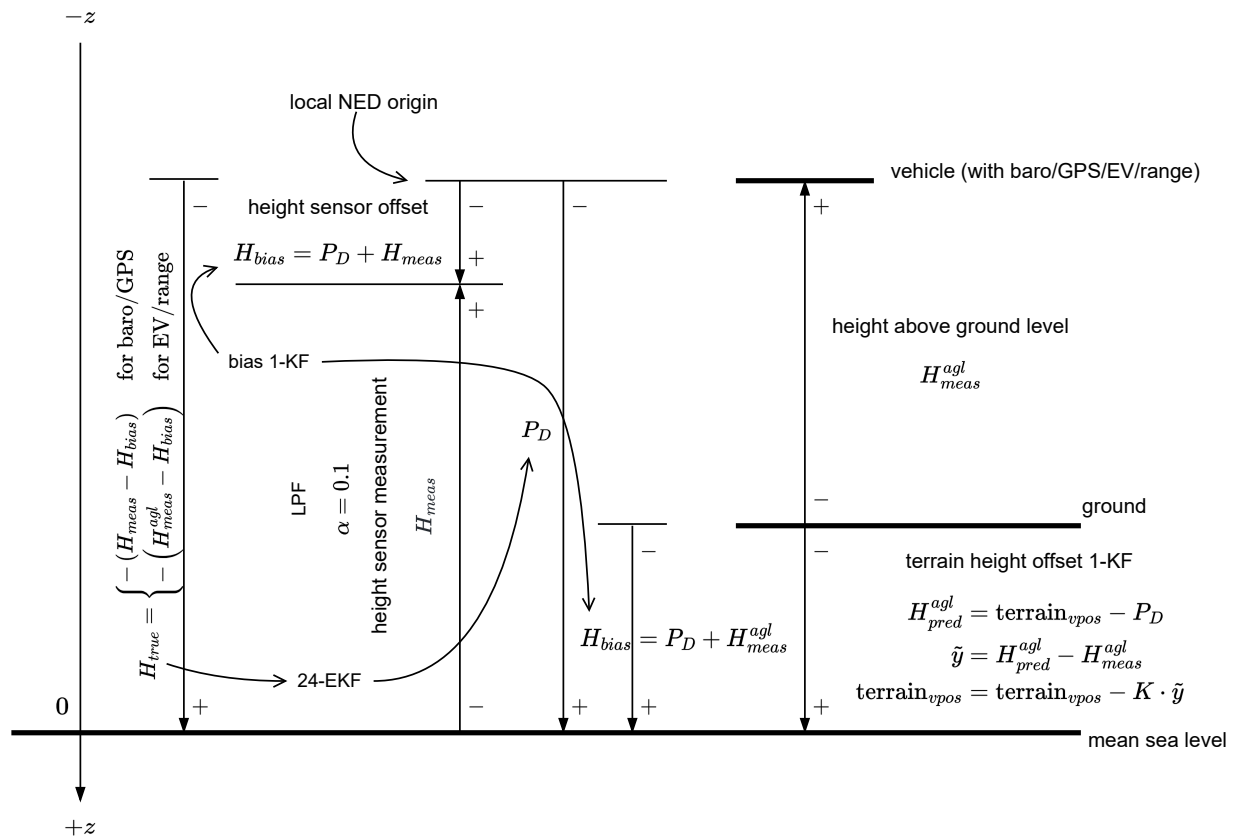
2. 光流提供水平面的速度 V_N 和 V_E ，算法参见第 6.2 节。
3. 外部视觉提供 P_{NED} 和 V_{NED} ，算法参见第 6.8 节。
4. 风场提供 V_{wind_N} 和 V_{wind_E} ，参见第 6.4 节，以及相关的合成侧滑 β ，参见第 6.5 节。

前 3 种数据源可以同时使用以校正水平面的速度与位置。因为估计风速的误差太大，所以只有在前 3 种数据源不可用的情况下，才会使用第 4 种数据源，这时 ECL 估计系统进入风场航位推算模式，如果连风速数据都没有，则会进入水平航位推算模式，参见第 19 章的讨论。软件根据速度新息测试率 (vel_test_ratio) 和位置新息测试率 (pos_test_ratio) 判断水平面数据的质量状况。

16 高度数据的管理

因为高度数据源较多，且基准不一样，偏差不一样，所以需要进行统一管理，以去除测量高度的偏差，并在相同规范下计算出高度新息和新息方差，再交由 24-EKF 估计出最优高度值。

在 24-EKF 的状态向量中， $P_{NED} = [P_N \ P_E \ P_D]^T$ 表示机体在 NED 坐标系中的位置，其中 P_D 为机体相对于海平面的高差，即负的海拔高度。注意，在 NED 坐标系中， z 轴向下为正。而高度传感器的测量值经过处理后，统一得到的是海拔高度，其中，虽然测距仪测量得到的是距地高度，但加上第 7 章维护的地形高度，仍然可以得到海拔高度，测量数据都是 z 轴向上为正。因此，在 ECL 估计系统中有如下的高度数据关系：



高度偏差估计器是一个标准卡尔曼滤波器，高度偏差 $H_{bias} = P_D + H_{meas}$ 用一维标准卡尔曼滤波算法进行最优估计，然后计算去除偏差的测量值 H_{true}

$$H_{true} = \begin{cases} -(H_{meas} - H_{bias}) & \text{for baro/GPS} \\ -(H_{meas}^{agl} - H_{bias}) & \text{for EV/range} \end{cases}$$

将其交由 24-EKF 对 P_D 进行最优估计。另外，因为气压测量的不稳定，其测量值用一个 α 低通过滤器进行平滑处理，其中 $\alpha = 0.1$ ，代表着在最小 5Hz 的采样时间内，高度不会有太大的变化。后面两节内容将讨论高度偏差估计器的设计 and 应用。

16.1 单状态偏差估计器

此处的偏差 (bias)，指的是高度传感器测量值相对于 P_D 的偏移 (offset)。我们假定机体在短时间内，即在最小 5Hz 的采样率下，偏差没有大的变化，因此采用固定模型，并且认为偏差由噪声驱动。

对于单状态偏差 x 的演化方程为：

$$\begin{aligned} x_{k+1} &= Fx_k + v_k \\ y_k &= x_k + w_k \end{aligned}$$

其中，噪声 $v_k \sim \mathcal{N}(0, Q_k)$, $w_k \sim \mathcal{N}(0, R_k)$ 。因为该状态由噪声驱动，则状态转换矩阵 $F = 1$ 。

在时间更新阶段，因为状态为固定模型，因此只预测状态协方差。我们有过程噪声标准差，即过程噪声谱密度 (process noise spectral density, nsd)，其平方为过程噪声方差，即状态过程平方谱密度 (state process power spectral density, psd) Q , $m^2/s^2/\text{Hz}$ ，我们要求当前时刻的状态不确定性方差 (state uncertainty variance) P , m^2 ：

$$\begin{aligned} P_{k|k-1} &= P_{k-1|k-1} + G \cdot Q \cdot dt \\ &= P_{k-1|k-1} + G \cdot \text{nsd}^2 \cdot dt \end{aligned}$$

其中， G 为激励增益，在正常情况下， $G = 1$ ，当新息序列检查时检测到有大的状态偏差，增大过程噪声， $G = 1000$ ，直到消除状态偏差。其判断标准是，如果新息的平均值在统计上过大， $|\text{signed_innov_test_ratio}| > 0.2$ ，或者新息的正负符号在一段时间内 (10s) 始终相同，则估计值存在大的偏差。

在测量更新阶段，有测量值 z ，因为高度是直接观测，所以观测矩阵 $H = 1$ 。首先计算新息方差

$$S = P + R$$

计算新息

$$\tilde{y} = z - x$$

计算卡尔曼增益

$$K = P/S$$

计算标准化新息平方测试率

$$\text{innov_test_ratio} = \tilde{y}^2 / (\text{gate}^2 \cdot S)$$

其中新息门限 $\text{gate} = 3$ 。当新息测试率 $\text{innov_test_ratio} < 1$ 时，继续执行卡尔曼滤波的后面两个步骤，否则将该离群的测量值剔除。更新状态值为

$$x = x + K \cdot \tilde{y}$$

更新状态不确定性方差

$$P = P - K \cdot P$$

最后检查新息序列以在下一个迭代中检测状态偏差情况。其算法是，我们用低通滤波统计带正负符号的新息测试率 $\text{signed_innov_test_ratio} = \text{sign}(\tilde{y}) \cdot \text{innov_test_ratio}$ 的值，正常情况下应该是一段时间内正负符号均匀出现（连续出现同一种符号的时间 $t < 10\text{ s}$ ），均值趋向零。如果不符合条件，则说明状态出现大的偏差，在下次迭代时，加大过程噪声方差。

16.2 高度估计器的应用

高度数据源比较多，有 4 种：

1. 气压计 (baro) 提供海拔高度。
2. GPS 提供海拔高度。
3. 外部视觉 (EV) 提供距地高度。
4. 测距仪 (range) 提供距地高度。

同一时刻只能使用其中一种。在 ECL 系统内部切换数据源的算法有点复杂。

首先定义有以下高度传感器类型：

1. BARO
2. GNSS
3. RANGE
4. EV
5. UNKNOWN

并为前 4 种实际的传感器创立 4 个单状态偏差估计器实例，每个实例保存自己的类型 ID，并维护一个 `bool active` 状态。并且在 ECL 内部定义使用高度传感器类型的两个参考变量：

- `_params.height_sensor_ref`：用户指定的第一优先级高度传感器，也代表不同的高度传感器选择优先级。
- `_height_sensor_ref`：如果该变量数值与上面相同，则代表第一优先级的高度传感器当前已被激活。如果该值为 `UNKNOWN`，则代表第一优先级高度传感器失效，估计器现在正在使用更低优先级的高度传感器。

选择优先级的列表如下：

	BARO 1st	GNSS 1st	RANGE 1st	EV 1st	UNKNOWN 1st
1	BARO	GNSS	RANGE	EV	GNSS
2	GNSS	BARO	EV	RANGE	BARO
3	EV	EV	BARO	BARO	EV
4	RANGE	RANGE	GNSS	GNSS	RANGE

所谓第一优先级高度传感器 (`sensor_id == _params.height_sensor_ref`), 因其测量值已经直接进入 24-EKF 中计算高度新息及其方差, 所以需要在其偏差估计器实例中旁路 1-KF 的计算。而其它实例在被激活后 (`active = true`), 使用标准卡尔曼滤波器对高度偏差进行最优估计。最后, 软件通过高度新息测试率 (`hgt_test_ratio`) 或距地高度新息测试率 (`hagl_test_ratio`) 发现高度数据源的故障, 并根据上述优先级选择下一个高度数据源。并且高度数据的标准化新息平方测试率 (`innov_test_ratio`) 与加速计的削波统计数据配合, 还可以发现加速计在 z 轴上的故障。

值得注意的是, 对于多旋翼无人机, 当其降落并使用气压作为高度数据源时, 因旋翼与地面相互作用引起的正静态压力瞬变, 会引起高度估计出现较大波动, 其表现为产生负的垂直位置新息(地效气压升高, 则海拔虚假升高, 因为 NED 坐标系 z 轴向下为正, 所以出现负的新息)。为补偿这种效应, 需要在降落小于一定高度时, 激活地面效应补偿, 该高度一般设置为 0.5 m , 应用于负的气压新息的地面效应死区为 5 m 。

17 估计器多实例的管理

为了容错, 很多飞行器会有两个 IMU, 甚至会有两个磁力计。在以前的应用里, 一个在工作一个在备份, 这实际上是一种浪费。如今可以根据 IMU 和磁力计的数量以及自动驾驶仪的 CPU 能力, 可以运行多个 24-EKF 实例, 将 IMU 和磁力计的数据交叉输入以计算估计值。通过每个 24-EKF 实例使用不同的传感器组合, 提供了一系列更广泛的传感器错误的保护。通过比较每个 24-EKF 实例的内部一致性, EKF 选择器能够确定具有最佳数据一致性的 24-EKF 和传感器组合。这样可以检测和隔离 IMU 偏差、饱和或数据卡顿等故障。

24-EKF 实例总数是由 `EKF2_MULTI_IMU` 和 `EKF2_MULTI_MAG` 所选择的 IMU 数量和磁力计数量的乘积, 由以下公式给出:

$$N_{\text{instances}} = \text{MAX}(\text{EKF2_MULTI_IMU}, 1) \times \text{MAX}(\text{EKF2_MULTI_MAG}, 1)$$

可处理的 IMU 或磁力计传感器的最大数量为每种传感器有 4 个, 因此理论上最大有 $4 \times 4 = 16$ 个 24-EKF 实例。实际上, 这种做法受到现有计算资源的限制。在开发这一功能的过程中, 使用基于 STM32F7 的硬件的 CPU 进行测试, 结果显示 4 个 24-EKF 实例具有可接受的处理负载和内存利用率裕度。

判断和选择当前工作的 24-EKF 实例的方法是为每一个实例维护两个新息测试率: 组合测试率 (`combined_test_ratio`) 和相对测试率 (`relative_test_ratio`), 其中, 前者是速度和位置平均测试率和高度测试率的最大值, 后者是在多实例 EKF 中与当前实例的组合测试率有最大差值的测试率。通过前者, 我们可以知道各个 24-EKF 实例的质量状况, 通过后者, 我们可以知道最佳候选实例。当最佳候选实例超过 10 s 稳定好于当前工作实例, 则将其切换为当前工作实例。这样, 外部用户就可以得到连续且稳定的最佳估计值。

18 GPS 的应用

在 ECL 估计系统中, GPS 有着非常重要且特殊的地位。在这里, GPS 有三种应用: 一是在主要的 24-EKF 中校正状态向量; 二是计算纬度/经度点之间的距离、方位角 (bearing); 三是配合全球地磁观测网站提供纬度/经度点的磁偏角测量值。第一种用途已经在前面详细讨论, 后面简略讨论后两种的算法。

18.1 计算纬度/经度点之间的距离与方位角

因为人体的高度和视野相对于巨大的地球而言十分渺小, 于是我们就误以为我们所处的大地是一个平面, 但其实地球是一个近似圆球的椭球体。在一般的短距离的应用中, 我们把地球近似为平均半径 $r = 6371 \text{ km}$ 的理想圆球。在这个假设中, ECL 选择方位等距投影 (Azimuthal Equidistant Projection) 方式计算纬度/经度点之间的距离与方位角。

方位投影 (Azimuthal Projection) 是一种地图投影, 其上所有点的方位都相对于中心正确显示。与地球两极之一相切的平面是极方位投影的基础。对于方位投影, “天顶 (zenithal)” 一词是一个更古老的术语。方位投影最显著的特征是在图上测量的从中心到任意一点的距离是真实的。因而, 地图上以投影中心为圆心的圆在真实地球上与投影中心是等距离的。同时, 从中心出发的任意方向也是真实的。该投影常用于展示多个地理位置与指定点的距离图。

方位等距投影 (Azimuthal Equidistant Projection) 是一种面积既不相等 (equal-area) 也不共形 (conformal) 的方位投影。等距方位投影可以保留距中心点的距离和方向。等距方位投影是假想球面与平面相切, 有极方位投影、赤道投影和斜轴投影三种, 分别是: 切于极点为正轴, 切于赤道为横轴, 切于极点和赤道之间的任意点为斜轴。经纬线形式同一般方位投影, 只是在中央经线上纬线间隔相等。其特点是: 由切点至任一方向的距离同实地相符; 最大角度和面积变形均为以切点为圆心的同心圆。这种投影常用于半球图, 交通图等。

ECL 采用的是局部方位等距投影 (Local Azimuthal Equidistant Projection), 也就是斜轴投影。其直觉与我们应用李群和李代数的关系类似, 在这里进行的是地理坐标系统 (geographic coordinate system) 和方位等距平面 (azimuthal equidistant plane) 之间的变换。我们以飞行器当前点为切点/原点建立局部水平面, 亦称局部方位等距平面 (local azimuthal equidistant plane)。在局部水平面中, 从原点出发的射线都是等距, 射线的方位角保真, 我们在此向量空间中进行向量计算, 计算出到达下一个航路点 (waypoint) 的距离与方位角, 然后收回到球面上, 就得到下一个航路点的纬度/经度, 或者是反过来计算, 将下一航路点的纬度/经度展开到当前水平面中, 由此计算飞行距离与方位角。到达下一个航路点后, 我们再在这个航路点上建立新的局部水平面, 并做下一步的计算。

计算纬度/经度点之间的距离与方位角的算法很古老也很成熟, 其中还涉及到基于向量的大地测量学知识。感兴趣的读者可以查看相关的书籍和网站。

18.2 提供纬度/经度点的磁偏角

这里的算法十分简单, 就是从开发和分发地球磁场模型的网站获取地磁数据档案, 建立查询表格。然后根据当前的纬度/经度, 查询表格, 获取当前点的磁偏角的测量值 $\psi^{DECLINATION}$, 将其提供给 24-EKF 进行方位校正。参见第 6.1.2 节的内容。另外, 双天线 GPS 可以提供航向角的测量值 ψ_{meas} , 参见第 6.3.1 节的内容。

完整的地磁数据档案可以取得当前点的磁偏角 (magnetic declination)、磁倾角 (magnetic inclination)、磁强度 (magnetic strength), 我们由此使用理论磁场向量的知识来计算合成磁场 z 轴的分量值。如果传感器测量中存在大量干扰, 我们可以用此 z 轴理论值替换现场测量的 z 轴的数值以合成磁场测量值, 这对于在第 6.1 节“磁力计融合”中的算法抗磁场干扰会非常有用。注意此时需要将 z 轴的新息和协方差设置为 0。

19 惯性航位推算

航位推算 (dead reckoning) 指的是无天文观测辅助导航, 或者是基于很少或没有信息的位置估计。在 ECL 的估计系统中, 当 GPS 类的全局定位数据不可用的情况下, 24-EKF 仅使用惯性数据进行航位推算, 并且大多执行的是水平惯性航位推算 (horizontal inertial dead reckoning)。这是因为存在磁力计就可以校正姿态, 存在气压计或测距仪就可以校正 z 轴高度, 但是只有 GPS 类的全局定位数据, 才能校正飞行器水平面的位置。当全局定位数据丢失或误差过大, 则 ECL 估计系统就进入航位推算模式。

能够提供水平面的位置与速度校正的数据源如下:

1. GPS 提供 \mathbf{P}_{NED} 和 \mathbf{V}_{NED} 。
2. 光流提供水平面的速度 V_N 和 V_E 。
3. 外部视觉提供 \mathbf{P}_{NED} 和 \mathbf{V}_{NED} 。
4. 风场提供 V_{wind_N} 和 V_{wind_E} , 以及相关的合成侧滑 β 。

例如, 如果存在光流则可以校正北/东方向速度, 从而间接校正了水平面的增量位移, 因此 ECL 估计系统不会进入航位推算模式。注意, 只有前 3 种的数据源完全可信, 可以校正水平面位置, 而风场速度因为误差太大, 只能是部分可信。

在没有上述前 3 种数据源的情况下, 航位推算模式还分为两种级别。第一种是存在水平风场数据, 则进入风场航位推算 (wind dead reckoning) 模式, 此时可以用水平风场向量校正北/东方向速度, 从而间接校正了水平面的增量位移, 这是第 6.4 节“空速融合方程”和第 6.5 节“合成侧滑融合方程”的内容。第二种是连水平风场向量都没有的情况下, 则是完全的水平惯性航位推算。

基于惯性的航位推算, 根据经验, 水平面位置的估计值, 大约最长有 5 ~ 10 s 的有效期。所以在 ECL 估计系统中, 设置 5 s 的有效期阈值, 并且超过 7 s 就将状态向量中的水平位置、高度和速度重置。并且估计系统还将使用第 6.10 节“虚拟位置融合方程”和第 6.11 节“虚拟高度融合方程”的算法以约束惯性导航系统的状态漂移。

20 总结

ECL EKF 算法最大的特点就是使用传感器组合进行状态估计。这使得系统具有很高的容错性, 即对传感器故障的容忍度, 但也使得算法具有更大的复杂度。但是在代码里, 更多的代码是对数据和协方差矩阵质量判断, 并对上下限进行钳制处理, 发现异常后更换数据源的处理。这些方法和经验数据, 是多年的经验积累, 是在一个个项目中一点点地磨炼出来的经验总结。这才是 ECL EKF 模块最有价值的地方, 因为主要算法可以在短时间内掌握, 而这些知识只有靠时间来积淀。

不过 ECL EKF 算法也有值得探讨的地方。数学上已经清楚，旋转 (包括四元数和旋转矩阵) 不可以直接用卡尔曼滤波器进行估计，因为旋转是群而不是向量，而卡尔曼滤波器里面最核心的概念 — 协方差 — 只能处理向量。因此直接把四元数放入卡尔曼滤波器中计算，估计的是四元数而不是旋转。

此外，对比误差状态 EKF 算法，就会发现这里的 Jacobian 矩阵展开后十分复杂，只要涉及旋转 (包括四元数和旋转矩阵) 的方程的 Jacobian 矩阵都很复杂，并且难以解释。而在误差状态 EKF 算法中，误差大多在零点附近抖动，因此有较好的近线性特性，因此 Jacobian 矩阵和标准的 KF 的矩阵很类似，因此计算速度会快一些，理论上精度也会高一些，并且容易解释。

但是，ECL EKF 模块已经在广大的产品和项目里得到验证，不论是在模拟器里还是在实际环境中，它的计算速度和精度都经受了检验。对此只能说卡尔曼滤波器真是一个很神奇的东西，在工程中实用为上。

21 参考文献

1. PX4: Estimation & Control Library for Guidance, Navigation and Control Applications - EKF
2. PX4 - EKF - documentation
3. PX4 - EKF - python code
4. PX4 - EKF - matlab code
5. Using the ECL EKF
6. EKF2 Estimation System
7. Extended Kalman Filter Navigation Overview and Tuning
8. Extended Kalman Filter (EKF)
9. NavEKF Change Overview 2020
10. PX4 State Estimation 2021 Update
11. Extended Kalman Filter
12. Recursive Attitude Estimation in the Presence of Multi-rate and Multi-delay Vector Measurements
13. PX4 的 ECL EKF 公式推导及代码解析
14. PX4 InertialNAV Filter 与 EKF2 概论