

论文提出一个“面向应用的集成处理器（ASIP）”，用于处理 9D IMU（陀螺 + 加速度计的 6D，再加磁罗盘）下的角度/姿态估计；系统在 FPGA 上实现并与传感器接口，输出既可记录到 PC，也可做实时演示。作者选择用四元数表达姿态，并用扩展卡尔曼滤波（EKF）做融合；关键创新是把滤波做成两阶段：第一阶段用加速度计修正，第二阶段用磁罗盘修正，从而获得更高灵活性、更低计算量，并对磁场异常更鲁棒。最终目标是做成更高级的 ASIC，把传感器与该处理器集成到一个封装里（磁罗盘可外接）。

段落 1（MEMS 与融合的背景）

作者说：MEMS 传感器因为低成本、小尺寸、低功耗已经大量进入消费级应用；工业/科研又希望 MEMS 系统能有更多功能、更高精度。学术界对“传感器融合算法”投入很大，因为融合能把多个传感器的原始数据加工成更有用的信息或更高精度的估计。

段落 2（IMU 高精度应用驱动）

作者进一步说：很多研究在提升陀螺精度，用于 IMU 的高精度场景（例如航迹推算/行人室内导航等）。这类系统的根本问题是：仅靠陀螺积分会漂移，需要引入其他传感器（如加速度计、磁罗盘）做校正与约束（这里后文会用 EKF/两阶段修正来解决）。

论文核心思想

- Stage 1（加速度计）：用“重力方向”纠正 roll/pitch，并显式避免影响 yaw。
- Stage 2（磁罗盘）：用“地磁方向”纠正 yaw，并显式避免磁异常污染 roll/pitch。

作者在算法总结里直接写明了这种“避免串扰”的做法：

- Stage 1：把修正四元数里与 yaw 相关的分量置零，保证不改 yaw；
- Stage 2：把与 roll/pitch 相关的分量置零，保证只改 yaw。

典型 dead reckoning 计算链路（纯 IMU 里程推算）

IMU 有两类核心量：

- 陀螺仪量到角速度 $\omega \rightarrow$ 积分得到姿态（方向/四元数/旋转矩阵）
- 加速度计量到“比力”（specific force），简单理解就是：它量到的不是纯线加速度，而是“线加速度 + 重力在传感器坐标系下的投影”等效的东西

所以典型做法是：

1. 先用陀螺估计姿态 $R(t)$
2. 根据姿态，把世界系的重力 \mathbf{g} 旋到传感器系，得到“预测的重力向量” $\hat{\mathbf{g}}_I$
3. 用加速度计测得的 $\tilde{\mathbf{a}}$ 减掉这个重力项：

$$\hat{\mathbf{a}}_{lin} = \tilde{\mathbf{a}} - \hat{\mathbf{g}}_I$$

得到“线加速度”（理想情况下）

4) 对线加速度积分得速度，再积分得位置：

$$\hat{\mathbf{v}}(t) = \int \hat{\mathbf{a}}_{lin}(t) dt, \quad \hat{\mathbf{p}}(t) = \int \hat{\mathbf{v}}(t) dt$$

Figure 1: alt text

为什么“两次积分会让误差迅速发散”

核心原因：任何微小的误差都会被积分不断累加，而且位置是“二次积分”，增长更快。

误差来源主要有两类：

A) 加速度计本身的误差（噪声、零偏）

- 速度误差 Δv 会随时间大致线性增长（偏置）或随机游走
- 位置误差 Δp 会更夸张（相当于再积分一次）

B) 更致命：姿态不准 → 重力扣错了

重力大约是 9.81 m/s^2 。如果姿态有一个很小的倾角误差 $\delta\theta$ （弧度），就会把一部分重力“误当成线加速度”：

$$\Delta a \approx g \cdot \delta\theta$$

举个量级概念： $\delta\theta = 1^\circ \approx 0.017 \text{ rad}$

$$\Rightarrow \Delta a \approx 9.81 \times 0.017 \approx 0.17 \text{ m/s}^2$$

这意味着：

- 速度误差每 1 秒就多 0.17 m/s
- 位置误差随时间呈 **二次增长**（几秒就会很离谱）

所以论文里才说：没有 GPS 或者没有速度/运动约束，只能短时间可靠。


 A Double-Stage Kalman Filter fo...

Figure 2: alt text

已有研究指出“主要误差源不是加速度噪声，而是重力向量估计错”

作者引用一项对现有 dead reckoning 单元的分析：主要误差源并非加速度噪声，而是用于相减的重力向量不准（根因是姿态估计不准）。因此本文把重点放在“更好的姿态估计系统”。

为什么只靠陀螺积分不行（漂移增长规律）

如果仅把陀螺输出积分成角度，误差会累积：

角随机游走（白噪声）导致角度误差随 \sqrt{t} 增长；

偏置不稳定、常值偏置（标定误差、温漂、焊接/封装应力等）导致误差随 t 线性增长。所以必须引入校正机制。

为什么典型距离是 \sqrt{N} 不是 N ?

关键看“能偏离 0 多远”的典型量级，也就是标准差。

对上面这个 ± 1 的例子：

- 每一步方差是 1
- 累加 N 步，方差会相加：

$$\text{Var}(S_N) = N$$

所以标准差：

$$\sigma_{S_N} = \sqrt{N}$$

这意味着：走 100 步，你通常会离 0 大约 10 左右；走 10,000 步，通常离 0 大约 100 左右。

不是线性涨，是“随机走路”那种涨。

Figure 3: alt text

更直观的说法

- **偏置**：每一步都朝同一个方向错一点 \rightarrow 越积越偏 ($\propto t$)
- **白噪声**：每一步随机向左/向右错一点 \rightarrow 不会一直抵消，走久了“离原点的典型距离”变大 ($\propto \sqrt{t}$)

Figure 4: alt text

为什么用四元数 + 融合（Kalman 最常见）

作者回顾：多数姿态估计系统用四元数而不是欧拉角，因为四元数更灵活、没有奇异点（万向节锁）、也更利于线性化；需要时也更容易转成旋转矩阵或欧拉角序列。为了抑制陀螺漂移，文献普遍使用传感器融合，其中 Kalman 滤波最常见。

相关工作对比（EKF/UKF、磁罗盘、简化思路）

作者列出典型路线：

有工作用 EKF，把状态里放“四元数 + 角速度 + 陀螺漂移”，能很好修正 roll/pitch，但 yaw 修正较弱；要 yaw 全修正通常需要磁罗盘。

也有工作用 9D（磁 + 角速率 + 重力）传感器做全姿态；为了简化 KF，先用额外算法（如 Gauss-Newton、QUEST）估计四元数，使后续可用线性 KF。

还有自适应增益 KF：在高加速度或静止状态下调整校正强度。

有人用 UKF 认为更准且实现成本低；但也有对比指出 EKF 与 UKF 精度相近，而 UKF 计算量明显更大。

本文定位与贡献（单封装目标 + 两阶段滤波的工程动机）

作者指出：上述多数方案是“分立传感器 + MCU/PC 跑算法”。本文目标是把系统做得更“单封装”：基于 SensorDynamics 的 SD746 6D IMU 原型（陀螺 + 加速度计），实现一套主要依赖 6D 的姿态估计系统，并可选外接磁罗盘。

核心算法是两阶段滤波：

- 可开关第二阶段（磁罗盘校正），带来更强的运行灵活性；
- 仅用加速度计能稳定估计 roll/pitch；加上磁罗盘才能把 yaw 也校正好；
- 两阶段拆分后矩阵更小、算力需求更低；
- 遇到磁异常时，影响被“隔离”到 yaw，roll/pitch 不会被磁干扰污染。

最后，作者给出硬件目标：把姿态估计系统集成进控制陀螺 + 加速度计的 ASIC，磁罗盘可通过 I²C 两线接口外接；并给出面积约束 0.5 mm²。

和 EKF 的区别

同一个滤波器，同一个状态 x （比如 q, b_g ），同一个协方差 P ：

1. 预测（用陀螺）：

$$x^-, P^-$$

2. 加速度计更新（重力观测）：

$$x_1, P_1 = \text{EKF_update}(x^-, P^-, z_{acc})$$

3. 磁力计更新（地磁观测）：

$$x^+, P^+ = \text{EKF_update}(x_1, P_1, z_{mag})$$

看起来也是“两步”，但关键是：**第二步（磁力计更新）是在同一个 x, P 上做的**，因此它在数学上可以通过协方差耦合影响所有姿态自由度（实现上你不做额外限制的话）。

也就是说，**mag 更新不仅仅“修 yaw”，它可能也会动 roll/pitch**（尤其当磁场矢量与姿态误差耦合、或线性化点有误差时）。

Figure 5: alt text

DSKF 的核心思想是：

- **Stage 1: 用 acc 把 roll/pitch 拉住** (重力只给俯仰/横滚参考)
- **Stage 2: 用 mag 主要修 yaw**, 并且在滤波结构/误差参数化上限制它对 roll/pitch 的影响范围 (或者至少弱耦合), 从而:
 - 磁干扰来了 → 主要污染 yaw, 不太拖坏 roll/pitch
 - 计算量也可降 (两级各自维度更小/更简单)

你可以把它理解为工程上的“分而治之：先稳住重力平面，再决定航向怎么修”。

Figure 6: alt text

- EKF: 一个滤波器里同时估计姿态/偏置, acc/mag 都是观测; 就算按顺序更新, 它们仍在同一 P 里耦合, mag 可能影响 roll/pitch。
- DSKF: 明确分两级: 先用 acc 稳 roll/pitch, 再用 mag 主要修 yaw, 并尽量隔离磁异常对 roll/pitch 的影响。

Kalman 框架：离散状态方程（从式 (1) 开始）

KF 的离散预测形式（式 (1)）

按照 KF 理论，需要定义离散时间的状态演化：由上一时刻状态 x_{k-1} 通过状态转移矩阵 A_k 演化，并可叠加输入 u_k 通过 B_k 进入系统。论文给出预测（a priori）形式：

$$\hat{x}_k^- = A_k \hat{x}_{k-1} + B_k u_k \quad (1)$$

这里：*⁻ 上标表示先验/预测值 * \hat{x} 表示真实状态未知、滤波器提供的是估计

为什么状态只选四元数（不把 bias 等扩维）

作者明确：状态方程里只用表示姿态的四元数做系统状态。加入更多变量（如角速度、漂移）并不会显著提高姿态精度，但会显著增加矩阵规模与计算量；加入 extra state 可能用于建模陀螺 bias，但仿真显示精度提升小于 1° ，不值当。

姿态表示：四元数与欧拉角

四元数定义

姿态用四元数表示：

$$q = [q_0, q_1, q_2, q_3]^T$$

其中 q_0 是实部， $[q_1, q_2, q_3]^T$ 是向量部 v 。

轴角与四元数关系

绕单位旋转轴 v 旋转角 α 的旋转可用：

$$q = \cos(\alpha/2) + \sin(\alpha/2)v \quad (3)$$

向量部给出旋转轴方向，实部对应旋转角信息

正确的维度展开写法

四元数通常写成：

$$q = \begin{bmatrix} q_0 \\ q_v \end{bmatrix} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}$$

其中：* $q_0 = q_w$ 是标量部 * $q_v = [q_x, q_y, q_z]^T$ 是向量部 (3×1)

轴角 (v, α) 到四元数的映射就是：

$$q_w = \cos(\alpha/2)$$

$$\mathbf{q}_v = v \sin(\alpha/2)$$

合起来：

$$q = \begin{bmatrix} \cos(\alpha/2) \\ v \sin(\alpha/2) \end{bmatrix}$$

这才是严格的 4×1 形式。

Figure 7: alt text

单位四元数约束与范数

要正确表示旋转，四元数必须是单位范数。范数：

$$|q| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (4)$$

(工程实现里通常每次更新后要做归一化以抑制数值漂移。)

四元数到欧拉角

文中选用 XYZ 欧拉角序列，并给出 roll ϕ 、pitch θ 、yaw ψ 的标准转换：

$$\phi = \text{atan2}\left(2(q_2q_3 + q_0q_1), q_0^2 - q_1^2 - q_2^2 + q_3^2\right) \quad (5)$$

$$\theta = \arcsin\left(\frac{2(q_0q_2 - q_1q_3)}{q_0^2 + q_1^2 + q_2^2 + q_3^2}\right) \quad (6)$$

$$\psi = \text{atan2}\left(2(q_1q_2 + q_0q_3), q_0^2 + q_1^2 - q_2^2 - q_3^2\right) \quad (7)$$

Figure 8: 四元数到欧拉角转换公式

连续时间状态方程：四元数运动学

四元数微分方程

系统的连续时间状态方程写为：

$$\dot{q}_n^b = \frac{1}{2}\Omega_{nb}^n q_n^b$$

- q_n^b : 表示 **机体系 b 相对导航/惯性系 n** 的姿态四元数（也就是“从 n 到 b ”或“从 b 到 n ”的某种约定；论文用的是上标下标来标坐标系）
- \dot{q} : 四元数对时间的导数（四元数在转动，所以它在变）
- $\Omega(\omega)$: 一个 **4×4 的矩阵**，由陀螺测得的角速度 $\omega_x, \omega_y, \omega_z$ 组成

Figure 9: alt text

Ω 的具体形式与陀螺角速度进入方式

式(9)把 $\Omega(\omega)$ 写成显式矩阵：

$$\Omega_{nb}^n = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$$

含义就是：把陀螺输出的 $\omega_x, \omega_y, \omega_z$ 填进去，就能算出 \dot{q} 。

Figure 10: alt text

它不是“凭空编出来的矩阵”，而是把**四元数乘法公式**写成“矩阵乘法”的结果。来源只有一个：**四元数微分方程**

$$\dot{q} = \frac{1}{2} q \otimes \omega_q \quad \text{或} \quad \dot{q} = \frac{1}{2} \omega_q \otimes q$$

取哪一个（左乘/右乘）取决于你用的坐标系约定，所以矩阵里正负号会略有差异；你截图那版对应其中一种约定。

下面给你“推导骨架”，你按这个思路就能理解它怎么出来。

Figure 11: alt text

1) 先把角速度写成“纯四元数”

把陀螺角速度 $\omega = [\omega_x, \omega_y, \omega_z]^T$ 写成四元数：

$$\omega_q = \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

Figure 12: alt text

2) 写出四元数乘法公式（这是关键）

设

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \quad p = \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

则 $q \otimes p$ 的分量为：

$$\begin{aligned} (q \otimes p)_0 &= q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3 \\ (q \otimes p)_1 &= q_0 p_1 + q_1 p_0 + q_2 p_3 - q_3 p_2 \\ (q \otimes p)_2 &= q_0 p_2 - q_1 p_3 + q_2 p_0 + q_3 p_1 \\ (q \otimes p)_3 &= q_0 p_3 + q_1 p_2 - q_2 p_1 + q_3 p_0 \end{aligned}$$

Figure 13: alt text

3) 把 p 换成 ω_q , 并把结果“改写成矩阵乘法”

令 $p = \omega_q = [0, \omega_x, \omega_y, \omega_z]^T$, 代进去:

$$q \otimes \omega_q = \begin{bmatrix} -(q_1\omega_x + q_2\omega_y + q_3\omega_z) \\ q_0\omega_x + q_2\omega_z - q_3\omega_y \\ q_0\omega_y - q_1\omega_z + q_3\omega_x \\ q_0\omega_z + q_1\omega_y - q_2\omega_x \end{bmatrix}$$

注意: 这可以写成一个 4×4 矩阵乘以 q :

$$q \otimes \omega_q = \underbrace{\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}}_{\Omega(\omega)} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

这! 就是你截图里的 $\Omega(\omega)$ 。

最后再乘上 $1/2$:

$$\dot{q} = \frac{1}{2} \Omega(\omega) q$$

Figure 14: alt text

$\dot{q}_n^b = \frac{1}{2} \Omega_{nb}^n q_n^b$ 连续系统把导数展开

$$\dot{q}_n^b = \lim_{T \rightarrow 0} \frac{q_n^b(t+T) - q_n^b(t)}{T} = A_{TC} q_n^b(t)$$

如果把它写成线性时变系统

$$\dot{q}(t) = A(t)q(t), \quad A(t) = \frac{1}{2} \Omega(w(t))$$

其中 $A_{TC} = \frac{1}{2} \Omega_{nb}^n$

用离散步长 T 得到离散更新

把数字系统里的固定采样周期 T 代入, 得到:

$$q_{bn}(t+T) = q_{bn}(t) + A_{TC} q_{bn}(t)T = (I + A_{TC}T)q_{bn}(t)$$

离散状态转移矩阵 A_k

因此离散矩阵为：

$$A_k = (I + A_{TC}T) = (I + \frac{1}{2}\Omega_{nb}^n T)$$

归一化说明

状态方程不会保持四元数范数。而姿态只能由单位四元数正确表示，所以要加一个归一化单元保证数据正确性。

就是每次预测或更新后做：

$$q \leftarrow \frac{q}{\|q\|}$$

从预测进入校正

- 第一阶段：用加速度计
- 第二阶段：用磁罗盘

标准校正方程与残差 residual 定义

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

- z_k 是实际测量（这里就是加速度计或者磁罗盘）
- $H\hat{x}_k^-$ 是先验状态对应的“期望测量”
- 二者的差值叫做残差 residual，残差乘以 K_k 得到校正因子，从而得到后验估计 \hat{x}_k （概率意义下的最小二乘解）

先验协方差

为了计算卡尔曼增益，先要算先验误差协方差 P_k^-

$$P_k^- = A_k P_{k-1} A_k^T + Q_k$$

卡尔曼增益

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$$

后验协方差

$$P_k = (I - K_k H_k) P_k^-$$

Stage1：用加速度计修正 roll/pitch

用当前的“先验姿态四元数”去预测重力方向（期望测量），再用加速度计实测减掉期望测量得到残差；残差乘 Kalman 增益得到一个“小的修正四元数增量” q_{e1} ；为了不让加速度计把 yaw 搅乱，把 q_{e1} 的 yaw 对应分量置零。

Step0：准备输入

- 先验姿态四元数（用陀螺积分得到）

$$\hat{q}_k^- = [q_0, q_1, q_2, q_3]^T$$

- 加速度计测量：

$$z_{k1} = [a_x, a_y, a_z]^T$$

关键点：加速度计测的是 $g + a_{lin}$ 。为了把它当“重力方向”用，要归一化（并在动态大时降权/弃用）：

$$\tilde{z} = \frac{z_{k1}}{\|z_{k1}\|}$$

Step1: 用四元数预测“期望重力方向” $h_1(\hat{q})$

“把导航系重力 $[0, 0, 1]$ ” 旋到 body 系后的方向（单位向量形式）：

$$h_1(\hat{q}) = \begin{bmatrix} 2q_1q_3 - 2q_0q_2 \\ 2q_0q_1 + 2q_2q_3 \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

它输出是 3×1 ，和归一化后的加速度计方向同维度。

用四元数表示旋转时，向量变换等价于

$$v_b = R(q) v_n$$

其中 $R(q)$ 是由四元数 $q = [q_0, q_1, q_2, q_3]^T$ (q_0 标量部) 生成的旋转矩阵。

你图里的 $h_1(\hat{q})$ 本质上就是

$$h_1(q) = R(q) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

也就是 $R(q)$ 的第三列（因为乘以 $[0, 0, 1]^T$ 就是取矩阵第三列）。

1) 从旋转矩阵 $R(q)$ 推出来（最快）

在“标量在前”约定下 $q = [q_0, q_1, q_2, q_3]^T$ ，对应的旋转矩阵常用形式之一是：

$$R(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

现在乘以 $e_3 = [0, 0, 1]^T$ ：

$$R(q)e_3 = \begin{bmatrix} 2(q_1q_3 + q_0q_2) \\ 2(q_2q_3 - q_0q_1) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

Figure 15: alt text

Step2: 算残差

$$r = \tilde{z} - h_1(\hat{q}_k^-)$$

这个式子代表：测到的重力方向减去姿态预测的重力方向

Step3: 算雅可比 $H_{k1} = \frac{\partial h_1}{\partial q} \quad (3 \times 4)$

对上面的 h_1 分量逐项求导，得到：

第一行 $h_x = 2q_1q_3 - 2q_0q_2$

$$\frac{\partial h_x}{\partial [q_0, q_1, q_2, q_3]} = [-2q_2, 2q_3, -2q_0, 2q_1]$$

第二行 $h_y = 2q_0q_1 + 2q_2q_3$

$$[2q_1, 2q_0, 2q_3, 2q_2]$$

第三行 $h_z = q_0^2 - q_1^2 - q_2^2 + q_3^2$

$$[2q_0, -2q_1, -2q_2, 2q_3]$$

所以：

$$H_{k1} = \begin{bmatrix} -2q_2 & 2q_3 & -2q_0 & 2q_1 \\ 2q_1 & 2q_0 & 2q_3 & 2q_2 \\ 2q_0 & -2q_1 & -2q_2 & 2q_3 \end{bmatrix}$$

Figure 16: alt text

Step4: 算创新协方差 S 和卡尔曼增益 K

$$S = H_{k1}P_k^-H_{k1}^T + R_{k1}K = P_k^-H_{k1}^TS^{-1}$$

- P_k^- : 四元数状态协方差
- R_{k1} : 加速度计观测噪声协方差，动态越大应设越大，降低加速度校正力度

Step5: 更新四元数并归一化

标准 KF 更新：

$$\hat{q}_k = \hat{q}_k^- + Kr$$

先算四元数增量

$$q_{e1} = Kr$$

隔离 yaw

$$(q_{e1})_3 \leftarrow 0$$

更新四元数

$$\hat{q}_k = \hat{q}_k^- + q_{e1}$$

归一化：

$$\hat{q}_k \leftarrow \frac{\hat{q}_k}{\|\hat{q}_k\|}$$

协方差更新：

$$P_k = (I - KH_{k1})P_k^-$$

Stage2: 用磁力计修正 yaw

Step0: 输入 (Stage1 更新后的结果)

- 来自 Stage1 的后验四元数、协方差：

$$\hat{q}_k^{(1)}, P_k^{(1)}$$

(你也可以记成 \hat{q}_k^- / P_k^- ，只是这里的“先验”是 Stage1 后的结果。)

- 磁力计测量：

$$z_{k2} = [m_x \quad m_y \quad m_z]^T$$

工程上建议先归一化 (只用方向)：

$$\tilde{z}_{k2} = \frac{z_{k2}}{\|z_{k2}\|}$$

并在磁干扰时增大 R_{k2} (或直接跳过 Stage2)。

Figure 17: alt text

Step1: 期望磁场（由先验四元数预测）

$$h_2(\hat{q}) = \begin{bmatrix} 2q_1q_2 + 2q_0q_3 \\ q_0^2 + q_1^2 - q_2^2 - q_3^2 \\ 2q_2q_3 - 2q_0q_1 \end{bmatrix}$$

Figure 18: alt text

Step2: 残差

$$r_2 = \tilde{z}_{k2} - h_2(\hat{q}_k^{(1)})$$

Figure 19: alt text

Step3: 测量雅可比 H_{k2}

$$H_{k2} = \begin{bmatrix} 2q_3 & 2q_2 & 2q_1 & 2q_0 \\ 2q_0 & -2q_1 & -2q_2 & -2q_3 \\ -2q_1 & -2q_0 & 2q_3 & 2q_2 \end{bmatrix}$$

Figure 20: alt text

这里的 q_i 用的当前用于 Stage2 的先验 $\hat{q}_k^{(1)}$ 里的分量。

Step4: 创新协方差与卡尔曼增益

$$S_2 = H_{k2} P_k^{(1)} H_{k2}^T + R_{k2}$$
$$K_{k2} = P_k^{(1)} H_{k2}^T S_2^{-1}$$

Figure 21: alt text

Step5: 算修正量，并隔离 roll/pitch, 只保留 yaw

$$q_{e2} = K_{k2} r_2$$

Figure 22: alt text

然后做隔离：把 roll/pitch 对应的分量置零，只保留 yaw 对应那一维。

Step6: 更新四元数、归一化、更新协方差

$$\hat{q}_k^{(2)} = \hat{q}_k^{(1)} + q_{e2}$$
$$\hat{q}_k^{(2)} \leftarrow \frac{\hat{q}_k^{(2)}}{\|\hat{q}_k^{(2)}\|}$$
$$P_k^{(2)} = (I - K_{k2} H_{k2}) P_k^{(1)}$$

Figure 23: alt text

面向硬件实现的算法简化

为了实现面积占用低、功耗低的集成系统，作者对第二节给出的 KF 算法做了简化；目标是在一个简化的算术逻辑单元（ALU）上运行，该 ALU 只能做乘法与加法。

简化 1：用 P_k 近似 P_k^- （把预测协方差更新拿掉）

标准 KF 有：* 先验协方差（预测后）： P_k^- * 后验协方差（更新后）： P_k

这里假定：

$$P_k^- \approx P_k$$

直觉解释：如果采样很快、每次更新很小，那么“预测一步”并不会让不确定度涨很多——于是把 P_k^- 当成上一轮更新后的 P_k 用，误差不大。

等价于：不再做 $P^- = APA^T + Q$ 这步，直接用一个“当前的 P ”参与后面的增益计算。

简化 2： P_k 离线预设 + 启动阶段阶跃（前 128 次）

作者更进一步：既然 $P_k^- \approx P_k$ ，那干脆把 P_k 也别在线算了，直接当“调参常数”。

做法是：

- 启动阶段：用一个较大的 P_0 （表示“我一开始对姿态很不确定”）
- 前 128 次迭代后：用阶跃切换到较小的 P_1 （表示“姿态已经收敛了，可以更自信”）

形式上是：

$$P_k = \begin{cases} P_0, & k < 128 \\ P_1, & k \geq 128 \end{cases}$$

直觉解释：

- 刚开机，姿态可能是倒置/乱的，靠重力/地磁要快速拉回 → 需要“大 P ”使增益 K 更大（更信测量、拉得更猛）。
- 稳定后，为了抑制噪声/抖动 → 用“小 P ”让 K 变小（更信陀螺预测、修正更温和）。

“两个 stage 用同一个 P_k ”：就是 Stage1（加速度）和 Stage2（磁力计）都用同一套预设 P ，进一步省资源。

风险/代价：

- 这已经不算严格意义上的自适应 KF 了，更像“固定增益滤波器”（或者说 KF 退化成常系数版本）。
- 如果传感器噪声变化很大（温漂、震动、磁干扰），固定 P 可能不够鲁棒，需要靠固定的 R_1, R_2 或门限机制兜底。

Figure 24: alt text

简化 3：消除除法

卡尔曼增益的核心难点是：

$$\underbrace{(HPH^T + R)}_S^{-1}$$

如果 P 、 R 都被固定成常数（离线给定），并且 H 的结构也比较固定/可简化，那么：

- S 的很多项会变成“固定表达式”
- 对 3×3 的 S 求逆，形式上是：

$$S^{-1} = \frac{1}{\det(S)} \text{adj}(S)$$

这里的 “ $\frac{1}{\det(S)}$ ” 是除法

作者的意思是：由于参数固定， $\det(S)$ 的值在硬件实现里可以被当成**固定常数**或者被近似成常数（或用查表/常数乘法代替除法），从而把“除法”变成“乘一个常数”。

直觉解释：

把在线求逆，从“通用求逆”降级成“常系数的线性组合 + 常数乘”。

硬件里常数乘非常便宜，除法很贵。

风险/代价：

- 如果真实 S 变化很大（例如测量噪声显著变大），你固定分母会让增益不再最优，可能过度相信测量或过度相信预测。
- 所以这通常要配合“门限/检测磁干扰/高动态降权”等策略。

Figure 25: alt text

简化 4：去掉四元数归一化

严格来说，四元数必须保持 $\|q\| = 1$ 。归一化需要：

- 求和： $q^T q$ (可以乘加)
- 开方： $\sqrt{\cdot}$ (硬件难)
- 除法： $q/\|q\|$ (硬件难)

作者观察到：在他们的滤波结构与参数下，更新步骤本身会把 q 拉回到接近单位长度（误差不积累），于是声称“工作阶段可不做归一化”。

你可以把它理解为一种经验现象：

- 预测（陀螺积分）+ 观测校正（重力/地磁）会持续把姿态拉向合理区域
- 四元数模长偏离会被校正项间接抑制
- 在他们的仿真/实验范围里足够稳定

风险/代价：

- 这不是数学保证，只是实验现象。换传感器、换噪声、换动态，可能就会漂离单位范数。
- 工程上更稳妥的替代：偶尔归一化（比如每 N 次做一次），或用近似归一化（牛顿迭代一两步，仅乘加）。

Figure 26: alt text