RoboMaster
视觉从入门到入土

Misaka21

Turn drawings of imagination into racetrack realization.

June 25, 2025

Thank you for using this book ❤️,

I hope you like it 😄

目录

1. 基础篇

我们在基础篇

1.1. 计算机系统

1.2. C++基本语法

1.3. 软件工程基础

1.4. ROS/ROS2

> Euler's identity is $e^{\pi i} + 1 = 0$. Do you really believe that they charged an armed enemy, or treated their children, their own flesh and blood, so cruelly, without a thought for their own interest or advantage? Such is Schrödinger's equation in
>
> $$i\hbar\frac{\partial}{\partial t}\Psi(x,t) = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t)\right]\Psi(x,t)$$
>
> > But who has any right to find fault with a man who chooses to enjoy a pleasure that has no annoying consequences, or one who avoids a pain that produces no resultant pleasure?

But I must explain to you how all this mistaken idea of reprobating pleasure and extolling pain arose. Increase ease-of-use to where `variable` and `print()` shall be of use.

```python
if a != b:
  print("Hello world!")
else if a == b:
  print("Goodbye world!")
else:
  print("This is a long sentence where I ramble until I get 80 characters here.")
```

代码 1　Example `python` code printing text.

2. 数学理论篇

2.1. 计算机视觉基础

2.2. 传统视觉算法

2.3. 深度学习和神经网络

2.4. 相机模型

2.5. 坐标变换

2.6. 卡尔曼滤波

Definition 2.6.1

本章使用的符号规定：

- **粗体小写字母** 表示向量，例如 $\boldsymbol{x}$
- **粗体大写字母** 表示矩阵，例如 $\boldsymbol{A}$
- **普通小写字母** 标量或向量的分量
- **普通大写字母** 表示矩阵元素

2.6.1. 什么是卡尔曼滤波?

2.6.2. 状态空间方程

状态空间方程是现代控制理论的基础, 它以矩阵的形式表达系统的状态变量, 输入及输出的关系。它可以描述和处理多输入多输出的系统。

2.6.3. 如何准确地测量体重

假设存在一个测量精度有限的体重秤。在实际测量过程中,该设备会产生随机误差,导致单次测量结果与真实体重之间存在偏差。为提高测量精度, 可以通过多次采样与数据处理的方法获得更稳定的估计值。

首先, 对同一对象进行 n 次测量，每次记录体重秤的显示值。由于体重秤的误差通常呈随机分布，重复测量可以在一定程度上抵消单次测量的偏差。随后，将全部测量值求平均，得到较为稳定的估计值：

$$\hat{x}_n = \frac{1}{n}(z_1 + z_2 + \cdots + z_n) = \frac{1}{n}\sum_{i=1}^{n} z_i$$

上述操作虽然在数学层面较为直观，但在工程应用中存在实现困难。根据平均值的定义，为了计算 $\hat{x}_n$，需要存储所有历史测量值 $z_1, z_2, \cdots, z_n$。当 n 较大时，会产生显著的内存开销，对嵌入式设备而言尤为不利。此外，每次获得新的测量值后都需要重新遍历全部历史数据进行计算，其时间复杂度为 $O(n)$，会对处理器造成不必要的负担。

因此，需要寻找一种递推（Recursive）形式，使得算法满足以下特性：无需保存全部历史数据，无需重复计算，仅依赖上一次估计 $\hat{x}_{n-1}$ 和当前测量 z_n

为实现上述目标, 可将平均值公式改写为状态更新方程。通过代数变换, 得到以下递推形式：

$$\hat{x}_n = \hat{x}_{n-1} + \frac{1}{n}(z_n - \hat{x}_{n-1})$$

该表达式揭示了递推估计的核心机制：第 n 时刻的状态估计 $\hat{x}_n$ 由两部分构成——前一时刻的估计值 $\hat{x}_{n-1}$ 以及基于当前测量值 z_n 的修正项 $(z_n - \hat{x}_{n-1})$。因此,算法无需保存全部历史数据,仅需利用上一时刻的估计结果即可完成当前状态的更新,显著降低了计算复杂度（降至 $O(1)$）与存储需求。

详细推导 Note 2.6.3.1

$$\begin{aligned}
\hat{x}_n &= \frac{1}{n}\sum_{i=1}^{n} z_i = \frac{1}{n}\left(\sum_{i=1}^{n-1} z_i + z_n\right) \\
&= \frac{1}{n}\sum_{i=1}^{n-1} z_i + \frac{1}{n}z_n = \frac{n-1}{n(n-1)}\sum_{i=1}^{n-1} z_i + \frac{1}{n}z_n \\
&= \frac{n-1}{n}\cdot\frac{1}{n-1}\sum_{i=1}^{n-1} z_i + \frac{1}{n}z_n = \frac{n-1}{n}\hat{x}_{n-1} + \frac{1}{n}z_n \\
&= \hat{x}_{n-1} - \frac{1}{n}\hat{x}_{n-1} + \frac{1}{n}z_n \\
&= \hat{x}_{n-1} + \frac{1}{n}(z_n - \hat{x}_{n-1})
\end{aligned}$$

符号	含义
x	体重的真值
z_n	第 n 次对体重的测量值
$\hat{x}_n$	基于前 n 次测量值,对 x 的估计值
$\hat{x}_{n-1}$	基于前 $n-1$ 次测量值,对 x 的估计值
$z_n - \hat{x}_{n-1}$	测量残差：当前测量值与先验估计之间的偏差

note 2　详细推导

至此，上述平均值递推公式已经可以被视为一种特殊形式的状态估计滤波器。其中，增益系数

$$K_n = \frac{1}{n}$$

决定了“当前观测”与“历史估计”在融合时各自的权重。该系数随测量次数增加而递减，表明随着可用数据增多，单次新测量对整体估计的影响逐渐减小。本例中，随着n的增加，$\frac{1}{n}$会下降。在一开始，因为没有足够的信息，第一次估计完全是基于第一次的测量值的$\frac{1}{n}|_{n=1} = 1$. 随着迭代进行，每次后续测量的权重都在下降，并且会逐渐变得可以忽略不计。这一结构与卡尔曼滤波的状态更新公式在形式上完全一致：

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k\left(z_k - h\left(\hat{x}_{k|k-1}\right)\right)$$

其中：$\hat{x}_{k|k-1}$ 为基于前 $k-1$ 次观测对第 k 时刻的预测值（先验估计）z_k 为第 k 次测量值$h(\cdot)$ 为观测函数，将状态映射到观测空间K_k 为卡尔曼增益，动态调节预测值与测量值的融合比例

在体重秤的简化场景中，系统状态不随时间变化（$\hat{x}_{k|k-1} = \hat{x}_{k-1}$），且观测函数为恒等映射（$h(x) = x$），因此递推平均公式可视为卡尔曼滤波在静态系统、确定性观测条件下的特例。

2.6.4. 从递推平均到滤波器结构

在前一章中，已经推导出递推平均的更新公式：

$$\hat{x}_n = \hat{x}_{n-1} + \frac{1}{n}(z_n - \hat{x}_{n-1})$$

表面上看，该公式仅是“多次求平均”的递推写法。然而，对其结构进行深入分析后可以发现，它实际上已经包含了滤波器（Filter）的基本要素。

2.6.4.1. 递推公式的结构解析

将上述公式重新标注为：

$$\hat{x}_n = \underbrace{\hat{x}_{n-1}}_{\text{先验估计}} + \underbrace{\frac{1}{n}}_{\text{增益系数}} \underbrace{(z_n - \hat{x}_{n-1})}_{\text{测量残差}}$$

这一形式揭示了三个关键组成部分：

先验估计（Prior Estimate）

$\hat{x}_{n-1}$ 代表在获得第 n 次测量之前，系统对状态的当前认知。对于静态系统（状态不随时间变化），先验估计即为上一时刻的后验估计。该项体现了系统对历史信息的继承。

测量残差（Measurement Residual）

$r_n = z_n - \hat{x}_{n-1}$ 反映了观测值与先验估计之间的偏差。若测量值高于估计值，则 $r_n > 0$；反之则 $r_n < 0$。测量残差是滤波器进行状态修正的驱动信号。

增益系数（Gain Coefficient）

$K_n = \frac{1}{n}$ 决定了测量残差在状态更新中所占的权重，是滤波器设计中的核心参数。

上述“先验估计 + 增益 × 残差”的结构，正是所有递推估计算法的通用框架。

2.6.4.2. 增益系数的物理意义

增益系数 K_n 的大小直接决定了滤波器对新信息的敏感程度：

- K_n 较大时，系统更倾向于信任当前测量值，历史估计的影响被削弱
- K_n 较小时，系统更倾向于保持先验估计，新测量的影响受限

递推平均的增益为 $K_n = \frac{1}{n}$，具有以下特性：

随着测量次数 n 增加，K_n 单调减小。在第一次测量时，$K_1 = 1$，意味着估计值完全由第一次测量决定。随着迭代进行，每次新测量的权重逐渐降低：第二次测量的权重为 $\frac{1}{2}$，第十次为 $\frac{1}{10}$，依此类推。而当 $n \to \infty$ 时，$K_n \to 0$，新增测量几乎无法改变估计值。此时系统处于“过度平滑”状态，对新信息的响应速度极慢。

从频域角度看，递推平均可视为一个低通滤波器，其截止频率随 n 增加而降低。这种特性在处理静态参数估计时是合理的，但对于动态系统则可能导致跟踪滞后。

因此引出了一个核心问题：能否根据系统动力学特性与噪声统计特性，自适应地确定增益系数？这一问题的答案正是卡尔曼滤波的核心贡献——通过最小化估计误差协方差，动态计算最优增益 $\boldsymbol{K}_k$？

2.6.5. 从静态估计到动态系统模型

2.6.5.1. 静态估计的局限性

在前一章中，通过递推平均的方法实现了对体重的高效估计。该方法基于一个关键假设：**被测量的物理量在整个测量过程中保持不变**。

然而，这一假设并不是在所有应用中都成立。考虑以下场景：

场景一：体重测量

当一个人站在体重秤上时，其体重在测量过程中基本保持恒定。即使进行多次测量，真实体重也不会发生显著变化。此时，所有测量误差均来自传感器噪声，递推平均方法能够有效工作。

场景二：赛车位置追踪

对于高速行驶的赛车，其位置每秒可能变化数十米甚至上百米。若仍使用递推平均方法，算法会持续“追赶”真实位置，但永远落后于实际状态。当赛车加速、转弯或制动时，这种滞后会更加明显。

场景三：无人机姿态估计

无人机在飞行过程中，其姿态角（俯仰、偏航、翻滚）会随操控指令快速变化。若不考虑姿态的动态演化规律，仅依靠传感器测量进行平均，将无法实现稳定的姿态控制。

上述对比揭示了静态估计方法的根本缺陷：**当系统状态本身随时间变化时，单纯融合测量数据已无法准确追踪真实状态**。为解决这一问题，必须引入**预测机制**——在测量到来之前，根据系统的动态规律主动推断未来状态。

换言之，对于动态系统而言：

不预测，就不能准确估计。

2.6.5.2. 预测什么？从单一变量到状态向量

既然预测是必需的,那么接下来需要回答一个更基本的问题:**预测的对象是什么?**

回到赛车追踪的例子。假设当前时刻赛车位于 100 米处,需要预测 1 秒后它会在哪里。仅凭“当前位置是 100 米”这一信息,能否完成预测?

答案是否定的。因为:

- 若赛车静止不动,1 秒后仍在 100 米处
- 若赛车以 10 m/s 行驶,1 秒后在 110 米处
- 若赛车以 50 m/s 行驶,1 秒后在 150 米处

可见,**仅知道位置无法预测未来位置**。必须同时知道速度,才能推断物体下一时刻会移动到何处。

同样地,仅知道速度也不够。速度告诉我们“每秒移动多少米”,但没有当前位置作为起点,同样无法确定未来位置的绝对值。

因此,要预测运动物体的未来状态,必须**同时跟踪位置和速度**。更一般地说,需要跟踪所有对系统未来行为有影响的变量。这些变量的集合,就构成了系统的**状态**。

2.6.5.3. 状态向量：描述系统的完整信息

为描述动态系统的行为，需要定义**状态（State）**。状态是能够完整刻画系统当前特性的变量集合。对于运动物体，仅知道位置是不够的——若不知道速度，就无法预测下一时刻位置会变化多少。

因此，对于一维匀速运动，状态向量定义为：

$$\boldsymbol{x}_k = \begin{pmatrix} p_k \\ v_k \end{pmatrix}$$

其中：

- p_k 表示第 k 时刻的位置（Position）
- v_k 表示第 k 时刻的速度（Velocity）

将位置和速度放在一个向量中是自然的选择：

- **位置会变**：物体不断移动，位置随时间更新
- **速度会变**：物体可能加速或减速，速度本身也是动态量
- **两者相关**：速度决定了位置的变化率

使用状态向量的目的，是为了让系统可以通过统一的矩阵形式组织、推广，并最终引出卡尔曼滤波的结构。

2.6.5.4. 最简单的动态模型：匀速运动

考虑最基本的动态系统——**匀速运动（Constant Velocity Model）**。假设物体在时间间隔 Δt 内以恒定速度运动，则有：

$$p_k = p_{k-1} + v_{k-1}\Delta t$$
$$v_k = v_{k-1}$$

第一个方程表明：新位置等于旧位置加上“速度乘以时间”。这是高中物理中的基本运动学公式。

第二个方程表明：速度保持不变。这正是“匀速”的含义。

将上述方程写为矩阵形式：

$$\begin{pmatrix} p_k \\ v_k \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_{k-1} \\ v_{k-1} \end{pmatrix}$$

即：

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1}$$

其中**状态转移矩阵（State Transition Matrix）**为：

$$\boldsymbol{F} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$$

该矩阵编码了系统的动态规律：

- 第一行 $[1, \Delta t]$：位置更新公式
- 第二行 $[0, 1]$：速度保持不变

状态转移矩阵 $\boldsymbol{F}$ 是预测的核心。只要给定当前状态 $\boldsymbol{x}_{k-1}$，就可以通过 $\boldsymbol{F}$ 计算下一时刻的状态 $\boldsymbol{x}_k$。

2.6.5.5. 模型不完美：过程噪声

然而，现实世界中不存在完美的匀速运动。物体可能受到以下影响：

- 轻微的加速度波动
- 外界扰动（风阻、摩擦力、路面颠簸）
- 模型简化带来的误差（例如忽略了加速度项）

为描述这些不确定性，在状态转移方程中加入**过程噪声（Process Noise）**项：

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}$$

其中 $\boldsymbol{w}_{k-1}$ 表示第 $k-1$ 时刻的过程噪声。该噪声通常假设服从高斯分布：

$$\boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q})$$

这里的 $\boldsymbol{Q}$ 称为**过程噪声协方差矩阵**，它刻画了模型不确定性的强度：

- $\boldsymbol{Q}$ **较大**：表示系统动态变化剧烈，模型预测不够可靠（例如赛车频繁加减速）
- $\boldsymbol{Q}$ **较小**：表示系统演化较为稳定，模型预测较为准确（例如匀速直线行驶）

过程噪声的引入使得模型更加贴近现实。它承认了一个事实：**任何数学模型都无法完美描述真实系统**。

2.6.5.6. 预测与更新：卡尔曼滤波的双重机制

至此，已建立了动态系统的基本描述框架。卡尔曼滤波正是基于这一框架，通过两个交替执行的步骤实现状态估计：

步骤一：预测（Prediction）

利用状态转移方程，根据上一时刻的估计值预测当前时刻的状态：

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1|k-1}$$

此时尚未获得新的测量数据，预测完全依赖系统的动态模型。

步骤二：更新（Update）

当新的测量值 $\boldsymbol{z}_k$ 到达后，将其与预测值融合，得到更优的状态估计：

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{z}_k - \boldsymbol{H}\hat{\boldsymbol{x}}_{k|k-1}\right)$$

其中 $\boldsymbol{H}$ 为**观测矩阵**，用于将状态空间映射到测量空间。例如，当状态包含位置和速度，但传感器只能测量位置时，$\boldsymbol{H}$ 负责从状态向量中提取出可观测的部分。

这一结构与前一章的递推平均公式完全一致，但增加了预测步骤。**预测使得算法能够主动追踪系统动态，而不是被动地对测量求平均**。

下一章，我们将首先探索如何用简单的 g 和 h 常数来确定增益 $\boldsymbol{K}_k$ (g-h 滤波器)，它决定了我们究竟该相信模型预测，还是相信传感器测量。为了透彻理解这一核心机制，下一章我们将暂时放下复杂的矩阵运算，通过“g-h 滤波器”这一简化模型，手动调节 g 和 h 两个常数增益。这将帮助我们建立对“滤波”的直观物理认识，并最终引出卡尔曼滤波是如何自动计算出最优增益的。

2.6.6. g-h 滤波器：手动调节的预测与修正

在上一章中，我们认识到动态系统估计的核心在于**预测**。仅靠测量求平均无法追踪运动物体，必须利用系统的动态规律主动推断未来状态。

本章将通过 **g-h 滤波器** 这一简化模型，建立对“预测-修正”机制的直观理解。它是卡尔曼滤波的“手动挡”版本，虽然简单，却蕴含了滤波算法的全部精髓。

2.6.6.1. 问题设定：追踪匀速运动的物体

假设需要追踪一辆在直线轨道上行驶的小车。根据上一章建立的框架：

- **状态向量**：

$$\boldsymbol{x}_k = \begin{pmatrix} p_k \\ v_k \end{pmatrix}$$

  其中 p_k 为位置，v_k 为速度。

- **动态模型**（匀速运动）：

$$p_k = p_{k-1} + v_{k-1}\Delta t$$
$$v_k = v_{k-1}$$

- **传感器测量**：GPS 每隔 $\Delta t = 1$ 秒提供一次位置测量 z_k，但测量值含有噪声。

核心问题在于：如何融合**模型预测**和**传感器测量**，得到最优的位置和速度估计？

2.6.6.2. 预测步骤：利用动态模型

在第 k 时刻的测量到达之前，先根据上一时刻的估计值进行预测。

位置预测

根据匀速运动模型：

$$p_{pred} = \hat{p}_{k-1} + \hat{v}_{k-1}\Delta t$$

这里的 $\hat{p}_{k-1}$ 和 $\hat{v}_{k-1}$ 是第 $k-1$ 时刻的最优估计（后验估计）。

假设上一时刻估计小车位于 100 米，速度为 20 米/秒，时间间隔 $\Delta t = 1$ 秒，则预测位置为：

$$p_{pred} = 100 + 20 \times 1 = 120 \ \ \textbf{米}$$

速度预测

对于匀速模型，速度保持不变：

$$v_{pred} = \hat{v}_{k-1}$$

此时，在测量到达前，先验估计为：

$$\begin{pmatrix} p_{pred} \\ v_{pred} \end{pmatrix} = \begin{pmatrix} 120 \\ 20 \end{pmatrix}$$

2.6.6.3. 测量到达：发现预测偏差

现在，GPS 测量值到达：$z_k = 115$ 米。

预测值与测量值之间存在**残差**（Residual）：

$$r_k = z_k - p_{pred} = 115 - 120 = -5 \ \ \textbf{米}$$

负号表示测量值**低于**预测值。这意味着什么？可能存在以下情况：

- 小车实际位置确实在 115 米，预测高估了
- GPS 测量有误差，小车实际就在 120 米附近
- 真实位置介于两者之间（最可能）

滤波器的任务，就是在预测和测量之间找到**合理的平衡点**。

2.6.6.4. 修正步骤：引入 g 和 h 增益

修正位置估计：g 增益

既不能完全信任测量，也不能完全信任预测。合理的做法是取加权平均：

$$\hat{p}_k = p_{pred} + g \cdot (z_k - p_{pred})$$

其中 g 称为**位置增益**，取值范围通常为 $0 < g < 1$。

物理意义：

- $g = 0$：完全信任预测，忽略测量（$\hat{p}_k = p_{pred}$）
- $g = 1$：完全信任测量，忽略预测（$\hat{p}_k = z_k$）
- $g = 0.3$：预测权重 70%，测量权重 30%

示例：选择 $g = 0.3$，修正后的位置估计为：

$$\hat{p}_k = 120 + 0.3 \times (-5) = 120 - 1.5 = 118.5 \text{ 米}$$

修正速度估计：h 增益

这是 g-h 滤波器的关键设计：**位置预测出现偏差，往往说明速度估计存在误差**。

考虑以下逻辑链：

1. 位置残差为 -5 米，说明预测位置偏高
2. 位置偏高，可能是因为**高估了上一时刻的速度**
3. 因此，应该**下调速度估计**

速度修正公式为：

$$\hat{v}_k = v_{pred} + \frac{h}{\Delta t} \cdot \big(z_k - p_{pred}\big)$$

其中 h 称为**速度增益**，通常取值 $0 < h < 1$。

为什么要除以 Δt?

残差的量纲是**米**，而速度的量纲是**米/秒**。除以 Δt 完成单位转换：

$$\frac{\textsf{位置误差}}{\textsf{时间间隔}} \approx \textsf{速度误差}$$

这个操作本质上是在估算: “位置错了 5 米, 速度应该调整多少才能在 1 秒内补偿这个误差?”

示例：选择 $h = 0.1$，$\Delta t = 1$ 秒，修正后的速度估计为：

$$\hat{v}_k = 20 + \frac{0.1}{1} \times (-5) = 20 - 0.5 = 19.5 \text{ 米/秒}$$

完整的更新过程

g-h 滤波器的一次完整迭代包括：

1. **预测**：

$$p_{pred} = \hat{p}_{k-1} + \hat{v}_{k-1}\Delta t$$
$$v_{pred} = \hat{v}_{k-1}$$

2. **修正**（测量到达后）：

$$\hat{p}_k = p_{pred} + g \cdot \big(z_k - p_{pred}\big)$$
$$\hat{v}_k = v_{pred} + \frac{h}{\Delta t} \cdot \big(z_k - p_{pred}\big)$$

2.6.6.5. g 和 h 的物理意义

增益系数的选择直接决定了滤波器的性能特性。

g 的作用：平滑性与响应速度的权衡

- **g 较小**（如 0.1）：对测量不敏感，输出平滑，但跟踪存在滞后
  适用场景：测量噪声大，系统变化缓慢（如匀速行驶的火车）
- **g 较大**（如 0.8）：快速响应测量变化，但容易受噪声干扰
  适用场景：测量精度高，系统动态变化快（如高速赛车）

h 的作用：速度修正的激进程度

- **h 较小**（如 0.05）：速度调整保守，适合长期稳定的系统

- **h 较大**（如 0.5）：速度调整激进，适合加速度变化频繁的系统

需要注意的是，g-h 滤波器的最大局限在于**需要手动调参**。不同系统、不同噪声水平需要不同的 g 和 h，这往往需要大量试错。

2.6.6.6. 矩阵形式：统一的数学框架

前面的标量公式虽然直观，但当系统维度增加时会变得繁琐。现在将其改写为矩阵形式，这不仅更简洁，也为后续的卡尔曼滤波做好准备。

状态与预测的矩阵表示

回顾上一章定义的状态向量和转移矩阵：

$$\boldsymbol{x}_k = \begin{pmatrix} p_k \\ v_k \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$$

预测步骤可以简洁地写为：

$$\boldsymbol{x}_{pred} = \boldsymbol{F}\hat{\boldsymbol{x}}_{k-1}$$

增益矩阵：打包 g 和 h

将两个增益系数 g 和 $\frac{h}{\Delta}t$ 组织成**增益向量**：

$$\boldsymbol{K}_{\text{g-h}} = \begin{pmatrix} g \\ \frac{h}{\Delta t} \end{pmatrix}$$

观测矩阵：提取可测量的分量

GPS 只能测量位置，无法直接测量速度。因此需要一个**观测矩阵 $\boldsymbol{H}$** 从状态向量中提取出位置分量：

$$\boldsymbol{H} = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

这样，预测位置可以表示为：

$$\boldsymbol{H}\boldsymbol{x}_{pred} = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} p_{pred} \\ v_{pred} \end{pmatrix} = p_{pred}$$

统一的更新公式

有了这些定义，g-h 滤波器的修正步骤可以写成统一的矩阵形式：

$$\hat{\boldsymbol{x}}_k = \boldsymbol{x}_{pred} + \boldsymbol{K}\big(z_k - \boldsymbol{H}\boldsymbol{x}_{pred}\big)$$

展开验证，看它是否还原了标量公式：

$$\begin{aligned} \begin{pmatrix} \hat{p}_k \\ \hat{v}_k \end{pmatrix} &= \begin{pmatrix} p_{pred} \\ v_{pred} \end{pmatrix} + \begin{pmatrix} g \\ \frac{h}{\Delta t} \end{pmatrix} \big(z_k - p_{pred}\big) \\ &= \begin{pmatrix} p_{pred} + g\big(z_k - p_{pred}\big) \\ v_{pred} + \frac{h}{\Delta t}\big(z_k - p_{pred}\big) \end{pmatrix} \end{aligned}$$

结果与标量公式完全一致。

这个统一的框架揭示了几个重要事实：

1. $\boldsymbol{K}$ 的物理意义清晰：第一行控制位置修正，第二行控制速度修正
2. 为卡尔曼滤波铺平道路：g-h 滤波器的 $\boldsymbol{K}$ 是**常数**；卡尔曼滤波只是增加了一步——**自动计算最优的 $\boldsymbol{K}$**

2.6.6.7. g-h 滤波器的局限性

虽然 g-h 滤波器简单直观，但存在以下根本性缺陷：

- **参数需要手动调节**：无法确定当前的 g 和 h 是否为最优
- **固定增益无法适应变化**：当系统动态特性改变时（例如赛车进入弯道），固定参数会失效
- **缺乏误差量化**：滤波器无法反馈“当前估计的可信度有多高”

g-h 滤波器已经具备了现代滤波器的基本结构：

$$\textbf{估计} = \textbf{预测} + \textbf{增益} \times \textbf{残差}$$

它唯一缺少的，是**自动计算最优增益**的机制。卡尔曼滤波正是通过以下两点解决了这个问题：

1. 通过**误差协方差矩阵** $\boldsymbol{P_k}$ 量化估计的不确定性
2. 通过**最小化估计误差**的准则，动态推导出最优增益 $\boldsymbol{K_k}$

下一章将揭开这一机制的数学原理，展示卡尔曼滤波如何将 g-h 滤波器的直观思想发展为完整的最优估计理论。

3. 实战技术篇

3.1. 通信协议设定

3.2. 相机标定与手眼标定

3.3. 时间戳对齐

3.4. 弹道解算

4. RoboMaster 应用篇

4.1. 工业相机

4.2. 装甲板识别

4.3. 装甲板的跟踪

4.4. 能量机关识别

4.5. 自瞄算法设计

4.6. 串口模块

5. 进阶篇

5.1. 雷达站视觉方案

5.2. 哨兵决策视觉

5.3. 性能优化技巧

5.4. 实战经验与坑点总结

6. 项目分析

6.1. rm_vision

6.2. rm.cv.fans

6.3. 同济自喵

Stokes' theorem Definition 6.3.1

Let Σ be a smooth oriented surface in $\mathbb{R}^3$ with boundary $\partial\Sigma \equiv \Gamma$. If a vector field $\boxed{\boldsymbol{F}(x,y,z)} = \big(F_x(x,y,z), F_y(x,y,z), F_z(x,y,z)\big)$ is defined and has continuous first order partial derivatives in a region containing Σ, then

$$\iint_{\Sigma} (\boldsymbol{\nabla} \times \boldsymbol{F}) \cdot \boldsymbol{\Sigma} = \boxed{\oint_{\partial\Sigma} \boldsymbol{F} \cdot \mathrm{d}\boldsymbol{\Gamma}}$$

Information extracted from a well-known public encyclopedia

Definition 3 Stokes' theorem

> 这是引用内容。 可以放多行文本。

Index of Listings