



中國農業大學
China Agricultural University

自动控制理论——滤波方法

胡标





目录

Contents

1. 鲁道夫·埃米尔·卡尔曼

2. 卡尔曼滤波

3. 扩展卡尔曼滤波

4. 无迹卡尔曼滤波

5. 粒子滤波

■ 鲁道夫·埃米尔·卡尔曼

英文名: Rudolf Emil Kálmán

出生: 1930年5月19日, 匈牙利布达佩斯

逝世: 2016年7月2日, 美国佛罗里达州盖恩斯维尔

国籍: 匈牙利裔美国籍



■ 生平与教育

早年经历: 卡尔曼出生于布达佩斯的一个犹太家庭。1943年, 为躲避纳粹迫害, 他随家人移居美国。

■ 教育背景:

1953年, 在麻省理工学院获得电气工程学士和硕士学位。

1957年, 在哥伦比亚大学获得博士学位, 师从著名控制理论专家约翰·拉格齐尼。

第一阶段：诞生与最初的忽视 (1960-1961)



■ 核心事件：

- 1960年，29岁的鲁道夫·卡尔曼在NASA埃姆斯研究中心访问时，完成了他的开创性论文《A New Approach to Linear Filtering and Prediction Problems》。
- 他随后在纽约举办的美国机械工程师学会（ASME）会议上首次公开介绍了这一理论。

■ 内容与突破性：

这篇论文提供了一套完整、严谨的数学框架，解决了在含有噪声的动态系统中进行最优状态估计的核心问题。

其革命性在于：

1. **递归性**：无需存储历史数据，只需根据上一时刻的估计和当前时刻的观测，就能计算出当前最优估计，计算量和存储量极小。
2. **时域法**：不同于当时主流的基于频域（维纳滤波）的方法，卡尔曼滤波直接在时域处理，更容易理解和应用于时变系统。
3. **最优性**：在线性高斯假设下，它被证明是最小方差无偏估计，是数学上最优的。

■ 最初的冷遇：

- 学术界的怀疑：许多控制理论领域的权威（如诺伯特·维纳的追随者）对这位年轻学者的新方法持怀疑态度，认为它只是对已有理论的重新包装，或者过于理论化。
- 工程界的迷茫：当时的工程师更熟悉频域方法和模拟电路，对这种基于矩阵运算、需要在线计算的离散时域算法感到陌生和难以实施。
- 计算机的限制：算法需要实时进行矩阵运算，而1960年代的计算机速度慢、体积大、成本高，限制了其在实际工程中的应用。

第二阶段：关键伯乐与阿波罗计划的一战成名 (1961-1969)



中國農業大學
China Agricultural University

■ 关键伯乐——斯坦利·施密特：

- NASA埃姆斯研究中心的一位工程师斯坦利·施密特最早认识到卡尔曼滤波的巨大潜力。他不仅理解了该理论，更积极地成为其“布道者”和实践推动者。

■ 杀手级应用——阿波罗登月计划：

阿波罗计划面临一个史诗级难题：如何精确地引导飞船从地球飞行38万公里后，在月球上指定位置着陆？这需要前所未有的高精度导航。：

1. **传统方法不足**：仅靠地面雷达测量（有噪声和延迟）或飞船上的惯性导航（误差会随时间累积）都无法单独满足要求。
2. **施密特的方案**：施密特提出，将卡尔曼滤波作为数据融合的核心算法，把地面雷达的测量数据与飞船惯性导航系统的预测数据结合起来，实时、最优地估计飞船的位置和速度。
3. **克服阻力**：这个提议在NASA内部遇到了巨大阻力。但施密特坚持不懈，通过仿真和演示，最终说服了项目高层。

■ 最初的冷遇：

- 卡尔曼滤波被成功应用于阿波罗飞船的导航计算机（AGC）。在整个阿波罗任务中，尤其是在著名的阿波罗11号登月和危机四伏的阿波罗13号救援中，它为飞船提供了可靠、精确的状态估计，确保了任务的成功。这被公认为卡尔曼滤波的第一个、也是最辉煌的里程碑应用。

第三阶段：军事与民用领域的快速扩散 (1970-现在)



中國農業大學
China Agricultural University

■ 阿波罗的成功为卡尔曼滤波赢得了无与伦比的声誉和可信度。其应用迅速在军事和航空航天领域开花结果：

- 战略武器：用于弹道导弹的制导和再入机动。
- 战术武器：用于战斗机、巡航导弹的火控和导航系统。
- 航天器：成为卫星轨道确定、姿态控制、交会对接的标准工具。
- 潜艇导航：整合惯性导航与外部信息。

■ 深度融入日常生活与前沿科技

- 消费电子：智能手机（导航、计步、图像防抖）、无人机（飞行稳定与导航）、运动相机。
- 自动驾驶：传感器融合的基石。高级别的自动驾驶汽车需要融合摄像头、激光雷达、毫米波雷达、GPS、IMU 等多种异构传感器的数据，以生成车辆自身及周围环境的统一、可靠的状态估计，卡尔曼滤波及其变种（如无迹卡尔曼滤波UKF）是完成这一任务的核心算法。
- 生物医学：用于心电图（ECG）、脑电图（EEG）等生理信号的噪声滤除和特征提取。
- 物联网与工业4.0：在传感器网络中实现高效的数据融合和状态监控。

- 1980年，张钟俊到佛罗里达访问了卡尔曼，翌年把卡尔曼请到了北京和上海访问。在上海，韩正之教授（张钟俊院士的博士后）记得卡尔曼在上海交通大学和上海科学会堂分别讲了控制理论的发展和系统建模问题。同样颇有意思的是，并没有数学学位的卡尔曼要求在所有会议广告和通知上注明他是一个数学家。
- 2005年卡尔曼在布拉格召开的第16届 IFAC大会上的演说结束时，他显示的ppt最后一页留下了后来为大家津津乐道的名言：
 1. **Get the physics Right.**
 2. **After that, it is all mathematics.**



目录

Contents

1. 鲁道夫·埃米尔·卡尔曼

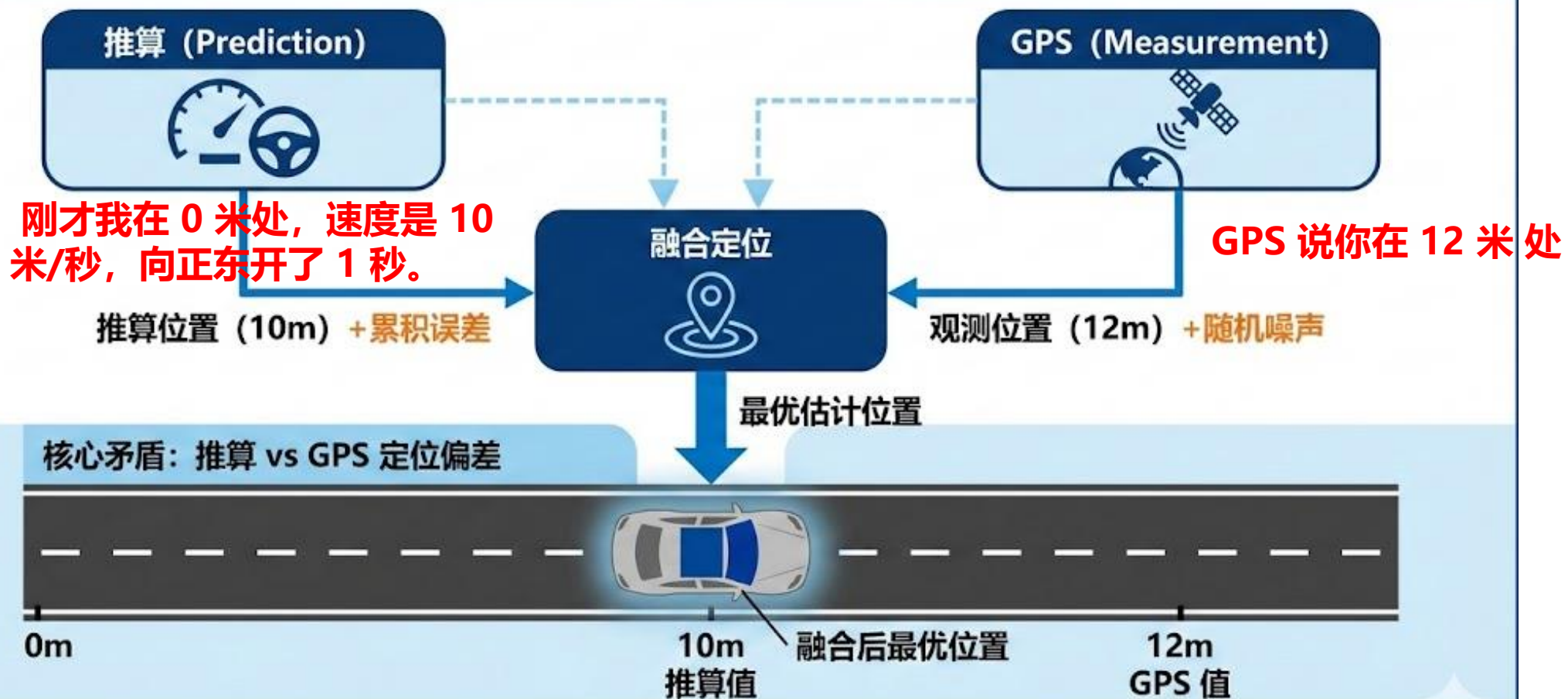
2. 卡尔曼滤波

3. 扩展卡尔曼滤波

4. 无迹卡尔曼滤波

5. 粒子滤波

车辆定位：推算与 GPS 融合的核心逻辑



解决累积误差与随机噪声的定位矛盾

■ 核心矛盾现在你面临一个困境：

- 你的推算说你在 10 米。
- 你的 GPS 说你在 12 米。

■ 你到底在哪？

1. 如果你完全相信推算，你会忽略 GPS，久而久之你会偏离路线几公里而不自知。
2. 如果你完全相信 GPS，你的车在地图上的图标会像无头苍蝇一样前后乱跳。

卡尔曼滤波就是在这两者之间充当 **“和事佬”**（或者说是一个超级理性的裁判）。它的工作流程如下：

- 第一步：综合判断（加权平均）卡尔曼滤波不会简单地取平均值 $(10+12)/2 = 11$ ，而是会看谁更靠谱。
 - 如果你的车是法拉利（推算很准），但你在隧道里（GPS 信号很差）：算法会给“推算”打高分（比如 90% 权重），给 GPS 打低分。
 - 结论：你在 10.2 米。
 - 如果你的车轮胎打滑严重（推算不准），但你在开阔地（GPS 信号极好）：算法会给 GPS 打高分（比如 90% 权重）。
 - 结论：你在 11.8 米。

- 承认“推算”和“观测”都有误差，通过计算两者的**不确定性（方差）**，找到一个**概率上最接近真实值**的结果（最优估计），并让结果是一条平滑、准确的轨迹。

其中：

- **状态方程（过程模型）：**

$$x_k = A x_{k-1} + B u_{k-1} + w_{k-1}$$

- **测量方程（观测模型）：**

$$z_k = H x_k + v_k$$

- $x_k \in \mathbb{R}^n$ 是时刻 k 的状态向量
- $u_k \in \mathbb{R}^m$ 是控制输入
- $z_k \in \mathbb{R}^p$ 是测量向量
- $A \in \mathbb{R}^{n \times n}$ 是状态转移矩阵
- $B \in \mathbb{R}^{n \times m}$ 是控制输入矩阵
- $H \in \mathbb{R}^{p \times n}$ 是观测矩阵
- $w_k \sim N(0, Q)$ 是过程噪声，协方差矩阵为 Q
- $v_k \sim N(0, R)$ 是测量噪声，协方差矩阵为 R

- 考虑一个在二维平面运动的车辆，状态向量为 $x = [p_x, p_y, v_x, v_y]^T$ ，其中 p_x, p_y 是位置， v_x, v_y 是速度。车辆配备GPS接收机，每0.1秒测量一次位置，但测量含有噪声。

- 系统建模

- 状态方程（匀速模型， $\Delta t=0.1s$ ）：
$$x_k = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + w_{k-1}$$

$$Q = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix} \leftarrow \text{过程噪声协方差反映模型不确定性}$$

- 测量方程（GPS测量位置）：
$$z_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k + v_k$$

$$R = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \leftarrow \text{测量噪声协方差 (GPS精度5米)}$$

- 我们将从最小均方误差估计的角度推导卡尔曼滤波方程。设 $\hat{x}_{k|k-1}$ 是基于直到 $k-1$ 时刻所有信息的预测估计, $\hat{x}_{k|k}$ 是基于直到 k 时刻所有信息的更新估计。

第一步: 状态预测

根据状态方程, 无噪声时的状态预测为:

$$\hat{x}_{k|k-1} = A \hat{x}_{k-1|k-1} + B u_{k-1}$$

预测误差为:

$$e_{k|k-1} = x_k - \hat{x}_{k|k-1} = A(x_{k-1} - \hat{x}_{k-1|k-1}) + w_{k-1}$$

预测误差协方差:

$$\begin{aligned} P_{k|k-1} &= E[e_{k|k-1} e_{k|k-1}^T] \\ &= A E[e_{k-1|k-1} e_{k-1|k-1}^T] A^T + E[w_{k-1} w_{k-1}^T] \\ &= A P_{k-1|k-1} A^T + Q \end{aligned}$$

这里利用了 $e_{k-1|k-1}$ 和 w_{k-1} 的不相关性。

- 我们将从最小均方误差估计的角度推导卡尔曼滤波方程。设 $\hat{x}_{k|k-1}$ 是基于直到 $k-1$ 时刻所有信息的预测估计, $\hat{x}_{k|k}$ 是基于直到 k 时刻所有信息的更新估计。

第二步: 测量预测

测量预测为:

$$\hat{z}_{k|k-1} = H \hat{x}_{k|k-1}$$

测量新息 (残差) 为:

$$\tilde{y}_k = z_k - \hat{z}_{k|k-1} = H(x_k - \hat{x}_{k|k-1}) + v_k = H e_{k|k-1} + v_k$$

新息协方差:

$$S_k = E[\tilde{y}_k \tilde{y}_k^T] = H P_{k|k-1} H^T + R$$

第三步：卡尔曼增益计算

我们希望找到一个线性更新：

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$$

其中 K_k 是卡尔曼增益。更新后的误差为：

$$e_{k|k} = x_k - \hat{x}_{k|k} = (I - K_k H) e_{k|k-1} - K_k v_k$$

更新误差协方差：

$$\begin{aligned} P_{k|k} &= E[e_{k|k} e_{k|k}^T] = (I - K_k H) P_{k|k-1} (I - K_k H)^T + K_k R K_k^T \\ &= P_{k|k-1} - K_k H P_{k|k-1} - P_{k|k-1} H^T K_k^T + K_k (H P_{k|k-1} H^T + R) K_k^T \end{aligned}$$

为了最小化估计误差的协方差矩阵的迹（即最小化均方误差），我们对 K_k 求导：

$$\frac{\partial \text{tr}(P_{k|k})}{\partial K_k} = -2(P_{k|k-1} H^T)^T + 2K_k (H P_{k|k-1} H^T + R) = 0$$

解得最优卡尔曼增益：

$$K_k = P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1}$$

第四步：协方差更新

将最优增益代入协方差公式，得到：

$$P_{k|k} = (I - K_k H) P_{k|k-1}$$

这是卡尔曼滤波的简化更新公式，数值上更稳定的形式是：

$$P_{k|k} = (I - K_k H) P_{k|k-1} (I - K_k H)^T + K_k R K_k^T$$

■ 预测阶段 (先验估计)

- $\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}$

- $P_{k|k-1} = AP_{k-1|k-1}A^T + Q$

■ 更新阶段 (后验估计)

- $\tilde{y}_k = z_k - H\hat{x}_{k|k-1}$

- $K_k = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R}$

- $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{y}_k$

- $P_{k|k} = (I - K_kH)P_{k|k-1}$

- 卡尔曼增益 K_k 是滤波器的核心，它决定了模型预测和新测量之间的权衡：
 - 当测量噪声 R 很小时 **(测量很精确)**， $K_k \approx H^{-1}$ ，滤波器更信任测量
 - 当过程噪声 Q 很小时 **(模型很精确)**， $P_{k|k-1}$ 很小， K_k 趋近于 0，滤波器更信任模型预测

■ 预测阶段 (先验估计)

- $\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}$
- $P_{k|k-1} = AP_{k-1|k-1}A^T + Q$

■ 更新阶段 (后验估计)

- $\tilde{y}_k = z_k - H\hat{x}_{k|k-1}$
- $K_k = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R}$
- $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{y}_k$
- $P_{k|k} = (I - K_kH)P_{k|k-1}$

■ 初始状态（上一时刻的估计）：

- 你仅仅凭经验猜测，水温大概是 20°C 。
- 由于是瞎猜的，你心里很不确定，估计的误差（方差）很大，设为 $Q = 5$ 。

■ 测量过程（当前时刻的观测）：

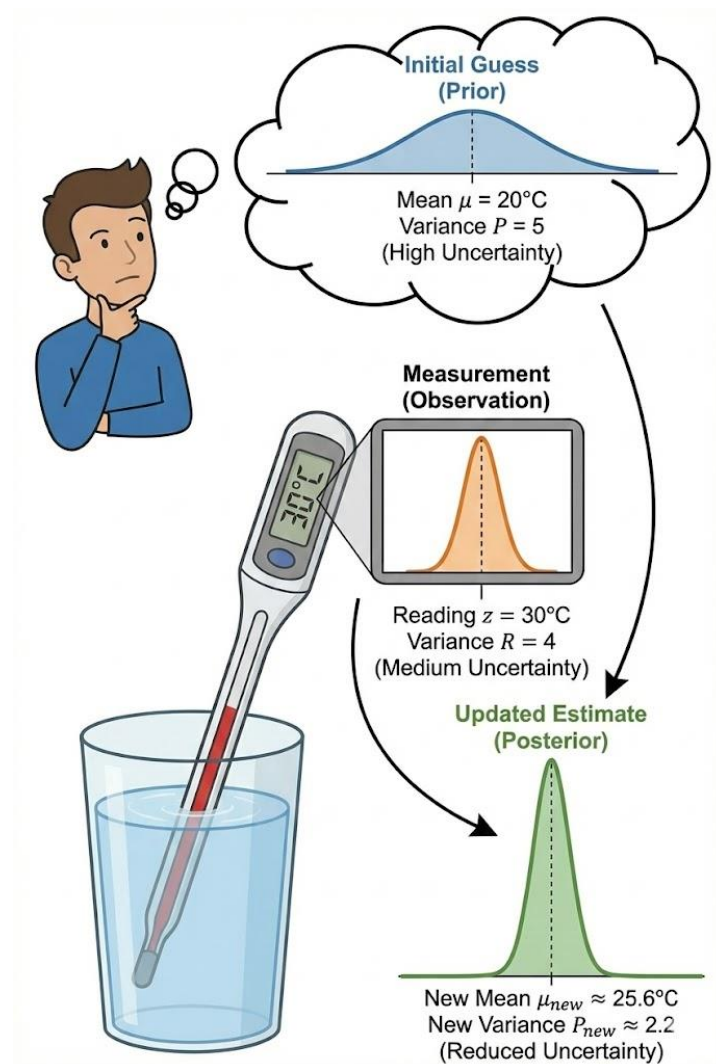
- 你拿一个温度计去测量，读数显示 30°C 。
- 这个温度计是个便宜货，不太准，说明书上说它的测量误差（方差）是 $R = 4$ 。

■ 第一步：计算卡尔曼增益

- $$K = \frac{\text{估计误差}}{\text{估计误差} + \text{测量误差}} = \frac{P}{P+R} = \frac{5}{5+4} = 0.56$$

■ 第二步：更新估计值

- 新估计值 = 旧估计值 + $K \times (\text{测量值} - \text{旧估计值}) = 20 + 0.56 \times (30 - 20) = 25.6$



■ 第一步：计算卡尔曼增益

- $$K = \frac{\text{估计误差}}{\text{估计误差} + \text{测量误差}} = \frac{P}{P+R} = \frac{5}{5+4} = 0.56$$

■ 第二步：更新估计值

- $$\text{新估计值} = \text{旧估计值} + K \times (\text{测量值} - \text{旧估计值}) = 20 + 0.56 \times (30 - 20) = 25.6$$

■ 第三步：更新误差

- 现在的我们有了结合了观测数据的结论，那么我们对这个结果的“不确定性”是不是应该降低了？
- 公式为：新误差 = $(1 - K) \times \text{旧误差} = (1 - 0.56) \times 5 = 2.2$
- 误差从最开始的 5 降到了 2.2。这说明通过引入观测数据，我们对水温的判断变得更有信心了。

■ 假如再测一次（迭代演示），卡尔曼滤波是循环迭代的。现在我们的状态变成了：

- 当前估计：25.6°C
- 当前误差：2.2
- 假设你又用那个破温度计测了一次，这次读数是 28°C（因为温度计有噪声，读数在跳动）。

- 假如再测一次（迭代演示），卡尔曼滤波是循环迭代的。现在我们的状态变成了：
 - 当前估计：25.6°C
 - 当前误差：2.2
 - 假设你又用那个破温度计测了一次，这次读数是 28°C（因为温度计有噪声，读数在跳动）。

1. 计算新的卡尔曼增益 K' ：

$$K' = \frac{2.2}{2.2+4} = 0.35$$

(注意：\$K\$ 变小了！因为我们要比刚才更信任现在的估计值了，现在的误差只有2.2，比温度计的误差4要小，所以我们开始“看不起”温度计了)

2. 计算新的温度：

$$\text{新温度} = 25.6 + 0.35 \times (28 - 25.6) = 26.44$$

$$\text{新误差} = (1 - 0.35) \times 2.2 = 1.43$$

通过这个简易的例子，你可以看到卡尔曼滤波的三个特性：

1. 融合数据：它把不准的模型（猜测）和不准的传感器（测量）结合起来，得到了一个比两者都准的结果。
2. 动态权重： K 值是动态变化的。随着测量次数增加，系统内部的误差（ P ）会越来越小，系统会越来越“自信”，对外部单次测量的噪声（ R ）就会越来越不敏感（表现为滤波效果，数据变得平滑）。
3. 不断收敛：你的估计误差从 $5 \rightarrow 2.2 \rightarrow 1.43$ ，说明只要持续观测，结果会越来越接近真实值。



目录

Contents

1. 鲁道夫·埃米尔·卡尔曼

2. 卡尔曼滤波

3. 扩展卡尔曼滤波

4. 无迹卡尔曼滤波

5. 粒子滤波

■ 非线性估计问题的挑战

1. **机器人运动学**: 涉及三角函数 $\sin(\theta)\sin(\theta)$ 、 $\cos(\theta)\cos(\theta)$
2. **航空航天器姿态动力学**: 四元数运算, 欧拉角奇异点
3. **化学反应动力学**: 非线性速率方程
4. **生物系统**: Michaelis-Menten动力学
5. **金融时间序列**: 非线性随机过程

■ 状态方程（过程模型）：

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1}$$

■ 测量方程（观测模型）：

$$z_k = h(x_k) + v_k$$

■ 线性化过程，预测阶段， $\hat{x}_{k-1|k-1}$ 处线性化 f ： $F_k = \frac{\partial f}{\partial x} \big|_{\hat{x}_{k-1|k-1}, u_{k-1}}$

■ 更新阶段， $\hat{x}_{k|k-1}$ 处线性化 h ： $H_k = \frac{\partial h}{\partial x} \big|_{\hat{x}_{k|k-1}}$

EKF算法推导

■ 预测阶段：

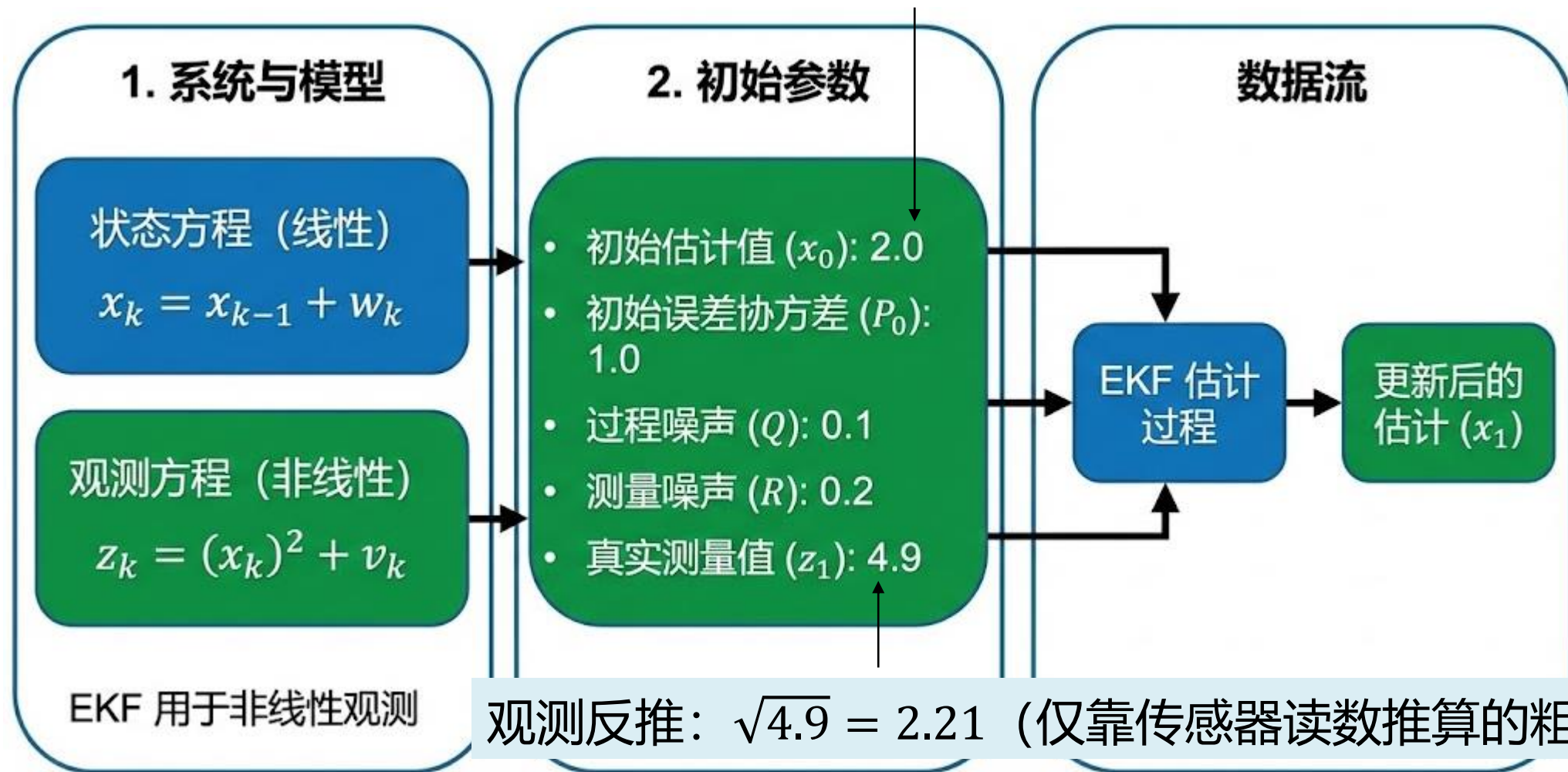
- 状态预测直接使用非线性函数： $\hat{x}_{k|k-1} = f(x_{k-1}, u_{k-1}) + w_{k-1}$
- 误差协方差预测使用线性化模型： $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$

■ 更新阶段：

- 计算预测测量： $\hat{z}_{k|k-1} = h(\hat{x}_{k|k-1})$
- 信息（残差）： $\tilde{y}_k = z_k - \hat{z}_{k|k-1}$
- 卡尔曼增益： $K_k = \frac{P_{k|k-1} H_k^T}{H_k P_{k|k-1} H_k^T + R_k}$
- 状态更新： $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$
- 协方差更新： $P_{k|k} = (I - K_k H_k) P_{k|k-1}$

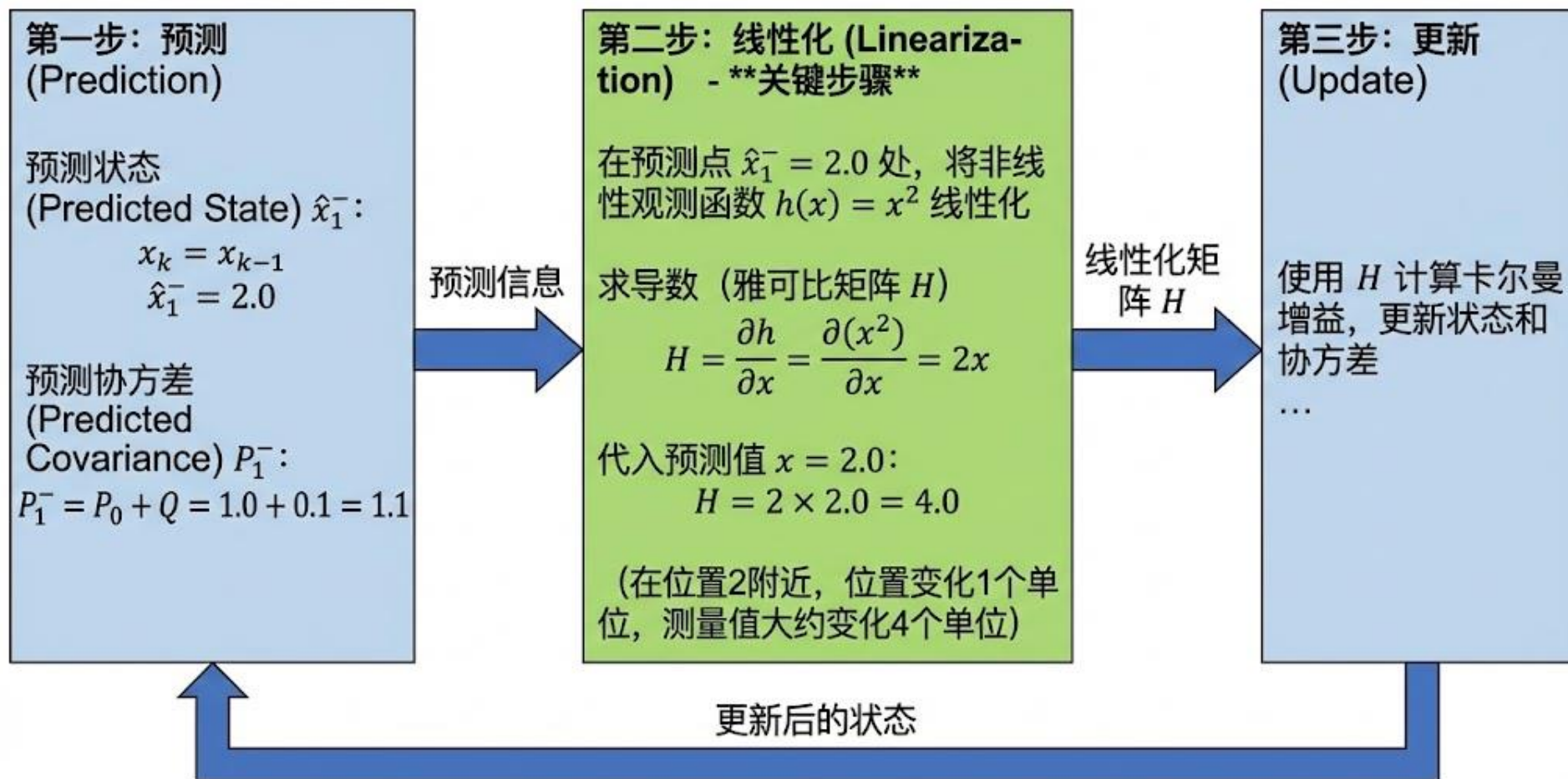
- 假设我们在估计一个物体的位置 x

初始猜测：2.0（我们以为在2.0）



■ 手把手计算步骤

EKF 预测及线性化程



扩展卡尔曼滤波 (EKF) - 更新步骤

第三步：计算卡尔曼增益 (Kalman Gain)

$$K = \frac{P_1^- \cdot H^T}{H \cdot P_1^- \cdot H^T + R}, \quad K = \frac{1.1 \times 4.0}{4.0 \times 1.1 \times 4.0 + 0.2}, \quad K = \frac{4.4}{17.8} \approx 0.247$$



卡尔曼增益 K

第四步：状态更新 (Measurement Update)

1. 计算残差 (Innovation)

$$y = z_1 - (\hat{x}_1^-)^2$$

$$y = 4.9 - (2.0)^2 = 4.9 - 4.0 = 0.9$$

2. 更新状态 (\hat{x}_1)

$$\hat{x}_1 = \hat{x}_1^- + K \cdot y$$

$$\hat{x}_1 = 2.0 + 0.247 \times 0.9$$

$$\hat{x}_1 = 2.0 + 0.222 = 2.222$$



更新后的状态 \hat{x}_1

EKF 结果: 2.222

第五步：协方差更新 (Covariance Update)

$$P_1 = (1 - K \cdot H) \cdot P_1^-, \quad P_1 = (1 - 0.247 \times 4.0) \times 1.1, \quad P_1 = (1 - 0.988) \times 1.1$$

$$P_1 = 0.012 \times 1.1 = 0.0132$$

不确定度 P : 1.0 \rightarrow 0.0132,
位置确定性提高了



目录

Contents

1. 鲁道夫·埃米尔·卡尔曼

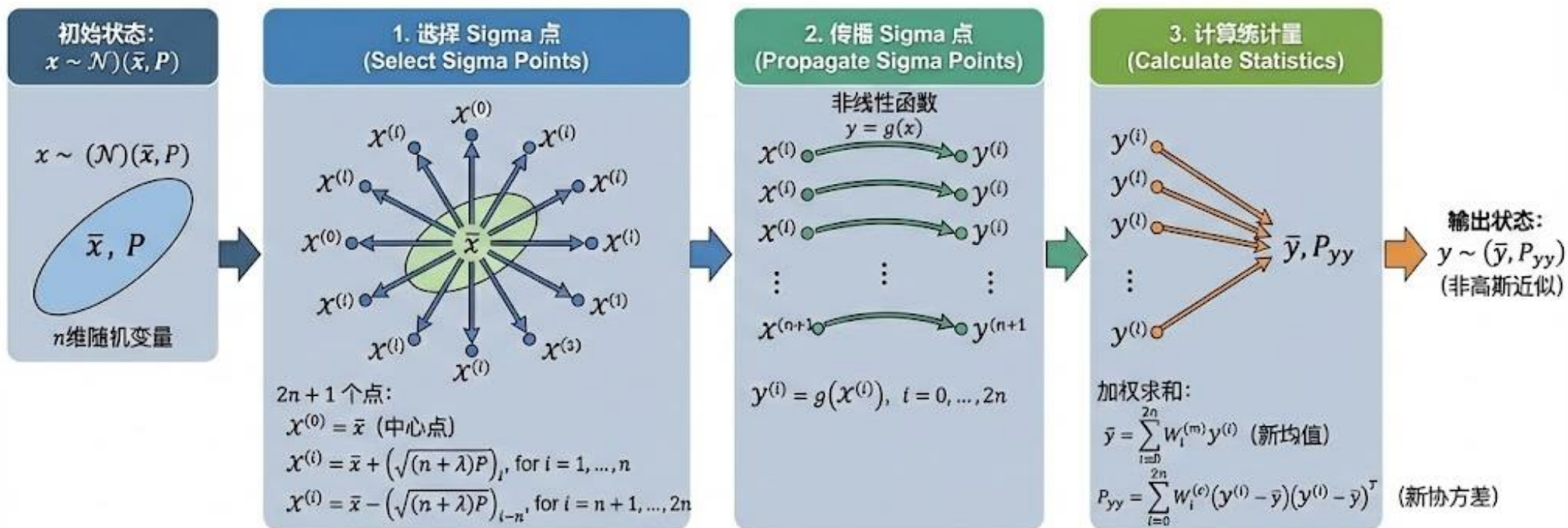
2. 卡尔曼滤波

3. 扩展卡尔曼滤波

4. 无迹卡尔曼滤波

5. 粒子滤波

- 与EKF的一阶线性化不同，UKF通过一组精心选择的样本点（Sigma点）来捕捉非线性变换的影响。



权重与参数 (Weights & Parameters)

权重计算:

$$W_0^{(m)} = \frac{\lambda}{n + \lambda}$$

$$W_0^{(c)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n+\lambda)}, \quad i = 1, \dots, 2n$$

参数:

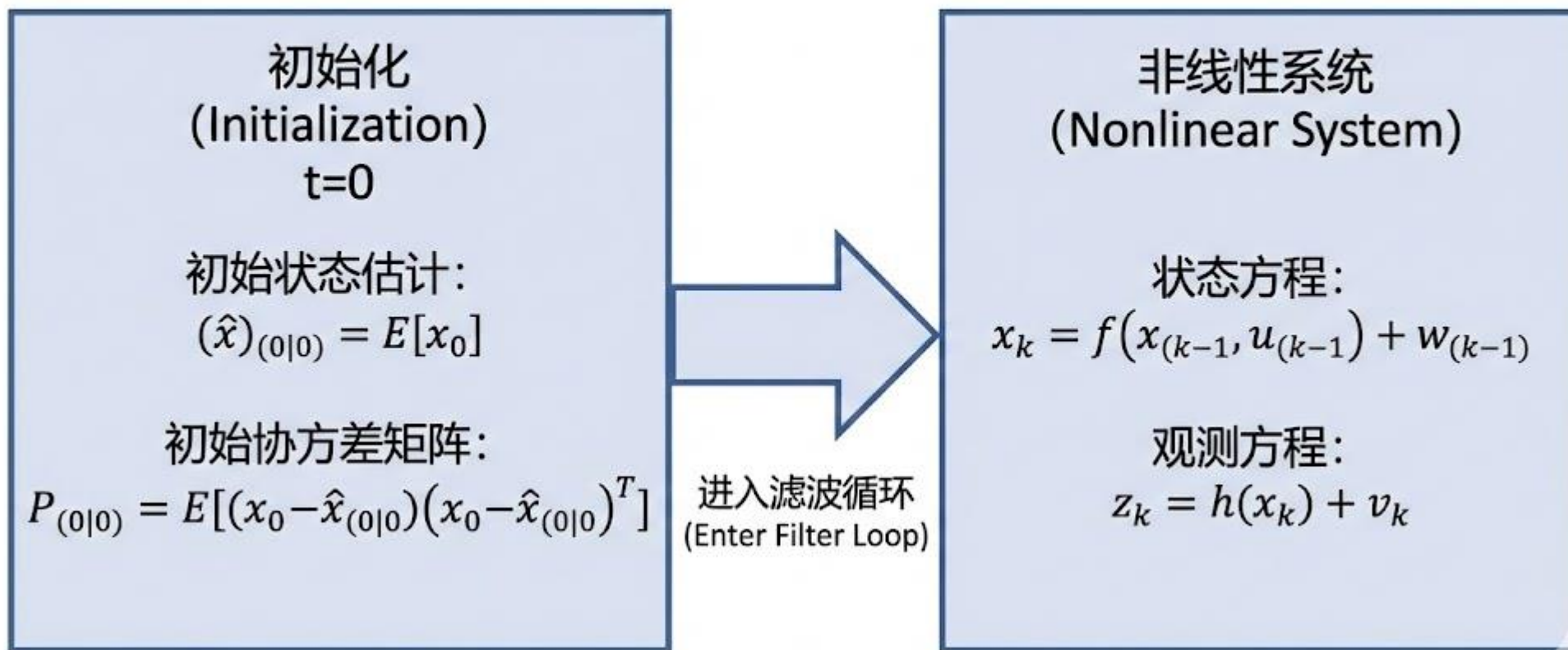
$$\lambda = \alpha^2(n + \kappa) - n$$

α : Sigma分散度 (通常 $10^{-4} \leq \alpha \leq 1$)

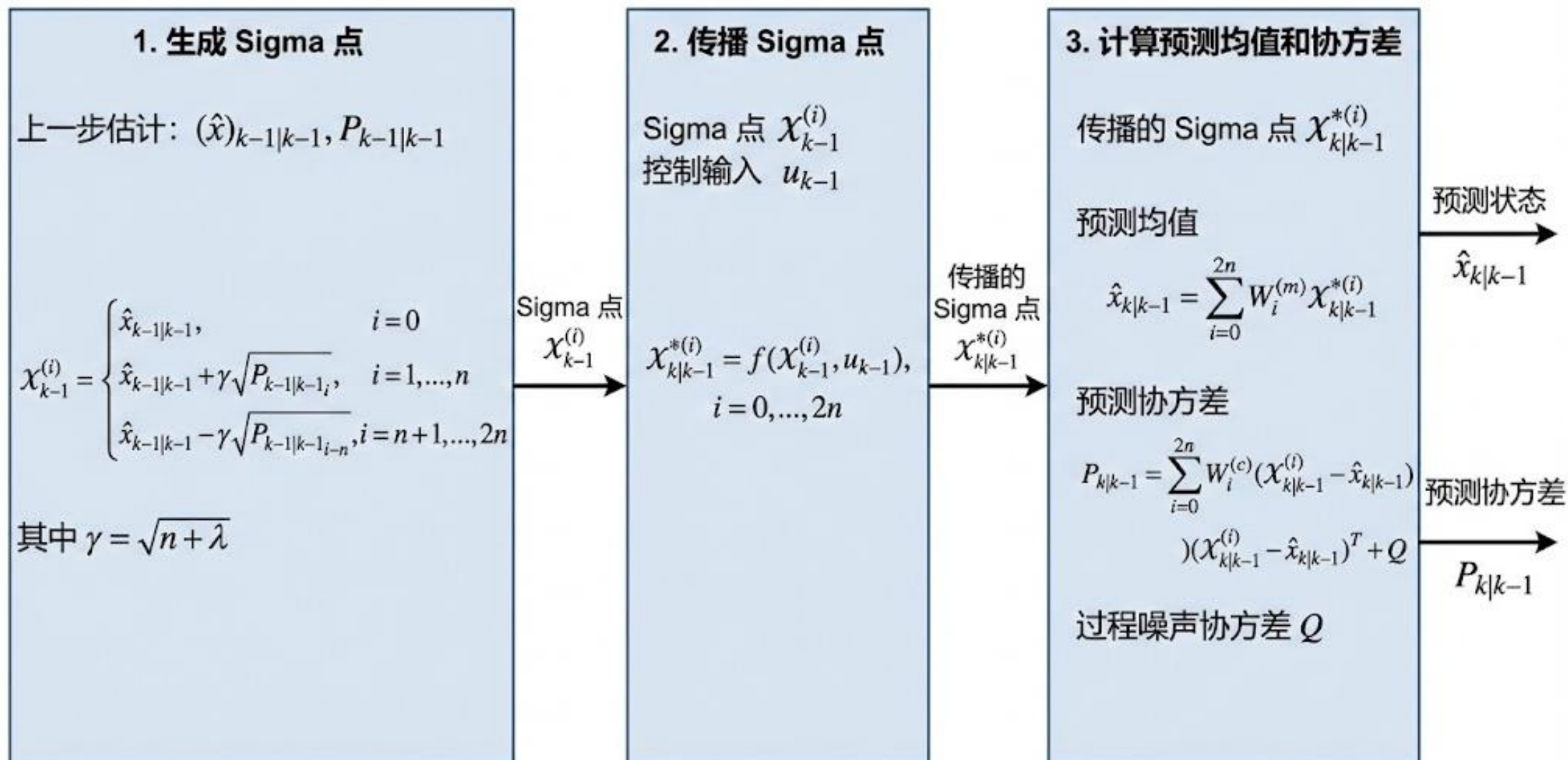
β : 分布先验 (高斯分布 $\beta = 2$)

κ : 辅助参数 (通常 $3 - n$)

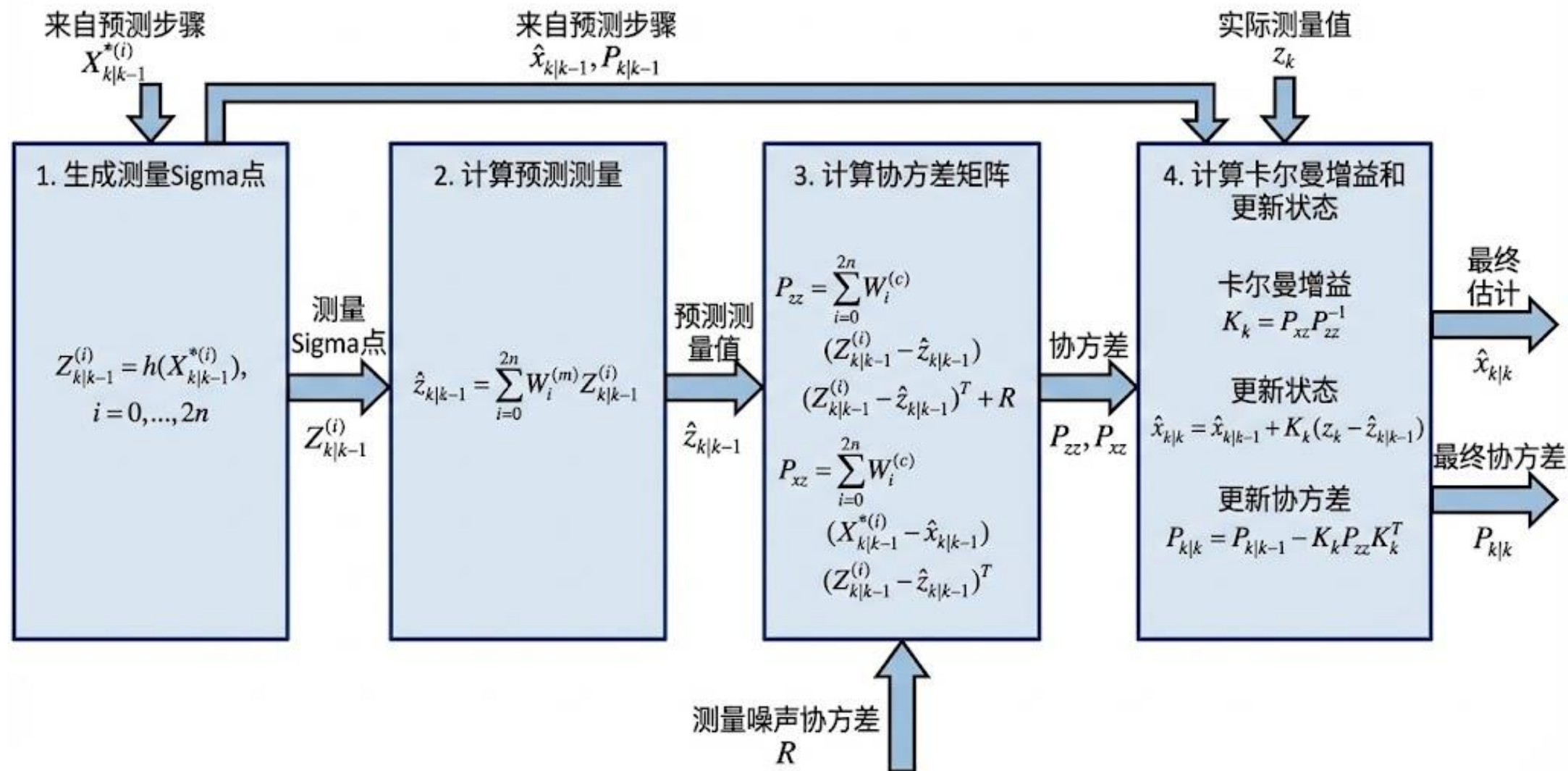
Unscented Kalman Filter: System & Initialization



UKF 预测步骤

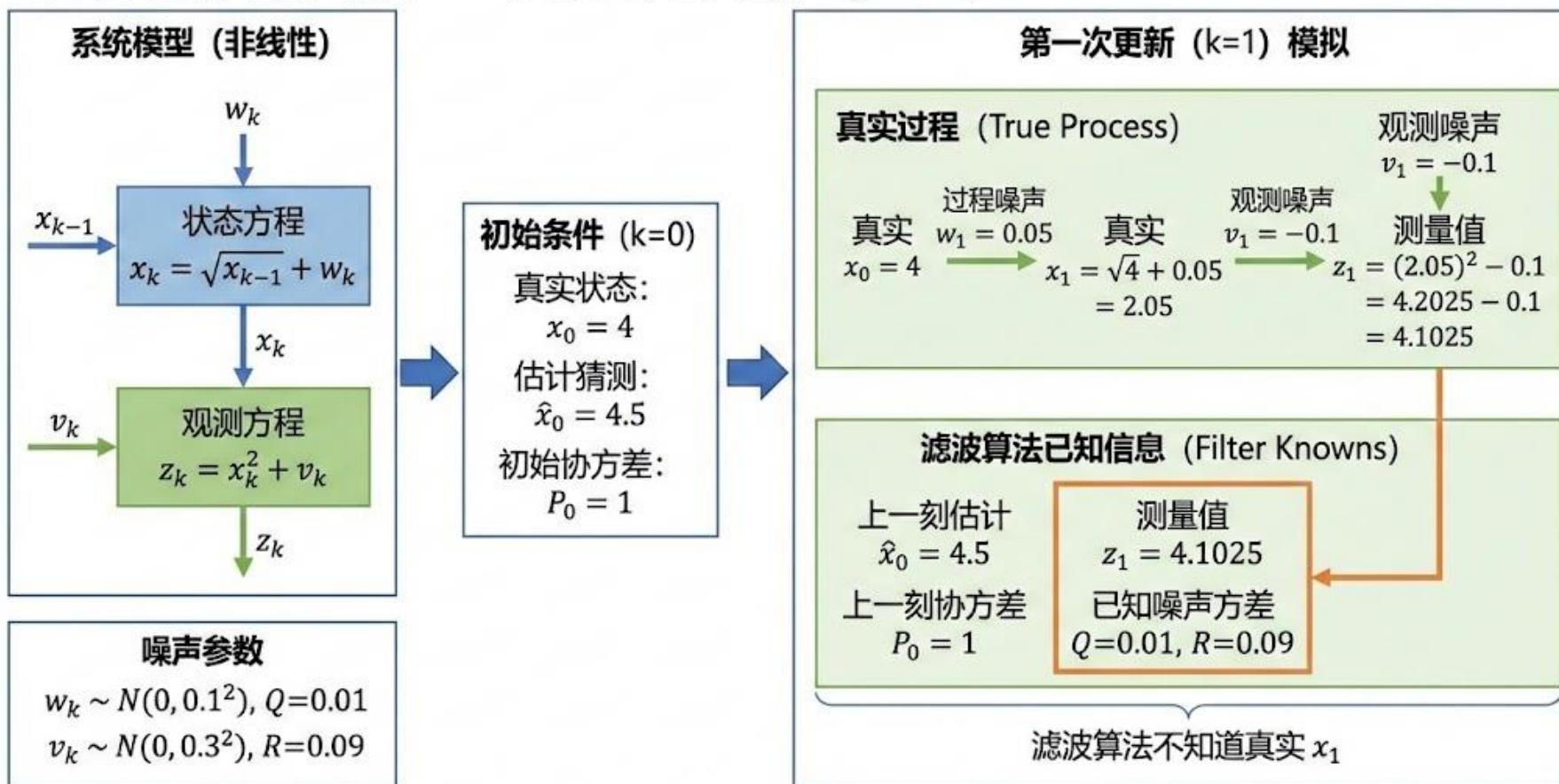


UKF 更新步骤

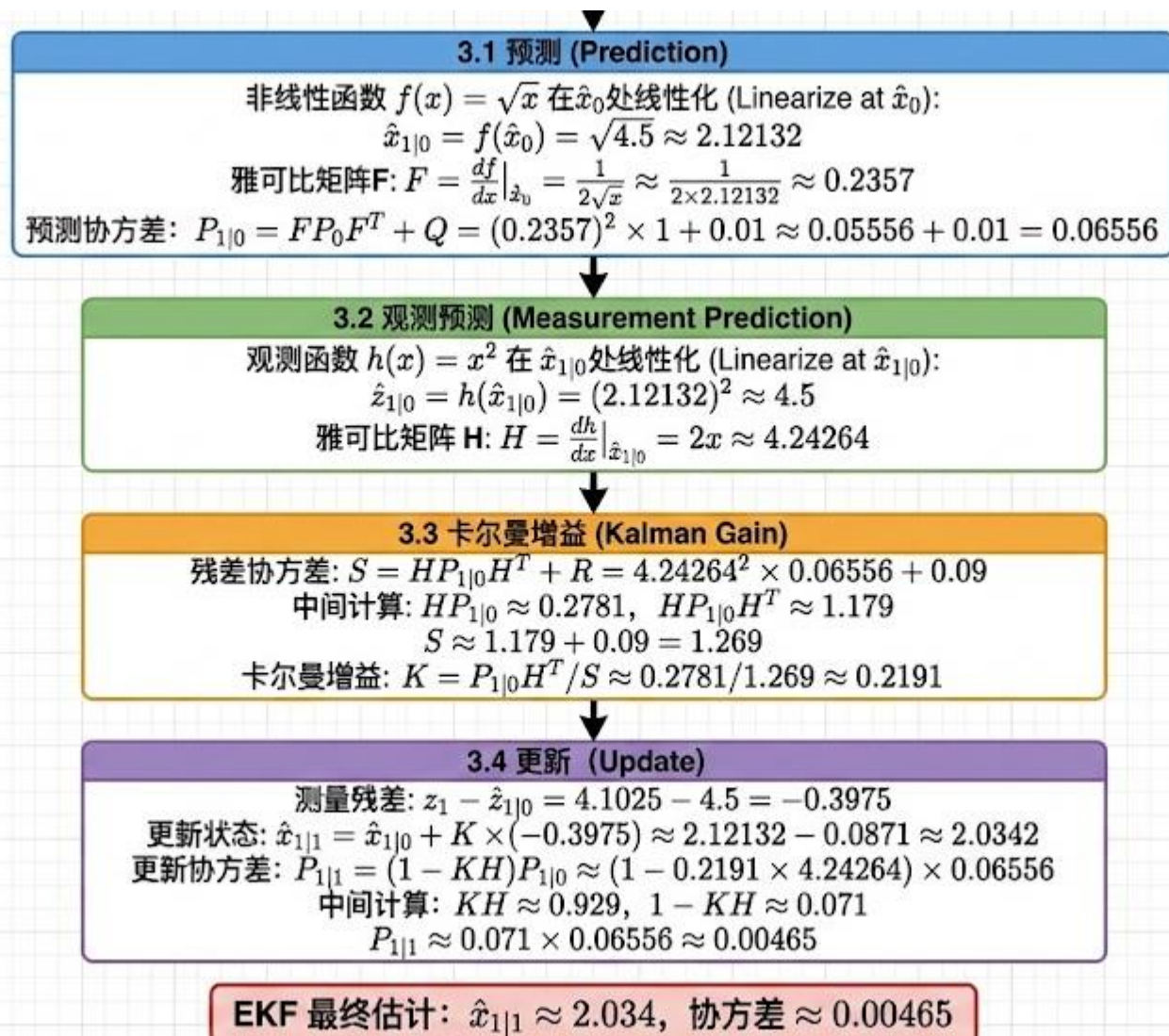


■ 某个变量 x 的变化过程如下

非线性系统与第一次更新模拟 ($k=1$)



■ EKF处理过程



■ UKF处理过程

UKF Process Visualization

UKF Parameters

$\alpha=1, \beta=2, \kappa=0$ ($\lambda = \alpha^2(n+\kappa)-n = 3-1=2, n=1$ state dimension)

4.1 Generate Sigma Points (at \hat{x}_0, P_0)

$n=1$, Select $2n+1=3$ points: $\chi_0 = \hat{x}_0 = 4.5$ $\chi_1 = \hat{x}_0 + \sqrt{(n+\lambda)P_0} = 4.5 + \sqrt{3 \times 1} \approx 4.5 + 1.732 = 6.232$

$$\chi_2 = \hat{x}_0 - \sqrt{(n+\lambda)P_0} \approx 4.5 - 1.732 = 2.768$$

Weights: $W_0^{(m)} = \frac{\lambda}{n+\lambda} = \frac{2}{3} \approx 0.6667$ $W_0^{(c)} = W_0^{(m)} + (1-\alpha^2+\beta) = 0.6667 + (1-1+2) = 2.6667$
 $W_i^{(m)} = W_i^{(c)} = \frac{1}{2(n+\lambda)} = \frac{1}{6} \approx 0.1667$ ($i=1,2$)

4.2 Prediction Step (Propagation through f)

$$\chi_0^* = \sqrt{4.5} \approx 2.12132, \quad \chi_1^* = \sqrt{6.232} \approx 2.496, \quad \chi_2^* = \sqrt{2.768} \approx 1.664$$

$$\text{Predicted Mean: } \hat{x}_1|_0 = \sum W_i^{(m)} \chi_i^* = 0.6667 \times 2.12132 + 0.1667 \times 2.496 + 0.1667 \times 1.664$$

$$\text{Calculation: } 1.4142 + 0.4160 + 0.2773 \approx 2.1075$$

$$\text{Predicted Covariance: } P_1|_0 = \sum W_i^{(c)} (\chi_i^* - \hat{x}_1|_0)^2 + Q$$

$$\text{Differences: } d_0 = 0.01382, d_1 = 0.3885, d_2 = -0.4435$$

$$\text{Squared Differences: } d_0^2 = 0.000191, d_1^2 = 0.1510, d_2^2 = 0.1967$$

$$\text{Weighted Sum: } 0.6667 \times 0.000191 \approx 0.000127; 0.1667 \times 0.1510 \approx 0.02518; 0.1667 \times 0.1967 \approx 0.03278$$

$$\text{Sum} = 0.058085$$

$$\text{Add } Q=0.01 \rightarrow P_1|_0 \approx 0.068085$$

UKF Process Visualization (Continued)

4.3 Observation Prediction (Propagation through h)

For each Sigma point through $h(x)=x^2$: $Z_0 = (2.12132)^2 = 4.5$, $Z_1 = (2.496)^2 \approx 6.230$, $Z_2 = (1.664)^2 \approx 2.769$

Predicted observation mean: $\hat{z}_{1|0} = 0.6667 \times 4.5 + 0.1667 \times 6.230 + 0.1667 \times 2.769 = 3.0 + 1.038 + 0.461 \approx 4.499$

4.4 Covariance and Kalman Gain

$S = \sum W_i^{(c)} (Z_i - \hat{z}_{1|0})^2 + R$ Residuals: $e_0 = 4.5 - 4.499 = 0.001$; $e_1 = 6.230 - 4.499 = 1.731$; $e_2 = 2.769 - 4.499 = -1.730$

Squared: $e_0^2 \approx 0.000001$, $e_1^2 \approx 2.997$, $e_2^2 \approx 2.993$

Weighted Sum: $0.6667 \times 0.000001 \approx 0$; $0.1667 \times 2.997 \approx 0.4995$; $0.1667 \times 2.993 \approx 0.4988$ Sum ≈ 0.9983

Add $R=0.09 \rightarrow S \approx 1.0883$

$P_{xz} = \sum W_i^{(c)} (\chi_i^* - \hat{x}_{1|0})(Z_i - \hat{z}_{1|0})$ Using previous d and e:

$i=0$: $0.01382 \times 0.001 \approx 0.00001382$ $i=1$: $0.3885 \times 1.731 \approx 0.672$ $i=2$: $-0.4435 \times (-1.730) \approx 0.767$

Weighted Sum: $0.6667 \times 0.00001382 \approx 0.0000092$; $0.1667 \times 0.672 \approx 0.112$; $0.1667 \times 0.767 \approx 0.128$ Sum ≈ 0.240

Kalman Gain: $K = \frac{P_{xz}}{S} \approx \frac{0.240}{1.0883} \approx 0.2205$

4.5 Update

$\hat{x}_{1|1} = \hat{x}_{1|0} + K(z_1 - \hat{z}_{1|0}) = 2.1075 + 0.2205 \times (4.1025 - 4.499) = 2.1075 + 0.2205 \times (-0.3965) = 2.1075 - 0.0874 \approx 2.0201$

$P_{1|1} = P_{1|0} - KSK^T = 0.068085 - 0.2205^2 \times 1.0883$

$K^2S \approx 0.04862 \times 1.0883 \approx 0.05291$

$P_{1|1} \approx 0.068085 - 0.05291 \approx 0.015175$

UKF Final Estimate: $\hat{x}_{1|1} \approx 2.020$, Covariance ≈ 0.0152



目录

Contents

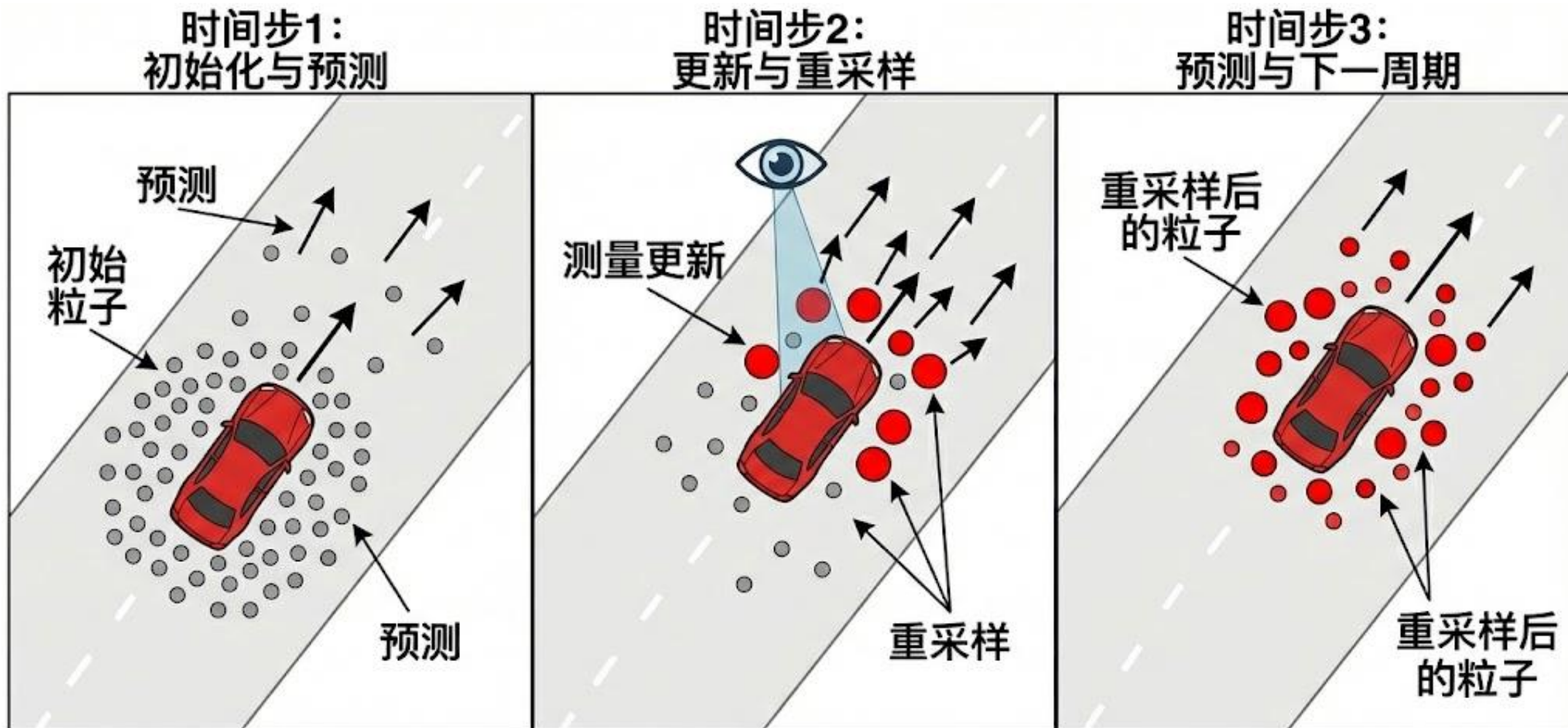
1. 鲁道夫·埃米尔·卡尔曼

2. 卡尔曼滤波

3. 扩展卡尔曼滤波

4. 无迹卡尔曼滤波

5. 粒子滤波



● 灰点: 粒子 (低权重)
● 红点: 粒子 (高权重)

🚗 红车: 真实状态
👁️ 眼睛图标: 传感器测量

- 粒子滤波器（PF）采用序贯蒙特卡洛方法，使用一组随机样本（粒子）来近似状态的后验概率分布。与EKF和UKF不同，PF不对系统模型做线性化假设，也不要求噪声是高斯的，因此适用于更广泛的非线性非高斯系统。
- 在贝叶斯框架下，状态估计问题可以表述为计算后验概率密度：

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}$$

- 其中预测分布：

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|z_{1:k-1}) dx_{k-1}$$

对于非线性非高斯系统，这些积分无法解析计算。粒子滤波器的核心思想是用一组带权重的粒子近似后验分布： $p(x_k|z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i)$ 其中 $\delta(\cdot)$ 是Dirac delta函数。

■ 序贯重要性采样 (SIS)

最基本的粒子滤波算法是序贯重要性采样：

初始化：从先验分布 $p(x_0)$ 中抽取 N 个粒子 $(x_0^i)_{i=1}^N$ ，权重 $w_0^i = \frac{1}{N}$

对于每个时间步 $k = 1, 2, \dots$ ：

1. 重要性采样：从建议分布 $q(x_k | x_{k-1}^i, z_k)$ 中抽取新粒子 $x_k^i \sim q(x_k | x_{k-1}^i, z_k)$
2. 权重更新：计算重要性权重 $w_k^i = w_{k-1}^i \times \frac{[p(z_k | x_k^i) p(x_k^i | x_{k-1}^i)]}{q(x_k^i | x_{k-1}^i, z_k)}$
3. 权重归一化： $\tilde{w}_k^i = \frac{w_k^i}{\sum_{j=1}^N w_k^j}$
4. 状态估计： $\hat{x}_k = \sum_{i=1}^N \tilde{w}_k^i x_k^i$

■ 序贯重要性重采样 (SIR) 滤波器:

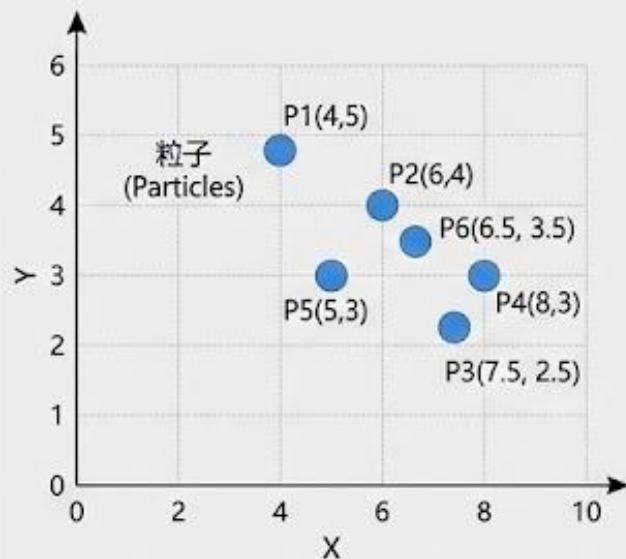
算法步骤:

1. 初始化: 抽取 N 个粒子 $x_0^i \sim p(x_0)$, $w_0^i = \frac{1}{N}$
2. 对于 $k = 1, 2, \dots$
 - a. 预测: $x_k^i \sim p(x_k | x_{k-1}^i)$
 - b. 权重更新: $w_k^i = p(z_k | x_k^i)$
 - c. 权重归一化: $\tilde{w}_k^i = \frac{w_k^i}{\sum_j w_k^j}$
 - d. 重采样: 根据 \tilde{w}_k^i 重采样 N 个新粒子
 - e. 重置权重: $w_k^i = \frac{1}{N}$

- 一个机器人在 $10m \times 10m$ 的房间里找一个小型信标（比如一个WiFi热点）。
 - - 真实信标位置：($x = 7.0, y = 3.0$)（单位：米）
 - - 机器人不知道信标位置，但知道大概在房间中央偏右区域。
 - - 机器人配备一个信号强度检测器，能测量到信标的距离（有误差）。
 - - 机器人可以走动并连续测量，不断更新对信标位置的估计。
- 状态：信标坐标 (x, y) （固定，但我们假装不知道）
- 观测：带噪声的距离测量值 d
- 目标：用粒子滤波估计信标的二维位置分布

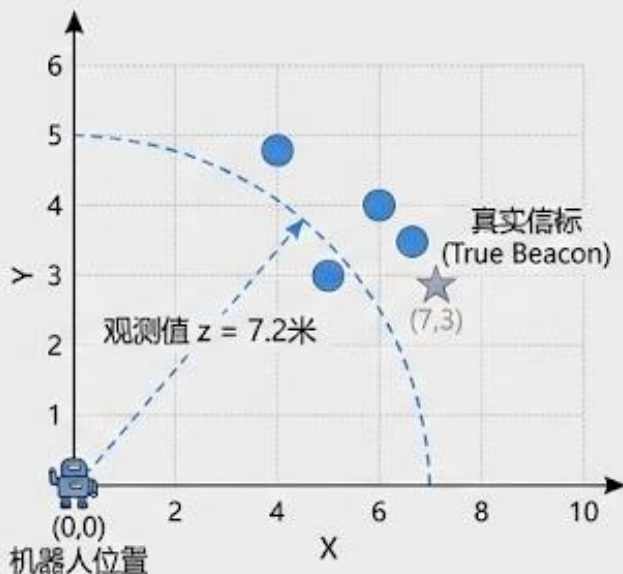
粒子滤波定位过程演示（二维距离观测）

第1步：初始化 (Initialization)



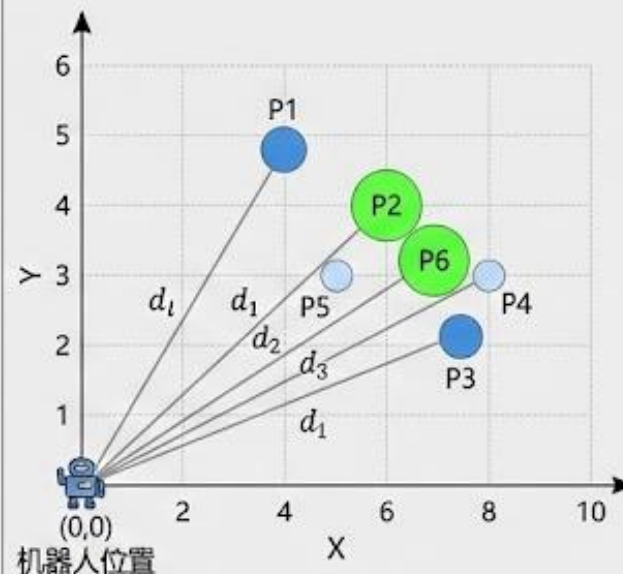
初始状态：6个粒子随机分布，权重相等。
 $w_i = \frac{1}{6} \approx 0.167$

第2步：第一次观测 (Observation)



真实距离 ≈ 7.616 米。
观测值 $z = 7.2$ 米（带噪声）。
噪声标准差 $\sigma = 1.0$ 米

第3步：更新权重 (Weight Update)



粒子 ID	位置 (x,y)	理论距离 d_i	似然 w_i^*	归一化权重 w_i
P1	(4,5)	6.403	0.728	0.170
P2	(6,4)	7.211	0.9999	0.233 (最高)
P3	(7.5,2.5)	7.906	0.780	0.182
P4	(8,3)	8.544	0.405	0.094
P5	(5,3)	5.831	0.392	0.091
P6	(6.5,3.5)	7.382	0.984	0.230 (次高)

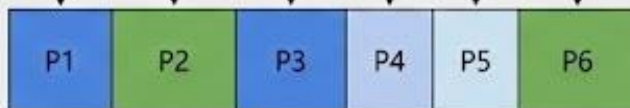
注：P2和P6最接近观测距离 $z = 7.2$ ，因此获得最大权重。

粒子滤波定位过程演示 (重采样与迭代)

第4步: 重采样 (Resampling)

累积权重区间 (Cumulative Weight Intervals)

0.45 0.82 0.25 0.60 0.95 0.10



重采样后新粒子集 (New Resampled Particles)

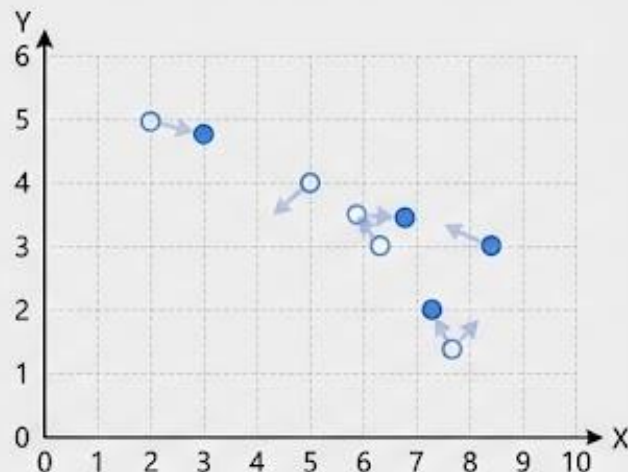
新ID	原ID	位置 (x,y)
1	P3	(7.5, 2.5)
2	P6	(6.5, 3.5) (复制)
3	P2	(6.0, 4.0)
4	P4	(8.0, 3.0)
5	P6	(6.5, 3.5) (复制)
6	P1	(4.0, 5.0)

当前估计 (Current Estimate) :

均值 $x \approx 6.417$, $y \approx 3.583$.

离真实 (7.0, 3.0) 有误差。

第5步: 预测 (Prediction - 加过程噪声)



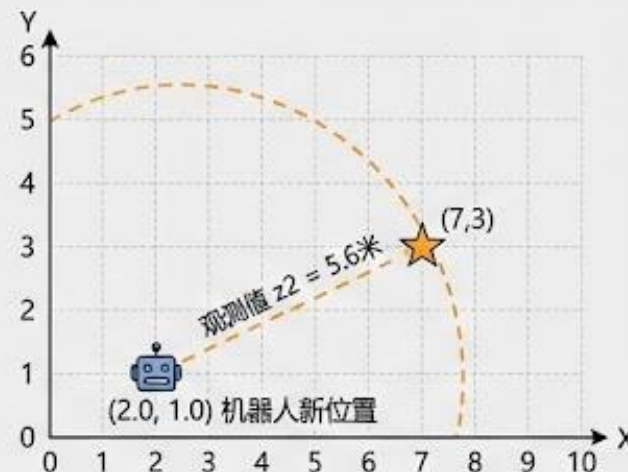
过程噪声 (Process Noise)

标准差 $\sigma = 0.1$ 米。

粒子扩散以继续收敛。

粒子 ID	新位置 (x,y) (带噪声)
1	(7.53, 2.48)
2	(6.62, 3.45)
3	(5.92, 4.01)
4	(7.97, 3.07)
5	(6.55, 3.47)
6	(4.10, 4.91)

第6步: 第二次观测 (Second Observation)



真实距离 ≈ 5.385 米。

观测值 $z2 = 5.6$ 米 (带噪声)

新权重计算 (New Weight Calculation)

粒子 ID	新位置 (x,y)	理论距离 d	似然 w^*
1	(7.53, 2.48)	5.724	0.992
2	(6.62, 3.45)	5.229	0.933
3	(5.92, 4.01)	4.943	0.806
4	(7.97, 3.07)	6.318	0.773
5	(6.55, 3.47)	5.177	0.914
6	(4.10, 4.91)	4.438	0.509

注: 高权重粒子 (靠近真实信标) 将在下次重采样中更多被复制, 粒子群逐渐向 (7.0, 3.0) 聚集。





中國農業大學
China Agricultural University

谢谢

