

卡尔曼滤波

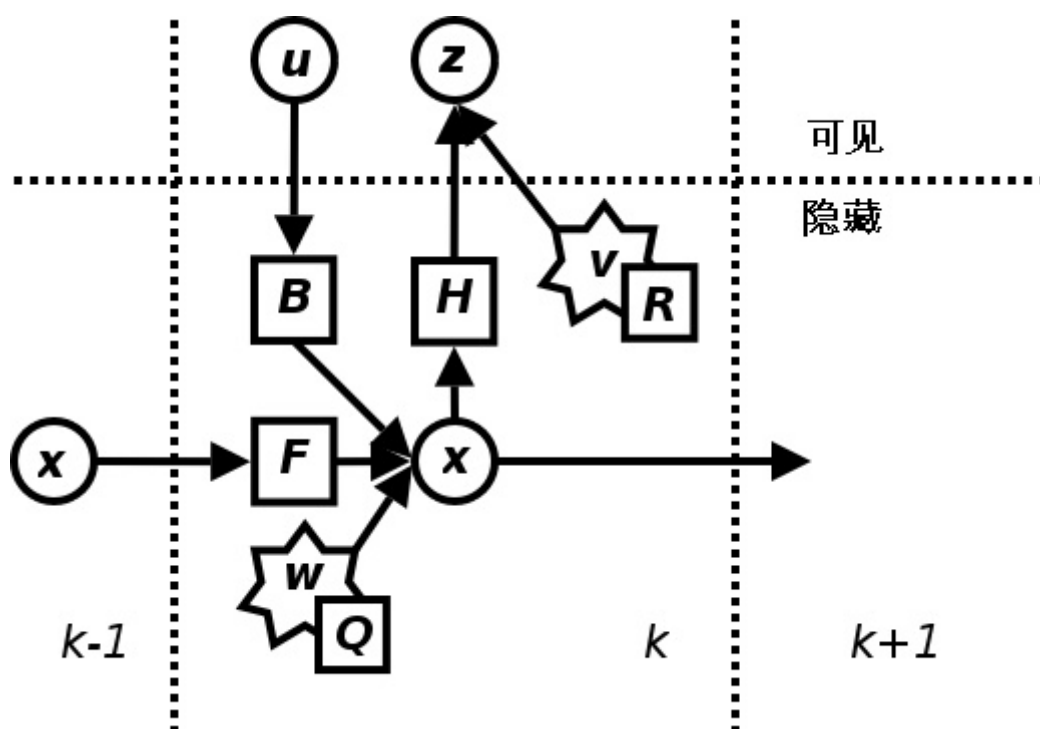
卡尔曼滤波以其发明者鲁道夫.E.卡尔曼 (Rudolph E. Kalman) 命名，但根据文献可知卡尔曼并非第一个提出该算法的人，Peter Swerling在更早之前就提出了类似算法¹。

卡尔曼滤波的典型实例是从一组有限、包含噪声的观测序列中预测出物体的位置和速度。最为常见的卡尔曼滤波器是锁相环。

线性卡尔曼滤波

待估计模型

卡尔曼滤波建立在隐马尔可夫模型上，其基本动态系统使用一个马尔可夫链表示。



使用公式表示即，对于需要求解的状态 $x \in R^n$ 满足

$$x_k = F_k x_{k-1} + B_k u_{k-1} + w_{k-1}$$

其中 F 为 $n \times n$ 状态的转移矩阵，在现实的过程中他可能是时变的，但是在这里假设其为常量。 $n \times l$ 矩阵 B 是可能存在的控制器 $u \in R^l$ 的输入-控制模型。对于这一隐马尔可夫链有观测值 $z \in R^m$ 满足

$$z_k = H_k x_k + v_k$$

其中 $m \times n$ 矩阵 H 为观测模型，在真实情况中观测模型可能会是时变的，但在此处假设其为常量。

随机变量 w_k, v_k 分别为过程噪声和观测噪声，假设他们是独立的高斯白噪声且

$$p(w_k) \sim N(0, Q_k)$$

$$p(v_k) \sim N(0, R_k)$$

推导

卡尔曼滤波器是建立在贝叶斯滤波器的基础上的²，首先介绍贝叶斯滤波器。

贝叶斯准则与全概率公式

贝叶斯准则：

$$p(x|y, z) = \frac{p(y|x, z)p(x|z)}{p(y|z)}$$

全概率公式：

$$p(x) = \int p(x|y)p(y)dy$$

贝叶斯滤波器

贝叶斯滤波器的思想即使用已有的观测值和控制输入来估计当前时刻的状态，对于状态 x_k ，观测值 z_k ，控制量 u_k ，写成概率的形式即：

$$bel(x_k) = p(x_k|z_{1:k}, u_{1:k}) = p(x_k|z_k, z_{1:k-1}, u_{1:k})$$

根据贝叶斯准则

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

分母与 x_k 无关，使用常数 $1/\eta$ 代替，由于隐马尔可夫链的性质，某时刻的观测值只与其状态有关，则 $p(z_k|x_k, z_{1:k-1}, u_{1:k}) = p(z_k|x_k)$ 上式可以化简为：

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

可以理解为使用先验和似然估计后验。则先验为：

$$\overline{bel}(x_k) = p(x_k|z_{1:k-1}, u_{1:k})$$

代入全概率公式，由于积分中乘积的前一项为根据前一时刻状态预测下一时刻状态，根据马尔可夫链的性质，这一概率与观测值 z 无关。同时，马尔可夫性质表明只有当前的控制信号和上一时刻的状态对当前时刻的状态有影响，则积分乘积前一项中的 $u_{1:k}$ 可以被化为 u_k ，乘积后一项中的 $u_{1:k}$ 可以被化为 $u_{1:k-1}$ 。同时观察可知乘积的第二项即 $bel(x_{k-1})$ ：

$$\begin{aligned}\overline{bel}(x_k) &= \int p(x_k|x_{k-1}, z_{1:k-1}, u_{1:k})p(x_{k-1}|z_{1:k-1}, u_{1:k})dx_{k-1} \\ &= \int p(x_k|x_{k-1}, u_k)bel(x_{k-1})dx_{k-1}\end{aligned}$$

后验为：

$$bel(x_k) = \eta p(z_k|x_k)\overline{bel}(x_k)$$

代入待估计模型

求解先验分布

由于假设噪声符合高斯分布，则 $bel(x_k)$ 也应符合高斯分布，设其均值为 μ_k ，方差 Σ_k 。对于先验公式，有

$$\begin{aligned}p(x_k|x_{k-1}, u_k) &\sim N(F_k x_{k-1} + B_k u_k, Q_k) \\ bel(x_{k-1}) &\sim N(\mu_{k-1}, \Sigma_{k-1})\end{aligned}$$

则 $\overline{bel}(x_k)$ 可以表示成 $\overline{bel}(x_k) = \gamma \int \exp(-L_k)dx_{k-1}$ 的形式，其中 γ 为常数，有

$$L_k = \frac{1}{2}[(x_k - F_k x_{k-1} - B_k u_k)^T Q_k^{-1} (x_k - F_k x_{k-1} - B_k u_k) + (x_{k-1} - \mu_{k-1})^T \Sigma_{k-1}^{-1} (x_{k-1} - \mu_{k-1})]$$

两个高斯分布的乘积 $\gamma \exp(-L_k)$ 也为高斯分布，而 L_k 是关于 x_{k-1} 的二次型，则求解

$\partial L_k / \partial x_{k-1} = 0$ 可以得到该分布的均值 μ'_k ， $\partial^2 L_k / \partial x_{k-1}^2$ 为其方差的倒数 Ψ_k^{-1} 。可以解得：

$$\mu'_k = \frac{F_k^T Q_k^{-1} x_k - F_k^T Q_k^{-1} B_k u_k + \Sigma_{k-1}^{-1} \mu_{k-1}}{\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k}$$

$$\Psi_k^{-1} = F_k^T Q_k^{-1} F_k + \Sigma_{k-1}^{-1}$$

为了求解积分，需要使用一些技巧来简化运算，定义二次型函数

$$L'_k(x_{k-1}, x_k) = \frac{1}{2}(x_{k-1} - \mu'_k)^T \Psi_k^{-1} (x_{k-1} - \mu'_k)$$

可以发现 $L_k - L'_k$ 为 x_k 的二次型，且有 $\det(2\pi\Psi_k)^{1/2} \int \exp(-L'_k) dx_{k-1} = 1$ ，则可以将 L_k 拆分来去积分。定义

$$L''_k(x_k) = L_k - L'_k$$

则

$$\overline{bel}(x_k) = \gamma \int \exp[-L''_k(x_k) - L'_k(x_{k-1}, x_k)] dx_{k-1}$$

$$= \gamma \exp[-L''_k(x_k)]$$

由于 L''_k 为关于 x_k 的二次型，且 $\overline{bel}(x_k)$ 符合高斯分布，则使用求一阶导数和二阶导数的方法来求取其方差 $\bar{\Sigma}_k$ 和均值 $\bar{\mu}_k$ 。

一阶导数：

$$\begin{aligned} \frac{\partial L''_k(x_k)}{\partial x_k} &= Q_k^{-1} (x_k - F_k x_{k-1} - B_k u_k) + \frac{Q_k^{-1} F_k}{\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k} (F_k^T Q_k^{-1} F_k + \Sigma_{k-1}^{-1}) (x_{k-1} - \mu'_k) \\ &= Q_k^{-1} (x_k - F_k x_{k-1} - B_k u_k) + Q_k^{-1} F_k (x_{k-1} - \mu'_k) \\ &= Q_k^{-1} (x_k - B_k u_k) - Q_k^{-1} F_k \mu'_k \\ &= Q_k^{-1} (x_k - B_k u_k) - Q_k^{-1} F_k \frac{F_k^T Q_k^{-1} (x_k - B_k u_k) + \Sigma_{k-1}^{-1} \mu_{k-1}}{\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k} \\ &= (Q_k^{-1} - Q_k^{-1} F_k \frac{F_k^T Q_k^{-1}}{\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k}) (x_k - B_k u_k) - Q_k^{-1} F_k \frac{\Sigma_{k-1}^{-1} \mu_{k-1}}{\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k} \quad (\text{求逆公式}) \\ &= (Q_k + F_k \Sigma_{k-1} F_k^T)^{-1} (x_k - B_k u_k) - Q_k^{-1} F_k \frac{\Sigma_{k-1}^{-1} \mu_{k-1}}{\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k} \end{aligned}$$

令一阶导数等于零

$$\begin{aligned}
\frac{\partial L_k''(x_k)}{\partial x_k} &= 0 \\
(Q_k + F_k \Sigma_{k-1} F_k^T)^{-1} (x_k - B_k u_k) &= Q_k^{-1} F_k \frac{\Sigma_{k-1}^{-1} \mu_{k-1}}{\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k} \\
x_k - B_k u_k &= (Q_k + F_k \Sigma_{k-1} F_k^T) Q_k^{-1} F_k \frac{\Sigma_{k-1}^{-1} \mu_{k-1}}{\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k} \\
x_k - B_k u_k &= (Q_k + F_k \Sigma_{k-1} F_k^T) Q_k^{-1} F_k (\Sigma_{k-1}^{-1} + F_k^T Q_k^{-1} F_k)^{-1} \Sigma_{k-1}^{-1} \mu_{k-1} \\
x_k - B_k u_k &= (Q_k + F_k \Sigma_{k-1} F_k^T) Q_k^{-1} F_k (I + \Sigma_{k-1} F_k^T Q_k^{-1} F_k)^{-1} \mu_{k-1} \\
x_k - B_k u_k &= (F_k + F_k \Sigma_{k-1} F_k^T Q_k^{-1} F_k) (I + \Sigma_{k-1} F_k^T Q_k^{-1} F_k)^{-1} \mu_{k-1} \\
x_k - B_k u_k &= F_k (I + \Sigma_{k-1} F_k^T Q_k^{-1} F_k) (I + \Sigma_{k-1} F_k^T Q_k^{-1} F_k)^{-1} \mu_{k-1} \\
x_k - B_k u_k &= F_k \mu_{k-1} \\
x_k &= F_k u_{k-1} + B_k u_k
\end{aligned}$$

则

$$\bar{\mu}_k = F_k u_{k-1} + B_k u_k \quad (1)$$

二阶导数:

$$\bar{\Sigma}_k^{-1} = \frac{\partial^2 L_k''(x_k)}{\partial x_k^2} = (Q_k + F_k \Sigma_{k-1} F_k^T)^{-1} \quad (2)$$

则得到先验分布 $\bar{bel}(x_k) \sim N(\bar{\mu}_k, \bar{\Sigma}_k)$

求解后验分布

由系统的观测方程可以得到 $p(z_k | x_k) \sim N(H_k x_k, R_k)$, 已经求得先验分布 $\bar{bel}(x_k) \sim N(\bar{\mu}_k, \bar{\Sigma}_k)$, 则后验分布

$$bel(x_k) = \eta p(z_k | x_k) \bar{bel}(x_k)$$

可以简化为 $bel(x_k) = \eta \exp(-J_k)$, 其中

$$J_k = \frac{1}{2} [(z_k - H_k x_k)^T R_k^{-1} (z_k - H_k x_k) + (x_k - \bar{\mu}_k)^T \bar{\Sigma}_k^{-1} (x_k - \bar{\mu}_k)]$$

由于 J_k 为 x_k 的二次型, 同时 $bel(x_k)$ 符合高斯分布, 继续通过求解一阶导数和二阶导数的方法求均值和方差。

一阶导数:

$$\frac{\partial J_k}{\partial x_k} = -H_k^T R_k^{-1} (z_k - H_k x_k) + \bar{\Sigma}_k^{-1} (x_k - \bar{\mu}_k)$$

令一阶导数等于零

$$\begin{aligned}
\frac{\partial J_k}{\partial x_k} &= 0 \\
H_k^T R_k^{-1} (z_k - H_k x_k) &= \bar{\Sigma}_k^{-1} (x_k - \bar{\mu}_k) \\
(H_k^T R_k^{-1} H_k + \bar{\Sigma}_k^{-1}) x_k &= H_k^T R_k^{-1} z_k + \bar{\Sigma}_k^{-1} \bar{\mu}_k \\
x_k &= \bar{\mu}_k + \frac{H_k^T R_k^{-1}}{H_k^T R_k^{-1} H_k + \bar{\Sigma}_k^{-1}} (z_k - H_k \bar{\mu}_k)
\end{aligned}$$

令

$$K_k = \frac{H_k^T R_k^{-1}}{H_k^T R_k^{-1} H_k + \bar{\Sigma}_k^{-1}} \quad (3)$$

化简 K_k 可以得到一个更常见的形式：

$$\begin{aligned} K_k &= \frac{H_k^T R_k^{-1} \bar{\Sigma}_k}{H_k^T R_k^{-1} H_k \bar{\Sigma}_k + I} \\ &= \frac{H_k^T R_k^{-1} \bar{\Sigma}_k}{H_k^T R_k^{-1} H_k \bar{\Sigma}_k + I} \\ &= \frac{R_k^{-1} \bar{\Sigma}_k}{R_k^{-1} H_k \bar{\Sigma}_k + (H_k^T)^{-1}} \\ &= \frac{\bar{\Sigma}_k}{H_k \bar{\Sigma}_k + R_k (H_k^T)^{-1}} \\ &= \frac{\bar{\Sigma}_k H_k^T}{H_k \bar{\Sigma}_k H_k^T + R_k} \end{aligned}$$

则

$$\mu_k = \bar{\mu}_k + K_k(z_k - H_k \bar{\mu}_k) \quad (4)$$

二阶导数：

$$\Sigma_k^{-1} = \frac{\partial^2 J}{\partial x_k^2} = H_k^T R_k^{-1} H_k + \bar{\Sigma}_k^{-1}$$

将 K_k 代入 Σ_k 的表达式中，可以得到更加简化的表达形式

$$\Sigma_k = (I - K_k H_k) \bar{\Sigma}_k \quad (5)$$

结论

至此，卡尔曼滤波器的五条公式全部推出，分别为式（1~5），即：

$$\begin{aligned} \bar{\mu}_k &= F_k u_{k-1} + B_k u_k \\ \bar{\Sigma}_k &= F_k \Sigma_{k-1} F_k^T + Q_k \\ K_k &= \frac{\bar{\Sigma}_k H_k^T}{H_k \bar{\Sigma}_k H_k^T + R_k} \\ \mu_k &= \bar{\mu}_k + K_k(z_k - H_k \bar{\mu}_k) \\ \Sigma_k &= (I - K_k H_k) \bar{\Sigma}_k \end{aligned}$$

可以看到，卡尔曼滤波器即是在贝叶斯滤波器的基础上加入了带有噪声的模型。观察 K_k 可以发现

$$\begin{aligned} \lim_{R_k \rightarrow 0} K_k &= H_k^{-1} \\ \lim_{\Sigma_k \rightarrow 0} K_k &= 0 \end{aligned}$$

即当观测误差趋于 0 时，引入观测值进行修正的权重将增大；当预测误差趋于 0 时，引入观测值进行修正的权重将减小并趋于 0。

卡尔曼滤波器的实现

卡尔曼滤波器通过如下的算法进行实现：

```

1:      Algorithm Kalman_filter( $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):
2:       $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$ 
3:       $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$ 
4:       $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$ 
5:       $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$ 
6:       $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$ 
7:      return  $\mu_t, \Sigma_t$ 

```

接下来在倒立摆模型中实现卡尔曼滤波器，倒立摆模型预测方程近似如下：

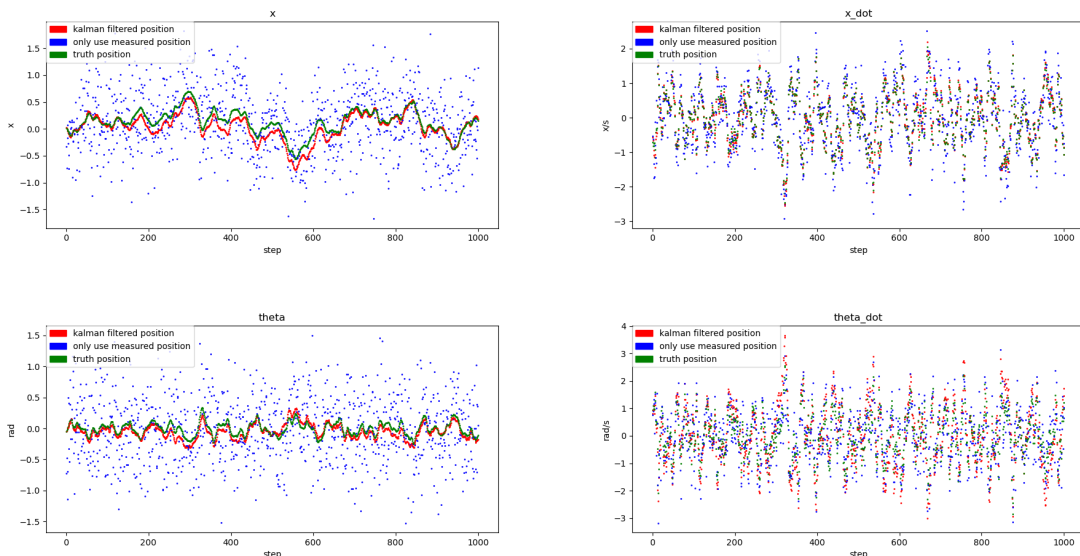
$$\begin{bmatrix} \theta_{k+1} \\ \dot{\theta}_{k+1} \\ x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_k \\ \dot{\theta}_k \\ x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} 0 \\ k \\ 0 \\ \frac{1-mlk}{M+m} \end{bmatrix} F$$

其中 $x, \theta, \dot{x}, \dot{\theta}$ 为小车位移、杆角度、小车速度、杆角速度， M, m 为小车重量、杆重量， l 为杆长度， F 为向小车施加的力， t 在离散系统中为两次更新间经过的时间， $k = -1/(l(4/3 - m/(m + M))(m + M))$ 。

在实现中一般不希望或不能观测所有的变量，而且在观测中会引入误差，假设只有小车速度和杆角速度能被测量，观测方程如下：

$$\begin{bmatrix} \theta_{k+1} \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{k+1} \\ \dot{\theta}_{k+1} \\ x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix}$$

使用卡尔曼滤波器实现状态估计。使用 Open AI 的开源库 gym 中的 CartPole-v1 进行仿真，使用 PID 算法进行控制。仿真结果如下图：



卡尔曼滤波器的 python 代码如下：

```

1 class KFilter:
2     def __init__(self, f_mat, b_mat, q_mat, h_mat, r_mat):
3         self.f_mat = f_mat
4         self.b_mat = b_mat

```

```
5         self.q_mat = q_mat
6         self.h_mat = h_mat
7         self.r_mat = r_mat
8
9     def kal_filter(self, x_mat, p_mat, z_mat, action):
10         x_predict = self.f_mat * x_mat + self.b_mat * action
11         p_predict = self.f_mat * p_mat * self.f_mat.T + self.q_mat
12         k_num = p_predict * self.h_mat.T * np.linalg.pinv(self.h_mat *
13 p_predict * self.h_mat.T + self.r_mat)
14         x_mat = x_predict + k_num * (z_mat - self.h_mat * x_predict)
15         p_mat = (np.eye(4) - k_num * self.h_mat) * p_predict
16         return x_mat, p_mat
```

倒立摆的 PID 算法修改自 [GitHub](#)

参考文献

- [1] [Wikipedia](#)
- [2] [卡尔曼滤波器的数学推导](#)
- [3] [The PID control implementation of cart-pole environment in gym](#)