深蓝学院
shenlanxueyuan.com

机器人学中的状态估计

Timothy Barfoot 著　高翔，谢晓佳等 译
Slides by Xiang Gao

2020年春

深蓝学院
shenlanxueyuan.com

第2讲 线性高斯系统的状态估计问题

- 离散时间的批量估计
- 离散时间的递归平滑算法
- 离散时间的滤波算法
- 连续时间的批量估计

深蓝学院
shenlanxueyuan.com

第2讲 线性高斯系统的状态估计问题

- 离散时间的批量估计
- 离散时间的递归平滑算法
- 离散时间的滤波算法
- 连续时间的批量估计

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 上节课中，我们介绍了：
  - 概率密度函数的定义和各种性质
  - 这些性质在高斯分布上的情况
  - 高斯推断
  - 高斯分布的线性变换和非线性变换，非线性变换的线性化

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 本节课的内容：
  - 线性高斯系统
  - 批量问题的解法
  - 递归式问题的解法
  - 卡尔曼滤波器

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 考虑离散时间的线性时变系统（Discrete-time, linear, time-varying）

$$\text{运动方程：} \boldsymbol{x}_k = \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{v}_k + \boldsymbol{w}_k, \quad k = 1, \cdots, K$$

$$\text{观测方程：} \boldsymbol{y}_k = \boldsymbol{C}_k\boldsymbol{x}_k + \boldsymbol{n}_k, \quad k = 0, \cdots, K$$

- 各个变量含义：

系统状态：$\boldsymbol{x}_k \in \mathbb{R}^N$

初始状态：$\boldsymbol{x}_0 \in \mathbb{R}^N \sim \mathcal{N}(\check{\boldsymbol{x}}_0, \check{\boldsymbol{P}}_0)$

输入：$\boldsymbol{v}_k \in \mathbb{R}^N$

过程噪声：$\boldsymbol{w}_k \in \mathbb{R}^N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$

测量：$\boldsymbol{y}_k \in \mathbb{R}^M$

测量噪声：$\boldsymbol{n}_k \in \mathbb{R}^M \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_k)$

而且：

- 除了 $\boldsymbol{v}_k$ 以外，其他变量都是随机变量
- 各时刻的噪声都不相关
- A称为转移矩阵
- C称为观测矩阵

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

白话：k时刻的状态等于k-1时刻的状态乘上k-1时刻的转换矩阵，加上k时刻的控制量（注意控制量不是随机变量，而是一个定值），最后再加上控制量的噪声。
K时刻的观测量等于k时刻的状态乘上观测矩阵，再加上观测噪声。

- 除系统模型之外，还知道：

运动方程：$\boldsymbol{x}_k = \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{v}_k + \boldsymbol{w}_k, \quad k = 1, \cdots, K$

观测方程：$\boldsymbol{y}_k = \boldsymbol{C}_k\boldsymbol{x}_k + \boldsymbol{n}_k, \quad k = 0, \cdots, K$

1. 初始状态 $\check{\boldsymbol{x}}_0$，以及相应的初始协方差矩阵 $\check{\boldsymbol{P}}_0$。有时候我们也不知道初始信息，那就必须在没有初始信息的情况下进行推导[2]。
2. 输入量 $\boldsymbol{v}_k$，通常来自控制器，是已知的[3]；它的噪声协方差矩阵是 $\boldsymbol{Q}_k$。
3. 观测数据 $\boldsymbol{y}_{k,\text{meas}}$ 是观测变量 $\boldsymbol{y}_k$ 的一次实现（realization），它的协方差为 $\boldsymbol{R}_k$。

- 状态估计问题：通过初始状态、各时刻的观测数据、输入数据，估计系统的真实状态

□ 带下帽子的变量称为先验，上帽子称为后验

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 状态估计问题的分类：
  - 系统是否为线性的？ 线性系统/非线性系统
  - 噪声是否为高斯的？ 高斯系统/非高斯系统
  - 问题是批量式还是递归式？ Batch/Recursive方法
- 我们从最简单批量线性高斯系统（Batch Linear Gaussian）开始介绍，逐步扩展范围
- 对于批量LG系统，有两类方法可以估计其状态：
  1. 贝叶斯推断（Bayesian Inference）
  2. 最大后验估计（Maximum A Posteriori, MAP）
- 接下来我们分别介绍这两种方法。先讲MAP方法，再讲贝叶斯方法。

深蓝学院 shenlanxueyuan.com

一句话概括MAP的结果，就是：最大似然估计转化为最小二乘问题
步骤：
1、写出目标函数 x(后验估计） = argmax p(x|v,y)
2、贝叶斯公式把xy进行换序 argmax p(x|v)p(y|x)
3、因式分解，每项展开
4、取对数，乘法变加法。变成高斯分布指数加和形式（马氏距离平方和）
5、写成提升形式

离散时间的批量估计问题

- 1. MAP方法：已知输入和观测时，求最大概率的状态

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{v}, \boldsymbol{y})$$

- 把这种不带下标的变量定义为宏观变量：

$$\boldsymbol{x} = \boldsymbol{x}_{0:K} = (\boldsymbol{x}_0, \cdots, \boldsymbol{x}_K), \quad \boldsymbol{v} = (\check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:K}) = (\check{\boldsymbol{x}}, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_K)$$

$$\boldsymbol{y} = \boldsymbol{y}_{0:K} = (\boldsymbol{y}_0, \cdots, \boldsymbol{y}_K)$$

- 首先用贝叶斯公式重写目标函数：

通俗地讲，贝叶斯公式用于交换竖线“|”两边的顺序

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{v}, \boldsymbol{y}) = \arg\max_{\boldsymbol{x}} \frac{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{v})\, p(\boldsymbol{x}|\boldsymbol{v})}{p(\boldsymbol{y}|\boldsymbol{v})} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x})\, p(\boldsymbol{x}|\boldsymbol{v})$$

- 分母与x无关，舍去

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{y}) = \arg\max_{\boldsymbol{x}} \frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{v})\,p(\boldsymbol{x}|\boldsymbol{v})}{p(\boldsymbol{y}|\boldsymbol{v})} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x})\,p(\boldsymbol{x}|\boldsymbol{v})$$

- 由于各时刻观测、输入的噪声都是无关的，上面两个项可以因式分解：

$$p(\boldsymbol{x}|\boldsymbol{v}) = p(\boldsymbol{x}_0|\check{\boldsymbol{x}}_0)\prod_{k=1}^{K} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{v}_k) \qquad p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=0}^{K} p(\boldsymbol{y}_k|\boldsymbol{x}_k)$$

- 同时，对目标函数取对数，对数是个单调映射，不影响最优解：

$$\ln\left(p(\boldsymbol{y}|\boldsymbol{x})\,p(\boldsymbol{x}|\boldsymbol{v})\right) = \ln p(\boldsymbol{x}_0|\check{\boldsymbol{x}}_0) + \sum_{k=1}^{K}\ln p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1},\boldsymbol{v}_k) + \sum_{k=0}^{K}\ln p(\boldsymbol{y}_k|\boldsymbol{x}_k)$$

- 这时因子相乘变成了对数项相加

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 高斯分布取对数之后有较好形式：

$$\ln p(\boldsymbol{x}_0|\check{\boldsymbol{x}}_0) = -\frac{1}{2}(\boldsymbol{x}_0 - \check{\boldsymbol{x}}_0)^{\mathrm{T}}\check{\boldsymbol{P}}_0^{-1}(\boldsymbol{x} - \check{\boldsymbol{x}}_0) \underbrace{-\frac{1}{2}\ln\left((2\pi)^N \det \check{\boldsymbol{P}}_0\right)}_{与\ \boldsymbol{x}\ 无关}$$

$$\ln p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{v}_k) = -\frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{v}_k)^{\mathrm{T}}\boldsymbol{Q}_k^{-1}(\boldsymbol{x}_k - \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{v}_k) \underbrace{-\frac{1}{2}\ln\left((2\pi)^N \det \boldsymbol{Q}_k\right)}_{与\ \boldsymbol{x}\ 无关}$$

$$\ln p(\boldsymbol{y}_k|\boldsymbol{x}_k) = -\frac{1}{2}(\boldsymbol{y}_k - \boldsymbol{C}_k\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{R}_k^{-1}(\boldsymbol{y}_k - \boldsymbol{C}_k\boldsymbol{x}_k) \underbrace{-\frac{1}{2}\ln\left((2\pi)^M \det \boldsymbol{R}_k\right)}_{与\ \boldsymbol{x}\ 无关}$$

舍掉与x无关的那些项，定义：

$$J_{v,k}(\boldsymbol{x}) = \begin{cases} \frac{1}{2}(\boldsymbol{x}_0 - \check{\boldsymbol{x}}_0)^{\mathrm{T}}\check{\boldsymbol{P}}_0^{-1}(\boldsymbol{x}_0 - \check{\boldsymbol{x}}_0), & k = 0 \\ \frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{v}_k)^{\mathrm{T}}\boldsymbol{Q}_k^{-1}(\boldsymbol{x}_k - \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} - \boldsymbol{v}_k), & k = 1, \cdots, K \end{cases}$$

$$J_{y,k}(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{y}_k - \boldsymbol{C}_k\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{R}_k^{-1}(\boldsymbol{y}_k - \boldsymbol{C}_k\boldsymbol{x}_k), \quad k = 0, \cdots, K$$

于是目标函数变为求这个式的最小化：

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} J(\boldsymbol{x}) \qquad J(\boldsymbol{x}) = \sum_{k=0}^{K}(J_{v,k}(\boldsymbol{x}) + J_{y,k}(\boldsymbol{x}))$$

这个问题就是常见的无约束最小二乘

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 写成更紧凑的矩阵形式（提升形式）：

$$
\mathbf{z} = \begin{bmatrix} \check{\mathbf{x}}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \\ \hline \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{bmatrix} \quad \mathbf{H} = \left[\begin{array}{ccccc} \mathbf{1} & & & & \\ -\mathbf{A}_0 & \mathbf{1} & & & \\ & \ddots & \ddots & & \\ & & -\mathbf{A}_{K-1} & \mathbf{1} \\ \hline \mathbf{C}_0 & & & \\ & \mathbf{C}_1 & & \\ & & \ddots & \\ & & & \mathbf{C}_K \end{array}\right] \quad \mathbf{W} = \left[\begin{array}{cccc|cccc} \check{\mathbf{P}}_0 & & & & & & & \\ & \mathbf{Q}_1 & & & & & & \\ & & \ddots & & & & & \\ & & & \mathbf{Q}_K & & & & \\ \hline & & & & \mathbf{R}_0 & & & \\ & & & & & \mathbf{R}_1 & & \\ & & & & & & \ddots & \\ & & & & & & & \mathbf{R}_K \end{array}\right]
$$

- 把运动和观测写在一起： $z = Hx + W$

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 提升形式目标函数： $J(\boldsymbol{x})=\frac{1}{2}(\boldsymbol{z}-\boldsymbol{H}\boldsymbol{x})^{\mathrm{T}}\boldsymbol{W}^{-1}(\boldsymbol{z}-\boldsymbol{H}\boldsymbol{x})$
- 它是个二次的，求其最小值，只要令自变量导数为零：

$$\left.\frac{\partial J(\boldsymbol{x})}{\partial \boldsymbol{x}^{\mathrm{T}}}\right|_{\hat{\boldsymbol{x}}}=-\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}(\boldsymbol{z}-\boldsymbol{H}\hat{\boldsymbol{x}})=\mathbf{0}$$

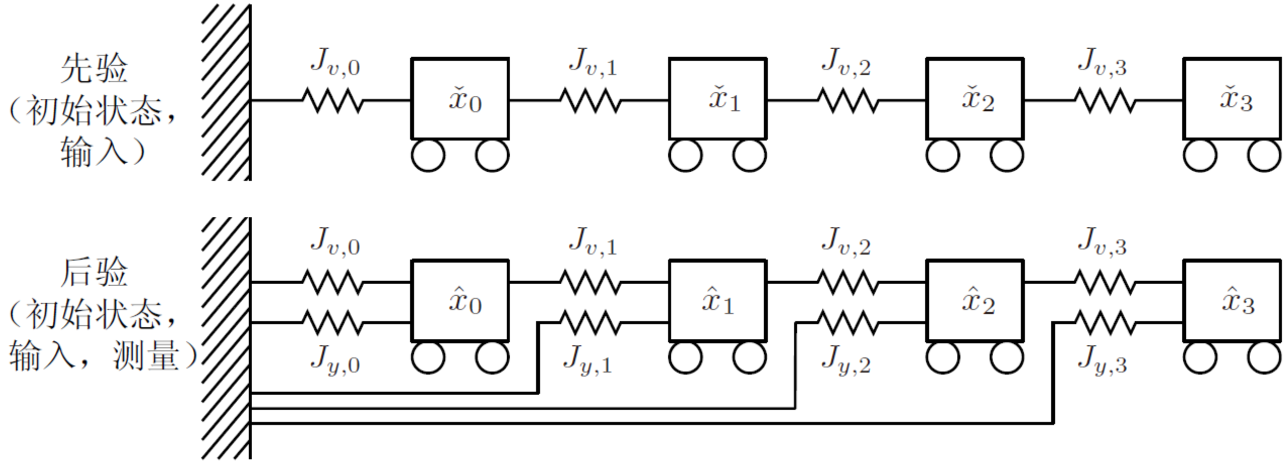
$$\Rightarrow\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H}\right)\hat{\boldsymbol{x}}=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{z}$$

- 于是就解析地得到了最优解
  - 这个解等价于经典的批量最小二乘法，也等价于固定区间平滑算法，或者也等价于伪逆
  - 由于上一页里H具有特殊的稀疏结构，这个问题也有特殊的解法，不需要暴力算矩阵求逆

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 最小二乘的直观解释
- 最小二乘系统可视为弹簧-重物系统，其最优解对应系统最小能量状态
- 可以想象成松开手后，系统最终的稳定状态



深蓝学院
shenlanxueyuan.com

一句话概括贝叶斯推断的结果：在LG系统下，得到后验估计的均值等于后验估计的模。同时还能得到其协方差

步骤：
1、写出先验模型和后验模型的提升形式
2、最关键一步，写出联合分布p(x,y|v)
3、高斯推断，代入公式即可
4、SWM公式，改变公式样子
5、各种矩阵整理

离散时间的批量估计问题

- 2. 贝叶斯推断

- 在LG系统中，可以根据运动方程和观测方程显式写出状态变量分布的变化过程

单个时刻：$\boldsymbol{x}_k = \boldsymbol{A}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{v}_k + \boldsymbol{w}_k$

提升形式：$\boldsymbol{x} = \boldsymbol{A}(\boldsymbol{v} + \boldsymbol{w})$

- 思考：v是什么形式?

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & & & & & \\ \boldsymbol{A}_0 & \mathbf{1} & & & & \\ \boldsymbol{A}_1\boldsymbol{A}_0 & \boldsymbol{A}_1 & \mathbf{1} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \boldsymbol{A}_{K-2}\cdots\boldsymbol{A}_0 & \boldsymbol{A}_{K-2}\cdots\boldsymbol{A}_1 & \boldsymbol{A}_{K-2}\cdots\boldsymbol{A}_2 & \cdots & \mathbf{1} & \\ \boldsymbol{A}_{K-1}\cdots\boldsymbol{A}_0 & \boldsymbol{A}_{K-1}\cdots\boldsymbol{A}_1 & \boldsymbol{A}_{K-1}\cdots\boldsymbol{A}_2 & \cdots & \boldsymbol{A}_{K-1} & \mathbf{1} \end{bmatrix}$$

- 提升形式里，右侧只有v和w，容易求得其均值和协方差：
  - 此处v是确定的，w是高斯的，因此为高斯分布的线性变换

$$\check{\boldsymbol{x}} = E[\boldsymbol{x}] = E[\boldsymbol{A}(\boldsymbol{v}+\boldsymbol{w})] = \boldsymbol{A}\boldsymbol{v}$$

$$\check{\boldsymbol{P}} = E\left[(\boldsymbol{x} - E[\boldsymbol{x}])(\boldsymbol{x} - E[\boldsymbol{x}])^{\mathrm{T}}\right] = \boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^{\mathrm{T}}$$

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 因此，先验部分写为：$p(\boldsymbol{x}|\boldsymbol{v}) = \mathcal{N}(\check{\boldsymbol{x}}, \check{\boldsymbol{P}}) = \mathcal{N}(\boldsymbol{A}\boldsymbol{v}, \boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^{\mathrm{T}})$
  - 先验的意思是：仅考虑运动方程时的条件概率分布
- 再看观测模型：

$$\boldsymbol{y}_k = \boldsymbol{C}_k \boldsymbol{x}_k + \boldsymbol{n}_k$$

单次观测

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{n}$$

提升形式

$$\boldsymbol{C} = \operatorname{diag}(\boldsymbol{C}_0, \boldsymbol{C}_1, \cdots, \boldsymbol{C}_K)$$

- 于是联合分布写为：

$$p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{v}) = \mathcal{N}\left(\begin{bmatrix} \check{\boldsymbol{x}} \\ \boldsymbol{C}\check{\boldsymbol{x}} \end{bmatrix}, \begin{bmatrix} \check{\boldsymbol{P}} & \check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}} \\ \boldsymbol{C}\check{\boldsymbol{P}} & \boldsymbol{C}\check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R} \end{bmatrix}\right)$$

- 已知v时，x的先验分布已经确定，于是y的分布亦可确定

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 已知联合分布，欲求条件分布，如何求?
  - 还记得上一节的高斯推断吗？联合=条件 * 边缘

$$p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{v}) = p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{y})p(\boldsymbol{y}|\boldsymbol{v})$$

$$p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{v}) = \mathcal{N}\left(\begin{bmatrix}\check{\boldsymbol{x}}\\ \boldsymbol{C}\check{\boldsymbol{x}}\end{bmatrix}, \begin{bmatrix}\check{\boldsymbol{P}} & \check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}}\\ \boldsymbol{C}\check{\boldsymbol{P}} & \boldsymbol{C}\check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{R}\end{bmatrix}\right)$$

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

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_y), \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}\right)$$

$$p(\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_{yy}\right)$$

- 联合分布
- 条件分布
- 边缘分布

高斯推断

  - 逐一代入，立得：

$$p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{y}) = \mathcal{N}\left(\check{\boldsymbol{x}} + \check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}}\left(\boldsymbol{C}\check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}(\boldsymbol{y}-\boldsymbol{C}\check{\boldsymbol{x}}),\right.$$
$$\left.\check{\boldsymbol{P}} - \check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}}\left(\boldsymbol{C}\check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}\boldsymbol{C}\check{\boldsymbol{P}}\right)$$

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 代入SMW式进行化简：

$$p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{y}) = \mathcal{N}\Big(\check{\boldsymbol{x}} + \check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}}\left(\boldsymbol{C}\check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{y} - \boldsymbol{C}\check{\boldsymbol{x}}\right),\ \check{\boldsymbol{P}} - \check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}}\left(\boldsymbol{C}\check{\boldsymbol{P}}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R}\right)^{-1}\boldsymbol{C}\check{\boldsymbol{P}}\Big)$$

- 得到： $$p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{y}) = \mathcal{N}\Big(\underbrace{\left(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)^{-1}\left(\check{\boldsymbol{P}}^{-1}\check{\boldsymbol{x}} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}\right)}_{\text{即均值 } \hat{\boldsymbol{x}}},\ \underbrace{\left(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)^{-1}}_{\text{即后验协方差 } \hat{\boldsymbol{P}}}\Big)$$

□ SMW(a)和(c)式

$$\boldsymbol{A}\boldsymbol{B}\left(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\right)^{-1} \equiv \left(\boldsymbol{A}^{-1} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{B}\boldsymbol{D}^{-1}$$

$$\left(\boldsymbol{A}^{-1} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1} \equiv \boldsymbol{A} - \boldsymbol{A}\boldsymbol{B}\left(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\right)^{-1}\boldsymbol{C}\boldsymbol{A}$$

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 来整理这里的均值项和协方差：

$$p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{y}) = \mathcal{N}\Bigg(\underbrace{\left(\check{\boldsymbol{P}}^{-1}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)^{-1}\left(\check{\boldsymbol{P}}^{-1}\check{\boldsymbol{x}}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}\right)}_{\text{即均值 } \hat{\boldsymbol{x}}}, \underbrace{\left(\check{\boldsymbol{P}}^{-1}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)^{-1}}_{\text{即后验协方差 } \hat{\boldsymbol{P}}}\Bigg)$$

均值部分：$\underbrace{\left(\check{\boldsymbol{P}}^{-1}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)}_{\hat{\boldsymbol{P}}^{-1}}\hat{\boldsymbol{x}} = \check{\boldsymbol{P}}^{-1}\check{\boldsymbol{x}}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}$

代入 $\check{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{v}$ 和 $\check{\boldsymbol{P}}^{-1} = (\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^{\mathrm{T}})^{-1} = \boldsymbol{A}^{-\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{A}^{-1}$,

得：$\underbrace{\left(\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{A}^{-1}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)}_{\hat{\boldsymbol{P}}^{-1}}\hat{\boldsymbol{x}} = \boldsymbol{A}^{-\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{v}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}$

- 由于A的结构，A逆有特殊形式：

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & & & & & \\ \boldsymbol{A}_0 & \mathbf{1} & & & & \\ \boldsymbol{A}_1\boldsymbol{A}_0 & \boldsymbol{A}_1 & \mathbf{1} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \boldsymbol{A}_{K-2}\cdots\boldsymbol{A}_0 & \boldsymbol{A}_{K-2}\cdots\boldsymbol{A}_1 & \boldsymbol{A}_{K-2}\cdots\boldsymbol{A}_2 & \cdots & \mathbf{1} & \\ \boldsymbol{A}_{K-1}\cdots\boldsymbol{A}_0 & \boldsymbol{A}_{K-1}\cdots\boldsymbol{A}_1 & \boldsymbol{A}_{K-1}\cdots\boldsymbol{A}_2 & \cdots & \boldsymbol{A}_{K-1} & \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{A}^{-1} = \begin{bmatrix} \mathbf{1} & & & & & \\ -\boldsymbol{A}_0 & \mathbf{1} & & & & \\ & -\boldsymbol{A}_1 & \mathbf{1} & & & \\ & & -\boldsymbol{A}_2 & \ddots & & \\ & & & \ddots & \mathbf{1} & \\ & & & & -\boldsymbol{A}_{K-1} & \mathbf{1} \end{bmatrix}$$

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 按照均值式： $\underbrace{\left(A^{-\mathrm{T}}Q^{-1}A^{-1}+C^{\mathrm{T}}R^{-1}C\right)}_{\hat{P}^{-1}}\hat{x}=A^{-\mathrm{T}}Q^{-1}v+C^{\mathrm{T}}R^{-1}y$
- 定义一些矩阵： $z=\begin{bmatrix} v \\ y \end{bmatrix},\quad H=\begin{bmatrix} A^{-1} \\ C \end{bmatrix},\quad W=\begin{bmatrix} Q & \\ & R \end{bmatrix}$
- 于是得： $\left(H^{\mathrm{T}}W^{-1}H\right)\hat{x}=H^{\mathrm{T}}W^{-1}z$
- 这个结论与MAP结果完全一致！

□ 回到MAP页

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- MAP结果和贝叶斯推断结果一致，说明了什么?
  - MAP只关心达到最大后验概率的一个点，这个点的状态称为MAP估计
  - 而贝叶斯推断写出了p(x|y,v)的完整形式，它是一个高斯分布，其均值与MAP估计相等；同时，给出了这个估计的协方差
  - 如果我们只关心状态估计变量取值，那么MAP给出了后验分布的模（Mode），贝叶斯推断给出了均值
  - 而在LG系统中，二者是一样的，使得这两类方法给出了同样的结果

系统的能观性：按问题来索引
1、用大白话说出什么是系统的能观性？
2、如何推导能观性的条件，结论是什么？
3、如果有先验，会是什么情况？
4、如果没有先验，又会是什么情况？

深蓝学院 shenlanxueyuan.com

离散时间的批量估计问题

- LG系统最优估计什么时候存在？什么时候唯一？

$$\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H}\right)\hat{\boldsymbol{x}}=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{z} \qquad \hat{\boldsymbol{x}}=\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{z}$$

- 显然这要求左侧部分可逆；
- x的维度：每时刻状态维度为N，共 0, … , K 总计K+1个时刻，因此 dim(x)=N(K+1)
- 所以可逆性要求： $\operatorname{rank}\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H}\right)=N(K+1)$
- 又由于协方差矩阵的对称正定性，即要求： $\operatorname{rank}\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\right)=\operatorname{rank}\left(\boldsymbol{H}^{\mathrm{T}}\right)=N(K+1)$
- H的具体形式取决于问题有没有0时刻的先验条件，对其分类讨论之。

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 1. 存在先验条件

$$
\begin{aligned}
&\operatorname{rank} \boldsymbol{H}^{\mathrm{T}} \\
=&\operatorname{rank}\left[\begin{array}{ccccc|ccccc}
\mathbf{1} & -\boldsymbol{A}_0^{\mathrm{T}} & & & & \boldsymbol{C}_0^{\mathrm{T}} & & & & \\
& \mathbf{1} & -\boldsymbol{A}_1^{\mathrm{T}} & & & & \boldsymbol{C}_1^{\mathrm{T}} & & & \\
& & \mathbf{1} & \ddots & & & & \boldsymbol{C}_2^{\mathrm{T}} & & \\
& & & \ddots & -\boldsymbol{A}_{K-1}^{\mathrm{T}} & & & & \ddots & \\
& & & & \mathbf{1} & & & & & \boldsymbol{C}_K^{\mathrm{T}}
\end{array}\right]
\end{aligned}
$$

- 显然这个是满秩的（每行有非零块打头），所以只要有先验，最优解就是存在的（且唯一）
- 大前提是噪声协方差为对称正定阵（默认成立）

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 2. 不存在先验
  - 这个矩阵阶梯部分必然是满秩的，主要问题在于第一行
  - 把第一行挪到最后 $\boldsymbol{A}_0^{\mathrm{T}}$，不影响矩阵的秩
  - 然后，用第一行乘 $\boldsymbol{A}_0^{\mathrm{T}}\boldsymbol{A}_1^{\mathrm{T}}$ 加至最后一行，再用第二行乘加至最后一行，依此类推；这些都是初等行变换，不影响矩阵求秩
  - 于是得：

$$
\operatorname{rank} \boldsymbol{H}^{\mathrm{T}}
= \operatorname{rank}\left[\begin{array}{cccc|ccccc}
\mathbf{1} & -\boldsymbol{A}_1^{\mathrm{T}} & & & \boldsymbol{C}_1^{\mathrm{T}} & & & & \\
 & \mathbf{1} & \ddots & & & \boldsymbol{C}_2^{\mathrm{T}} & & & \\
 & & \ddots & -\boldsymbol{A}_{K-1}^{\mathrm{T}} & & & \ddots & & \\
 & & & \mathbf{1} & & & & & \boldsymbol{C}_K^{\mathrm{T}} \\
\hline
 & & & & \boldsymbol{C}_0^{\mathrm{T}} & \boldsymbol{A}_0^{\mathrm{T}}\boldsymbol{C}_1^{\mathrm{T}} & \boldsymbol{A}_0^{\mathrm{T}}\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{C}_2^{\mathrm{T}} & \cdots & \boldsymbol{A}_0^{\mathrm{T}}\cdots\boldsymbol{A}_{K-1}^{\mathrm{T}}\boldsymbol{C}_K^{\mathrm{T}}
\end{array}\right]
$$

$$
\operatorname{rank} \boldsymbol{H}^{\mathrm{T}}
= \operatorname{rank}\left[\begin{array}{cccc|cccc}
-\boldsymbol{A}_0^{\mathrm{T}} & & & & \boldsymbol{C}_0^{\mathrm{T}} & & & \\
\hline
\mathbf{1} & -\boldsymbol{A}_1^{\mathrm{T}} & & & \boldsymbol{C}_1^{\mathrm{T}} & & & \\
 & \mathbf{1} & \ddots & & & \boldsymbol{C}_2^{\mathrm{T}} & & \\
 & & \ddots & -\boldsymbol{A}_{K-1}^{\mathrm{T}} & & & \ddots & \\
 & & & \mathbf{1} & & & & \boldsymbol{C}_K^{\mathrm{T}}
\end{array}\right]
$$

$$
\operatorname{rank} \boldsymbol{H}^{\mathrm{T}}
= \operatorname{rank}\left[\begin{array}{cccc|cccc}
\mathbf{1} & -\boldsymbol{A}_1^{\mathrm{T}} & & & \boldsymbol{C}_1^{\mathrm{T}} & & & \\
 & \mathbf{1} & \ddots & & & \boldsymbol{C}_2^{\mathrm{T}} & & \\
 & & \ddots & -\boldsymbol{A}_{K-1}^{\mathrm{T}} & & & \ddots & \\
 & & & \mathbf{1} & & & & \boldsymbol{C}_K^{\mathrm{T}} \\
\hline
-\boldsymbol{A}_0^{\mathrm{T}} & & & & \boldsymbol{C}_0^{\mathrm{T}} & & &
\end{array}\right]
$$

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 最后判定条件为: $\operatorname{rank}\begin{bmatrix} C_0^{\mathrm{T}} & A_0^{\mathrm{T}}C_1^{\mathrm{T}} & A_0^{\mathrm{T}}A_1^{\mathrm{T}}C_2^{\mathrm{T}} & \cdots & A_0^{\mathrm{T}}\cdots A_{K-1}^{\mathrm{T}}C_K^{\mathrm{T}} \end{bmatrix} = N$
- 如果系统为时不变的，即A,C矩阵随时间不变，且K>>N，那么:

$$\begin{aligned} &\operatorname{rank}\begin{bmatrix} C^{\mathrm{T}} & A^{\mathrm{T}}C^{\mathrm{T}} & A^{\mathrm{T}}A^{\mathrm{T}}C^{\mathrm{T}} & \cdots & \left(A^{\mathrm{T}}\right)^{K}C^{\mathrm{T}} \end{bmatrix} \\ =&\operatorname{rank}\begin{bmatrix} C^{\mathrm{T}} & A^{\mathrm{T}}C^{\mathrm{T}} & \cdots & \left(A^{\mathrm{T}}\right)^{(N-1)}C^{\mathrm{T}} \end{bmatrix} \end{aligned}$$

- 这里要用到卡莱-哈密顿定理（Caylay-Hamilton Theory），该定理说明，矩阵A满足自身的特征方程
- 把rank里面部分称为能观性矩阵: $\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{(N-1)} \end{bmatrix}$
- 能观性矩阵满不满秩，可作为唯一解存在性判断条件

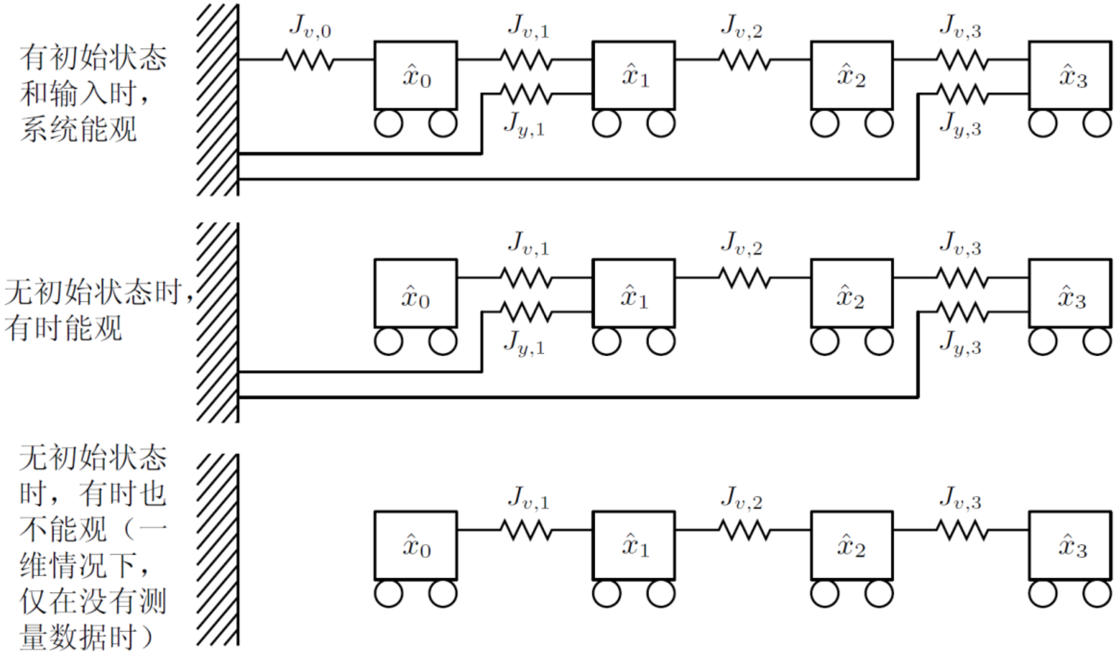
- 矩阵特征方程: $\det(\lambda I - A) = 0$

  是一个关于λ的N次方程。
- Caylay-Hamilton引理说明，把A代入特征方程后，仍然成立
- 这说明A的零次、一次、直到N次是线性相关的，进而大于N次都可以写成0至N-1次的线性组合

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- 能观性的直观解释：能否通过观测变量唯一地确定系统状态
    - 相应地，自控理论中还有能控性的讨论，但状态估计问题中不涉及控制



- 不能观：存在一个或多个自由度无法确定

深蓝学院
shenlanxueyuan.com

离散时间的批量估计问题

- MAP估计的协方差
- 有时候，除了拿到MAP解之外，我们还关心MAP解的置信度（协方差）
- 实际上，在贝叶斯推断时，已经给出了后验协方差的形式：

$$p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{y}) = \mathcal{N}\Bigg(\underbrace{\left(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)^{-1}\left(\check{\boldsymbol{P}}^{-1}\check{\boldsymbol{x}} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}\right)}_{\text{即均值 } \hat{\boldsymbol{x}}}, \underbrace{\left(\check{\boldsymbol{P}}^{-1} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)^{-1}}_{\text{即后验协方差 } \hat{\boldsymbol{P}}}\Bigg)$$

$$\underbrace{\left(\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{A}^{-1} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{C}\right)}_{\hat{\boldsymbol{P}}^{-1}}\hat{\boldsymbol{x}} = \boldsymbol{A}^{-\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{v} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{y}$$

- 所以，实际上这个方程左侧的系数矩阵即为协方差之逆： $\underbrace{\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H}\right)}_{\text{协方差的逆}}\underbrace{\hat{\boldsymbol{x}}}_{\text{均值}} = \underbrace{\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{z}}_{\text{信息向量}}$

□ 所以大家以后可以不用再问MAP解置信度怎么算的问题了

深蓝学院
shenlanxueyuan.com

第2讲 线性高斯系统的状态估计问题

- 离散时间的批量估计
- 离散时间的递归平滑算法
- 离散时间的滤波算法
- 连续时间的批量估计

深蓝学院
shenlanxueyuan.com

概括递归平滑算法：
对线性方程左侧进行Cholesky分解，求解过程分为前向过程和后向过程，前向过程就对应递归平滑算法。

离散时间的递归平滑算法

- 很多在线问题当中（比如定位），我们有上一个时刻的先验估计，希望通过这个时刻的控制和观测，计算这个时刻的状态估计
- 下面我们来推导递归问题的解法。
- 为了推出递归算法，首先来看批量问题的一种特殊的解法。批量问题的核心是解线性方程： $\left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H}\right)\hat{\boldsymbol{x}} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{z}$
- 之前我们提到左侧矩阵不必蛮力求解，因为它有特殊结构（三对角块）：

$$\boldsymbol{H} = \left[\begin{array}{cccc} 1 & & & \\ -\boldsymbol{A}_0 & 1 & & \\ & \ddots & \ddots & \\ & & -\boldsymbol{A}_{K-1} & 1 \\ \hline \boldsymbol{C}_0 & & & \\ & \boldsymbol{C}_1 & & \\ & & \ddots & \\ & & & \boldsymbol{C}_K \end{array}\right] \quad \boldsymbol{W} = \left[\begin{array}{cccc|cccc} \check{\boldsymbol{P}}_0 & & & & & & & \\ & \boldsymbol{Q}_1 & & & & & & \\ & & \ddots & & & & & \\ & & & \boldsymbol{Q}_K & & & & \\ \hline & & & & \boldsymbol{R}_0 & & & \\ & & & & & \boldsymbol{R}_1 & & \\ & & & & & & \ddots & \\ & & & & & & & \boldsymbol{R}_K \end{array}\right]$$

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H} = \begin{bmatrix} * & * & & & \\ * & * & * & & \\ & * & * & * & \\ & & \ddots & \ddots & \ddots \\ & & * & * & * \\ & & & * & * \end{bmatrix}$$

- 这一步请同学们务必自己验证一下（只需验证有无即可）

离散时间的递归平滑算法

为什么只提Cholesky而不提其他的?

- 因为递归方法可以从Cholesky解法引出

- 求解这类方程的一种方式是Cholesky分解
  - 注意这只是其中一种求解的方式，当然还有其他的解方程方法
  - Cholesky分解 $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H}=\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$，对于一般矩阵而言，L是下三角阵
  - 但是对于三对角块矩阵而言，L的形式更简单：

$$\boldsymbol{L}=\begin{bmatrix} * & & & & & \\ * & * & & & & \\ & * & * & & & \\ & & \ddots & \ddots & & \\ & & & * & * & \\ & & & & * & * \end{bmatrix}$$

- Cholesky解方程的手段：
  - 先解： $\boldsymbol{L}\boldsymbol{d}=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{z}$ 得到d，从上往下解；
  - 再解：$\boldsymbol{L}^{\mathrm{T}}\hat{\boldsymbol{x}}=\boldsymbol{d}$ 得到最优状态，从下往上解；
  - 注意这种解法对一般线性方程也是有效的，不光是针对状态估计问题
  - 这两步分别称为前向过程和后向过程（forward/backward），我们以例子来说明

  - 下面几页会涉及到一些计算细节，部分公式以手写内容注明

深蓝学院
shenlanxueyuan.com

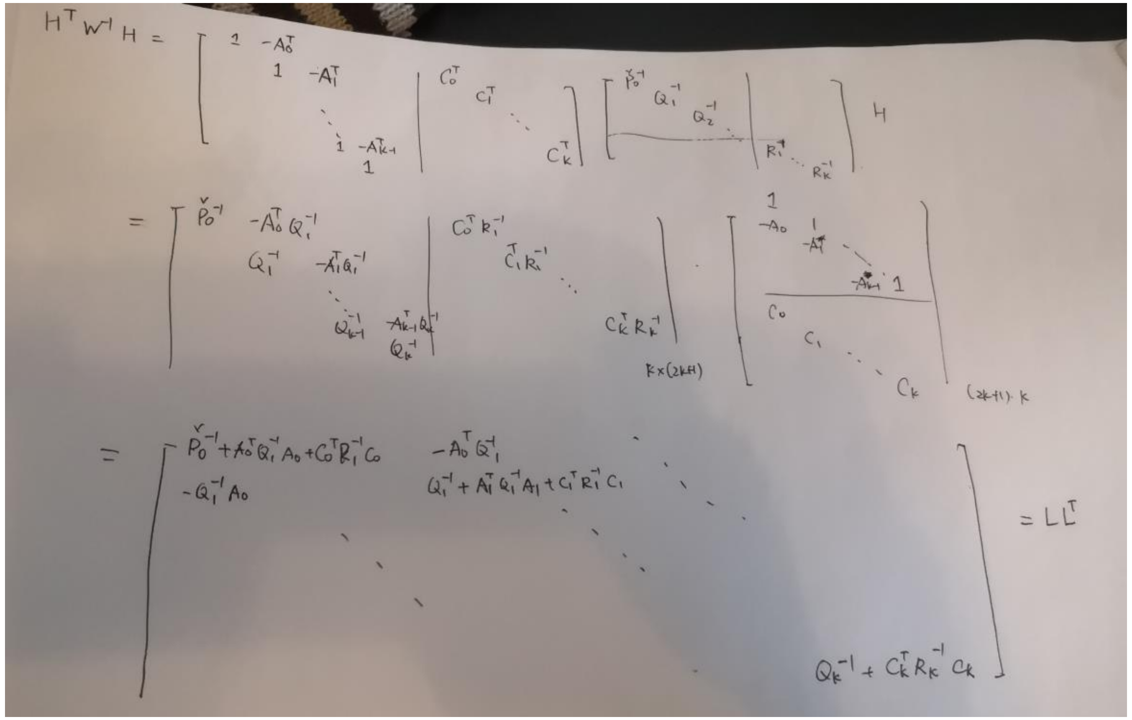
离散时间的递归平滑算法

- L的形式：

$$
\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_0 & & & & & \\ \boldsymbol{L}_{10} & \boldsymbol{L}_1 & & & & \\ & \boldsymbol{L}_{21} & \boldsymbol{L}_2 & & & \\ & & \ddots & \ddots & & \\ & & & \boldsymbol{L}_{K-1,K-2} & \boldsymbol{L}_{K-1} & \\ & & & & \boldsymbol{L}_{K,K-1} & \boldsymbol{L}_K \end{bmatrix}
$$

第1步是Cholesky分解： $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{-1}\boldsymbol{H} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$

$$
\boldsymbol{H} = \left[\begin{array}{ccccc} 1 & & & & \\ -\boldsymbol{A}_0 & 1 & & & \\ & \ddots & \ddots & & \\ & & & -\boldsymbol{A}_{K-1} & 1 \\ \hline \boldsymbol{C}_0 & & & & \\ & \boldsymbol{C}_1 & & & \\ & & & \ddots & \\ & & & & \boldsymbol{C}_K \end{array}\right]
\qquad
\boldsymbol{W} = \left[\begin{array}{cccc|cccc} \check{\boldsymbol{P}}_0 & & & & & & & \\ & \boldsymbol{Q}_1 & & & & & & \\ & & \ddots & & & & & \\ & & & \boldsymbol{Q}_K & & & & \\ \hline & & & & \boldsymbol{R}_0 & & & \\ & & & & & \boldsymbol{R}_1 & & \\ & & & & & & \ddots & \\ & & & & & & & \boldsymbol{R}_K \end{array}\right]
$$



- 把这步写开，见上

离散时间的递归平滑算法

- 对比左右块：

$$L_0 L_0^T = \underbrace{\check{P}_0^{-1} + C_0^T R_0^{-1} C_0}_{I_0} + A_0^T Q_1^{-1} A_0 \qquad (3.61a)$$

$$L_{10} L_0^T = -Q_1^{-1} A_0 \qquad (3.61b)$$

$$L_1 L_1^T = \underbrace{-L_{10} L_{10}^T + Q_1^{-1} + C_1^T R_1^{-1} C_1}_{I_1} + A_1^T Q_2^{-1} A_1 \qquad (3.61c)$$

$$L_{21} L_1^T = -Q_2^{-1} A_1 \qquad (3.61d)$$

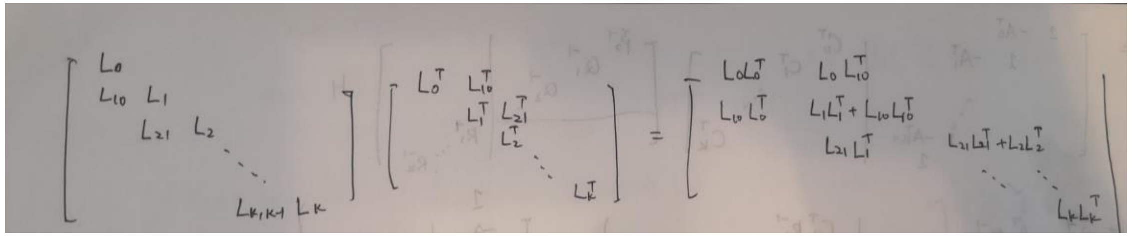
$$\vdots$$

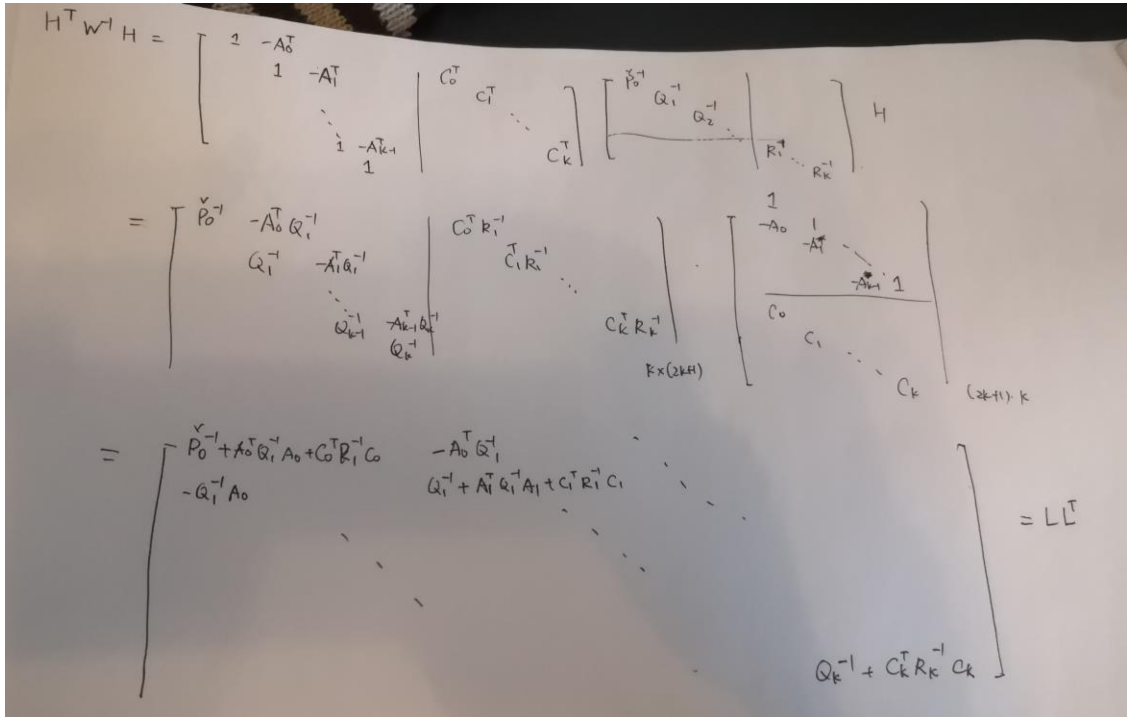
$$L_{K-1} L_{K-1}^T = \underbrace{-L_{K-1,K-2} L_{K-1,K-2}^T + Q_{K-1}^{-1} + C_{K-1}^T R_{K-1}^{-1} C_{K-1}}_{I_{K-1}} + A_{K-1}^T Q_K^{-1} A_{K-1} \qquad (3.61e)$$

$$L_{K,K-1} L_{K-1}^T = -Q_K^{-1} A_{K-1} \qquad (3.61f)$$

$$L_K L_K^T = \underbrace{-L_{K,K-1} L_{K,K-1}^T + Q_K^{-1} + C_K^T R_K^{-1} C_K}_{I_K} \qquad (3.61g)$$

- 这里的I是中间变量，后面要用（且有物理意义）
- 解L的过程就是从上至下，把上面结果代入下面方程
- 于是就解出了L





深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 第2步，解方程 $Ld = H^T W^{-1} z$ ，未知量是d

$$L_0 d_0 = \underbrace{\check{P}_0^{-1} \check{x}_0 + C_0^T R_0^{-1} y_0}_{q_0} - A_0^T Q_1^{-1} v_1$$

$$L_1 d_1 = \underbrace{-L_{10} d_0 + Q_1^{-1} v_1 + C_1^T R_1^{-1} y_1}_{q_1} - A_1^T Q_2^{-1} v_2$$

$$\vdots$$

$$L_{K-1} d_{K-1} = \underbrace{-L_{K-1,K-2} d_{K-2} + Q_{K-1}^{-1} v_{K-1} + C_{K-1}^T R_{K-1}^{-1} y_{K-1}}_{q_{K-1}} - A_{K-1}^T Q_K^{-1} v_K$$

$$L_K d_K = \underbrace{-L_{K,K-1} d_{K-1} + Q_K^{-1} v_K + C_K^T R_K^{-1} y_K}_{q_K}$$

- 这里q依然是中间变量（且有物理意义）
- 这一串方程依然从上往下解，得到各个d

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 第3步，解 $L^{\mathrm{T}}\hat{x} = d$
- 和前面基本一样：

$$
\begin{aligned}
L_K^{\mathrm{T}}\hat{x}_K &= d_K \\
L_{K-1}^{\mathrm{T}}\hat{x}_{K-1} &= -L_{K,K-1}^{\mathrm{T}}\hat{x}_K + d_{K-1} \\
&\vdots \\
L_1^{\mathrm{T}}\hat{x}_1 &= -L_{21}^{\mathrm{T}}\hat{x}_2 + d_1 \\
L_0^{\mathrm{T}}\hat{x}_0 &= -L_{10}^{\mathrm{T}}\hat{x}_1 + d_0
\end{aligned}
$$

- 但这个是从后往前的

总结：Cholesky方法共3步

1. 解L： $H^{\mathrm{T}}W^{-1}H = LL^{\mathrm{T}}$
2. 解d： $Ld = H^{\mathrm{T}}W^{-1}z$
3. 解x： $L^{\mathrm{T}}\hat{x} = d$

第1，2步是正向的，第3步是反向的；

整个算法由一次正向迭代和一次反向迭代组成，下面来归纳它的过程

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 把中间量I和q也写进来，整理算法流程：
- 初始值： $I_0 = \check{P}_0^{-1} + C_0^{\mathrm{T}} R_0^{-1} C_0$

$$q_0 = \check{P}_0^{-1}\check{x}_0 + C_0^{\mathrm{T}} R_0^{-1} y_0$$

- 前向迭代：

前向： $k = 1, \cdots, K$

$$\begin{aligned} L_{k-1} L_{k-1}^{\mathrm{T}} &= I_{k-1} + A_{k-1}^{\mathrm{T}} Q_k^{-1} A_{k-1} \\ L_{k-1} d_{k-1} &= q_{k-1} - A_{k-1}^{\mathrm{T}} Q_k^{-1} v_k \\ L_{k,k-1} L_{k-1}^{\mathrm{T}} &= -Q_k^{-1} A_{k-1} \\ I_k &= -L_{k,k-1} L_{k,k-1}^{\mathrm{T}} + Q_k^{-1} + C_k^{\mathrm{T}} R_k^{-1} C_k \\ q_k &= -L_{k,k-1} d_{k-1} + Q_k^{-1} v_k + C_k^{\mathrm{T}} R_k^{-1} y_k \end{aligned}$$

后向初始值： $\hat{x}_K = L_K^{-\mathrm{T}} d_K$

后向迭代：

后向： $k = K, \cdots, 1$

$$L_{k-1}^{\mathrm{T}} \hat{x}_{k-1} = -L_{k,k-1}^{\mathrm{T}} \hat{x}_k + d_{k-1}$$

于是就得到了递归的算法流程

这个过程等价于传统的Rauch-Tung-Striebel平滑算法，五个前向的迭代等价于著名的卡尔曼滤波器

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- Rauch-Tung-Striebel平滑算法（RTS Smoother）
  - RTS Smoother是Herber Rauch, Frank Tung, Charlotte Striebel共同提出的

- 先看前向迭代的五个式子

前向：$k = 1, \cdots, K$

$$
\begin{aligned}
L_{k-1}L_{k-1}^{\mathrm{T}} &= I_{k-1} + A_{k-1}^{\mathrm{T}}Q_k^{-1}A_{k-1} \\
L_{k-1}d_{k-1} &= q_{k-1} - A_{k-1}^{\mathrm{T}}Q_k^{-1}v_k \\
L_{k,k-1}L_{k-1}^{\mathrm{T}} &= -Q_k^{-1}A_{k-1} \\
I_k &= -L_{k,k-1}L_{k,k-1}^{\mathrm{T}} + Q_k^{-1} + C_k^{\mathrm{T}}R_k^{-1}C_k \\
q_k &= -L_{k,k-1}d_{k-1} + Q_k^{-1}v_k + C_k^{\mathrm{T}}R_k^{-1}y_k
\end{aligned}
$$

- □ 由第3式解出L(k,k-1)：$L_{k,k-1} = -Q_k^{-1}A_{k-1}L_{k-1}^{-T}$
- □ 代入第4式： （第2行代入第1式）

$$
\begin{aligned}
I_k &= -\left(-Q_k^{-1}A_{k-1}L_{k-1}^{-T}\right)\left(-L_{k-1}^{-1}A_{k-1}^{T}Q_k^{-T}\right) + Q_k^{-1} + C_k^{T}R_k^{-1}C_k \\
&= -\left(Q_k^{-1}A_{k-1}\left(L_{k-1}L_{k-1}^{T}\right)^{-1}A_{k-1}^{T}Q_k^{-T}\right) + Q_k^{-1} + C_k^{T}R_k^{-1}C_k \\
&= -\left(Q_k^{-1}A_{k-1}\left(I_{k-1} + A_{k-1}^{T}Q_k^{-1}A_{k-1}\right)A_{k-1}^{T}Q_k^{-T}\right) + Q_k^{-1} + C_k^{T}R_k^{-1}C_k
\end{aligned}
$$

- □ 再使用SMW化简：

$$
I_k = \underbrace{Q_k^{-1} - Q_k^{-1}A_{k-1}\left(I_{k-1} + A_{k-1}^{\mathrm{T}}Q_k^{-1}A_{k-1}\right)^{-1}A_{k-1}^{\mathrm{T}}Q_k^{-1}}_{\text{由式（2.75）：}\left(A_{k-1}I_{k-1}^{-1}A_{k-1}^{\mathrm{T}}+Q_k\right)^{-1}} + C_k^{\mathrm{T}}R_k^{-1}C_k
$$

- □ SMW(2)式：$(D + CAB)^{-1} \equiv D^{-1} - D^{-1}C\left(A^{-1} + BD^{-1}C\right)^{-1}BD^{-1}$

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

这一页几个重要结论都已经用矩形框了出来

- 于是： $I_k = \left(A_{k-1} I_{k-1}^{-1} A_{k-1}^T + Q_k\right)^{-1} + C_k^T R_k^{-1} C_k$
- 设： $\hat{P}_{k,f} = I_k^{-1}$ 那么上式可写为两步：

$$\boxed{\begin{aligned}\check{P}_{k,f} &= A_{k-1}\hat{P}_{k-1,f}A_{k-1}^{\mathrm{T}} + Q_k \\ \hat{P}_{k,f}^{-1} &= \check{P}_{k,f}^{-1} + C_k^{\mathrm{T}} R_k^{-1} C_k\end{aligned}}$$

- ◻ 先验的
- ◻ 后验的，或者修正的

公式分为三部分，必须记，必须通过白话来记：
1、不带卡尔曼增益的协方差传递律
2、卡尔曼增益三公式
3、带卡尔曼增益的协方差传递律

1、先验的协方差等于上一时刻后验的协方差传递后加上控制噪声协方差（协方差传递）
后验的信息等于先验的信息加上观测噪声的信息传递（信息传递）
2、卡尔曼增益的基础定义式：
K = 后验协方差 × 观测矩阵T × 观测噪声逆
通过代入，得到第二个，通过SWM得到第三个（乾坤大挪移）
3、先验协方差，左乘(I-KC)，得到后验协方差
不对，应该这么说，左乘单位阵减去卡尔曼增益乘以观测矩阵

- 再把它写成经典卡尔曼滤波形式，定义： $\boxed{K_k = \hat{P}_{k,f} C_k^{\mathrm{T}} R_k^{-1}}$
- 那么上面第2式写为： $\boxed{K_k = \left(\check{P}_{k,f}^{-1} + C_k^{\mathrm{T}} R_k^{-1} C_k\right)^{-1} C_k^{\mathrm{T}} R_k^{-1} = \check{P}_{k,f} C_k^{\mathrm{T}} \left(C_k \check{P}_{k,f} C_k^{\mathrm{T}} + R_k\right)^{-1}}$
  - 此式可用于计算K

SMW(4)： $(D + CAB)^{-1} CA \equiv D^{-1} C \left(A^{-1} + BD^{-1} C\right)^{-1}$

- K已经算得，那么代入上面2式，即有：

$$\check{P}_{k,f}^{-1} = \hat{P}_{k,f}^{-1} - C_k^{\mathrm{T}} R_k^{-1} C_k = \hat{P}_{k,f}^{-1}\left(1 - \underbrace{\hat{P}_{k,f} C_k^{\mathrm{T}} R_k^{-1}}_{K_k} C_k\right) = \hat{P}_{k,f}^{-1}\left(1 - K_k C_k\right)$$

即： $\boxed{\hat{P}_{k,f} = \left(1 - K_k C_k\right)\check{P}_{k,f}}$

- ◻ K为卡尔曼增益，上式即为协方差更新

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 回到前向迭代：

前向：$k = 1, \cdots, K$

$$
\begin{aligned}
\boldsymbol{L}_{k-1}\boldsymbol{L}_{k-1}^{\mathrm{T}} &= \boldsymbol{I}_{k-1} + \boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1} \\
\boldsymbol{L}_{k-1}\boldsymbol{d}_{k-1} &= \boldsymbol{q}_{k-1} - \boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{v}_k \\
\boldsymbol{L}_{k,k-1}\boldsymbol{L}_{k-1}^{\mathrm{T}} &= -\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1} \\
\boldsymbol{I}_k &= -\boldsymbol{L}_{k,k-1}\boldsymbol{L}_{k,k-1}^{\mathrm{T}} + \boldsymbol{Q}_k^{-1} + \boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{C}_k \\
\boldsymbol{q}_k &= -\boldsymbol{L}_{k,k-1}\boldsymbol{d}_{k-1} + \boldsymbol{Q}_k^{-1}\boldsymbol{v}_k + \boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{y}_k
\end{aligned}
$$

- 由2式： $d_{k-1} = L_{k-1}^{-1}\left(q_{k-1} - A_{k-1}^T Q_k^{-1} v_k\right)$

- 由3式： $L_{k,k-1} = -Q_k^{-1} A_{k-1} L_{k-1}^{-T}$

那么： $\boldsymbol{L}_{k,k-1}\boldsymbol{d}_{k-1} = -\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\boldsymbol{L}_{k-1}\boldsymbol{L}_{k-1}^{\mathrm{T}}\right)^{-1}\left(\boldsymbol{q}_{k-1} - \boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{v}_k\right)$

又可以代入1式：

$$
\begin{aligned}
\boldsymbol{q}_k = {} & \underbrace{\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\boldsymbol{I}_{k-1} + \boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\right)^{-1}}_{\text{由式（2.75）：}\left(\boldsymbol{A}_{k-1}\boldsymbol{I}_{k-1}^{-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}+\boldsymbol{Q}_k\right)^{-1}\boldsymbol{A}_{k-1}\boldsymbol{I}_{k-1}^{-1}} \boldsymbol{q}_{k-1} \\
& + \underbrace{\left(\boldsymbol{Q}_k^{-1} - \boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\boldsymbol{I}_{k-1} + \boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\right)^{-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\right)}_{\text{由式（2.75）：}\left(\boldsymbol{A}_{k-1}\boldsymbol{I}_{k-1}^{-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}+\boldsymbol{Q}_k\right)^{-1}}\boldsymbol{v}_k \\
& + \boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{y}_k
\end{aligned}
$$

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 在qk的表达式中代入前面的定义：

$$\boxed{\hat{P}_{k,f} = I_k^{-1}}$$

$$\boxed{\begin{aligned}\check{P}_{k,f} &= A_{k-1}\hat{P}_{k-1,f}A_{k-1}^{\mathrm{T}} + Q_k \\ \hat{P}_{k,f}^{-1} &= \check{P}_{k,f}^{-1} + C_k^{\mathrm{T}}R_k^{-1}C_k\end{aligned}}$$

$$\begin{aligned}q_k = {} & \underbrace{Q_k^{-1}A_{k-1}\left(I_{k-1} + A_{k-1}^{\mathrm{T}}Q_k^{-1}A_{k-1}\right)^{-1}}_{\text{由式（2.75）: } \left(A_{k-1}I_{k-1}^{-1}A_{k-1}^{\mathrm{T}}+Q_k\right)^{-1}A_{k-1}I_{k-1}^{-1}} q_{k-1} \\ & + \underbrace{\left(Q_k^{-1} - Q_k^{-1}A_{k-1}\left(I_{k-1} + A_{k-1}^{\mathrm{T}}Q_k^{-1}A_{k-1}\right)^{-1}A_{k-1}^{\mathrm{T}}Q_k^{-1}\right)}_{\text{由式（2.75）: } \left(A_{k-1}I_{k-1}^{-1}A_{k-1}^{\mathrm{T}}+Q_k\right)^{-1}} v_k \\ & + C_k^{\mathrm{T}}R_k^{-1}y_k\end{aligned}$$

- 且令： $\hat{P}_{k,f}^{-1}\hat{x}_{k,f} = q_k$

$$q_k = \check{P}_{k,f}^{-1} A_{k-1} \hat{P}_{k-1,f} q_{k-1} + \check{P}_{k,f}^{-1} v_k + C_k^{T} R_k^{-1} y_k$$

$$\hat{P}_{k,f}^{-1} \hat{x}_{k,f} = \check{P}_{k,f}^{-1} A_{k-1} \hat{x}_{k-1,f} + \check{P}_{k,f}^{-1} v_k + C_k^{T} R_k^{-1} y_k$$

$$= \check{P}_{k,f}^{-1} (\underbrace{A_{k-1} \hat{x}_{k-1,f} + v_k}_{\check{x}_{k,f}}) + C_k^{T} R_k^{-1} y_k.$$

- 于是得：

$$\begin{aligned}\check{x}_{k,f} &= A_{k-1}\hat{x}_{k-1,f} + v_k \\ \hat{P}_{k,f}^{-1}\hat{x}_{k,f} &= \check{P}_{k,f}^{-1}\check{x}_{k,f} + C_k^{\mathrm{T}}R_k^{-1}y_k\end{aligned}$$

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 处理一下前面协方差逆的形式：$\hat{P}_{k,f}^{-1}\hat{x}_{k,f}=\check{P}_{k,f}^{-1}\check{x}_{k,f}+C_k^{\mathrm{T}}R_k^{-1}y_k$
- 两侧左乘后验协方差矩阵：$\hat{x}_{k,f}=\underbrace{\hat{P}_{k,f}\check{P}_{k,f}^{-1}}_{1-K_kC_k}\check{x}_{k,f}+\underbrace{\hat{P}_{k,f}C_k^{\mathrm{T}}R_k^{-1}}_{K_k}y_k$
- 于是得到均值更新方程：$\hat{x}_{k,f}=\check{x}_{k,f}+K_k\left(y_k-C_k\check{x}_{k,f}\right)$

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 最后来处理后向迭代

前向：$k = 1, \cdots, K$

$$\boldsymbol{L}_{k-1}\boldsymbol{L}_{k-1}^{\mathrm{T}} = \boldsymbol{I}_{k-1} + \boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}$$

$$\boldsymbol{L}_{k-1}\boldsymbol{d}_{k-1} = \boldsymbol{q}_{k-1} - \boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{v}_k$$

$$\boldsymbol{L}_{k,k-1}\boldsymbol{L}_{k-1}^{\mathrm{T}} = -\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}$$

$$\boldsymbol{I}_k = -\boldsymbol{L}_{k,k-1}\boldsymbol{L}_{k,k-1}^{\mathrm{T}} + \boldsymbol{Q}_k^{-1} + \boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{C}_k$$

$$\boldsymbol{q}_k = -\boldsymbol{L}_{k,k-1}\boldsymbol{d}_{k-1} + \boldsymbol{Q}_k^{-1}\boldsymbol{v}_k + \boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{y}_k$$

后向：$k = K, \cdots, 1$

$$\boldsymbol{L}_{k-1}^{\mathrm{T}}\hat{\boldsymbol{x}}_{k-1} = -\boldsymbol{L}_{k,k-1}^{\mathrm{T}}\hat{\boldsymbol{x}}_k + \boldsymbol{d}_{k-1}$$

后向式两侧乘L(k-1)，得：

$$\hat{\boldsymbol{x}}_{k-1} = \left(\boldsymbol{L}_{k-1}\boldsymbol{L}_{k-1}^{\mathrm{T}}\right)^{-1}\boldsymbol{L}_{k-1}\left(-\boldsymbol{L}_{k,k-1}^{\mathrm{T}}\hat{\boldsymbol{x}}_k + \boldsymbol{d}_{k-1}\right)$$

代入前三式：

$$\begin{aligned}
\hat{x}_{k-1} &= (I_{k-1} + A_{k-1}^{T} Q_k^{-1} A_{k-1}) \cdot \left(-(L_{k,k-1} \cdot L_{k-1}^{T})^{T}\hat{x}_k + L_{k-1} d_{k-1}\right) \\
&= (I_{k-1} + A_{k-1}^{T} Q_k^{-1} A_{k-1}) \left(A_{k-1}^{T} Q_k^{-T}\hat{x}_k + q_{k-1} - A_{k-1}^{T} Q_k^{-1} v_k\right) \\
&= (I_{k-1} + A_{k-1}^{T} Q_k^{-1} A_{k-1}) \cdot A_{k-1}^{T} Q_k^{-1} (\hat{x}_k - v_k) \\
&\quad + (I_{k-1} + A_k^{T} Q_k^{-1} A_{k-1}) q_{k-1}
\end{aligned}$$

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 再由SMW公式：

$$\hat{x}_{k-1} = \underbrace{\left(I_{k-1} + A_{k-1}^{T} Q_k^{-1} A_{k-1}\right)^{-1} A_{k-1}^{T} Q_k^{-1}}_{\text{由式（2.75）: } I_{k-1}^{-1} A_{k-1}^{T}\left(A_{k-1} I_{k-1}^{-1} A_{k-1}^{T} + Q_k\right)^{-1}} \left(\hat{x}_k - v_k\right) + \underbrace{\left(I_{k-1} + A_{k-1}^{T} Q_k^{-1} A_{k-1}\right)^{-1}}_{\text{由式（2.75）: } I_{k-1}^{-1} - I_{k-1}^{-1} A_{k-1}^{T}\left(A_{k-1} I_{k-1}^{-1} A_{k-1}^{T} + Q_k\right)^{-1} A_{k-1} I_{k-1}^{-1}} q_{k-1}$$

- 且有：

$$\boxed{\hat{P}_{k,f} = I_k^{-1}}$$

$$\boxed{\begin{aligned} \check{P}_{k,f} &= A_{k-1} \hat{P}_{k-1,f} A_{k-1}^{T} + Q_k \\ \hat{P}_{k,f}^{-1} &= \check{P}_{k,f}^{-1} + C_k^{T} R_k^{-1} C_k \end{aligned}}$$

$$\boxed{\hat{P}_{k,f}^{-1} \hat{x}_{k,f} = q_k}$$

$$\boxed{\begin{aligned} \check{x}_{k,f} &= A_{k-1} \hat{x}_{k-1,f} + v_k \\ \hat{P}_{k,f}^{-1} \hat{x}_{k,f} &= \check{P}_{k,f}^{-1} \check{x}_{k,f} + C_k^{T} R_k^{-1} y_k \end{aligned}}$$

$$\begin{aligned} \hat{x}_{k-1} &= \hat{P}_{k-1,f} A_{k-1}^{T} \check{P}_{k,f} (\hat{x}_k - v_k) + (\hat{P}_{k-1,f} - \hat{P}_{k-1,f} A_{k-1}^{T} \check{P}_{k,f} A_{k-1} \hat{P}_{k-1,f}) \cdot \hat{P}_{k-1,f}^{-1} \hat{x}_{k-1,f} \\ &= \hat{x}_{k-1,f} + \hat{P}_{k-1,f} A_{k-1}^{T} \check{P}_{k,f} (\hat{x}_k - \underbrace{v_k - A_{k-1} \hat{x}_{k-1,f}}_{\check{x}_{k,f} = A_{k-1} \hat{x}_{k-1,f} + v_k}) \\ &= \hat{x}_{k-1,f} + \hat{P}_{k-1} A_{k-1}^{T} \check{P}_{k,f} (\hat{x}_k - \check{x}_{k,f}) \end{aligned}$$

深蓝学院
shenlanxueyuan.com

离散时间的递归平滑算法

- 综合上面的式子，我们得到经典的RTS Smoother：

前向：$k = 1, \cdots, K$

$$\check{P}_{k,f} = A_{k-1}\hat{P}_{k-1,f}A_{k-1}^{\mathrm{T}} + Q_k$$

$$\check{x}_{k,f} = A_{k-1}\hat{x}_{k-1,f} + v_k$$

$$K_k = \check{P}_{k,f}C_k^{\mathrm{T}}\left(C_k\check{P}_{k,f}C_k^{\mathrm{T}} + R_k\right)^{-1}$$

$$\hat{P}_{k,f} = (1 - K_kC_k)\check{P}_{k,f}$$

$$\hat{x}_{k,f} = \check{x}_{k,f} + K_k(y_k - C_k\check{x}_{k,f})$$

后向：$k = K, \cdots, 1$

$$\hat{x}_{k-1} = \hat{x}_{k-1,f} + \hat{P}_{k-1,f}A_{k-1}^{\mathrm{T}}\check{P}_{k,f}^{-1}(\hat{x}_k - \check{x}_{k,f})$$

With 初始值：

$$\check{P}_{0,f} = \check{P}_0$$

$$\check{x}_{0,f} = \check{x}_0$$

$$\hat{x}_K = \hat{x}_{K,f}$$

- 前五个式子就是经典的卡尔曼滤波器
- 整个过程没有任何的近似
- 和批量解法本质上是一样的（核心即Cholesky分解解方程）
- 不过，卡尔曼滤波器也可以从很多其他的角度推出，此处的推导是从Cholesky解方程出发的，而卡尔曼滤波不必跟Cholesky分解挂钩

深蓝学院
shenlanxueyuan.com

第2讲 线性高斯系统的状态估计问题

- 离散时间的批量估计
- 离散时间的递归平滑算法
- 离散时间的滤波算法
- 连续时间的批量估计

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- RTS Smoother是无法在线运行的（非因果的Not causal）
  - 它的后向迭代过程使用下个时刻的信息更新之前的估计
  - 初始值中需要知道x(K)的后验
- 如果希望算法实时运行，就要去掉后向迭代的过程

平滑算法利用了所有能用到的数据来估计状态

$$\overbrace{\underbrace{\check{x}_0, y_0, v_1, y_1, v_2, y_2, \ldots, v_{k-1}, y_{k-1}, v_k, y_k}_{\hat{x}_{k,f}}, v_{k+1}, y_{k+1}, \ldots, v_K, y_K}^{\hat{x}_k}$$

滤波器仅利用了过去/现在的数据来估计状态

- 接下来我们来看滤波器的推导

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

通过MAP估计和贝叶斯推断都能很快得到卡尔曼滤波。具体不展开了

- 通过MAP推导卡尔曼滤波
- 因为只有前向过程，所以把下标中的f去掉
- 假设已有k-1时刻的状态均值和方差：$\{\hat{x}_{k-1}, \hat{P}_{k-1}\}$，目标是估计k时刻的状态：$\{\hat{x}_k, \hat{P}_k\}$
- 这其中需要用到k时刻的输入和观测，即 $\{\hat{x}_{k-1}, \hat{P}_{k-1}, v_k, y_k\} \mapsto \{\hat{x}_k, \hat{P}_k\}$
- 定义：

$$z = \begin{bmatrix} \hat{x}_{k-1} \\ v_k \\ y_k \end{bmatrix}, \quad H = \begin{bmatrix} 1 & \\ -A_{k-1} & 1 \\ & C_k \end{bmatrix}, \quad W = \begin{bmatrix} \hat{P}_{k-1} & & \\ & Q_k & \\ & & R_k \end{bmatrix} \qquad \hat{x} = \begin{bmatrix} \hat{x}'_{k-1} \\ \hat{x}_k \end{bmatrix}$$

- 那么MAP的线性方程为：$(H^T W^{-1} H)\hat{x} = H^T W^{-1} z$

- 带撇表示用k时刻数据推出k-1时刻状态
- 最小二乘约束解释：既要符合先验，又要符合当前的运动和观测

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 代入MAP最优解：

$$\begin{bmatrix} \hat{P}_{k-1}^{-1} + A_{k-1}^T Q_k^{-1} A_{k-1} & -A_{k-1}^T Q_k^{-1} \\ -Q_k^{-1} A_{k-1} & Q_k^{-1} + C_k^T R_k^{-1} C_k \end{bmatrix} \begin{bmatrix} \hat{x}'_{k-1} \\ \hat{x}_k \end{bmatrix} = \begin{bmatrix} \hat{P}_{k-1}^{-1} \hat{x}_{k-1} - A_{k-1}^T Q_k^{-1} v_k \\ Q_k^{-1} v_k + C_k^T R_k^{-1} y_k \end{bmatrix} \tag{3.110}$$

- 但是只关心x(k)，所以对上式进行消元，把系数矩阵第2行第1列消去：

$$\begin{bmatrix} \hat{P}_{k-1}^{-1} + A_{k-1}^T Q_k^{-1} A_{k-1} & -A_{k-1}^T Q_k^{-1} \\ 0 & \begin{matrix} Q_k^{-1} - Q_k^{-1} A_{k-1} \left(\hat{P}_{k-1}^{-1} + A_{k-1}^T Q_k^{-1} A_{k-1} \right)^{-1} \\ \times A_{k-1}^T Q_k^{-1} + C_k^T R_k^{-1} C_k \end{matrix} \end{bmatrix} \begin{bmatrix} \hat{x}'_{k-1} \\ \hat{x}_k \end{bmatrix}$$

$$= \begin{bmatrix} \hat{P}_{k-1}^{-1} \hat{x}_{k-1} - A_{k-1}^T Q_k^{-1} v_k \\ Q_k^{-1} A_{k-1} \left(\hat{P}_{k-1}^{-1} + A_{k-1}^T Q_k^{-1} A_{k-1} \right)^{-1} \left(\hat{P}_{k-1}^{-1} \hat{x}_{k-1} - A_{k-1}^T Q_k^{-1} v_k \right) \\ + Q_k^{-1} v_k + C_k^T R_k^{-1} y_k \end{bmatrix} \tag{3.112}$$

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 于是最优解的线性方程为：

$$\left(\underbrace{\boldsymbol{Q}_k^{-1}-\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\hat{\boldsymbol{P}}_{k-1}^{-1}+\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\right)^{-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}}_{\text{由式（2.75）：}\left(\boldsymbol{Q}_k+\boldsymbol{A}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}\right)^{-1}}+\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{C}_k\right)\hat{\boldsymbol{x}}_k$$

$$=\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\hat{\boldsymbol{P}}_{k-1}^{-1}+\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\right)^{-1}\left(\hat{\boldsymbol{P}}_{k-1}^{-1}\hat{\boldsymbol{x}}_{k-1}-\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{v}_k\right)$$

$$+\boldsymbol{Q}_k^{-1}\boldsymbol{v}_k+\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{y}_k$$

- 定义先验和后验协方差：

$$\check{\boldsymbol{P}}_k=\boldsymbol{Q}_k+\boldsymbol{A}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}$$

$$\hat{\boldsymbol{P}}_k=\left(\check{\boldsymbol{P}}_k^{-1}+\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{C}_k\right)^{-1}$$

- 那么：

$$\begin{aligned}\hat{\boldsymbol{P}}_k^{-1}\hat{\boldsymbol{x}}_k&=\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\hat{\boldsymbol{P}}_{k-1}^{-1}+\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\right)^{-1}\\&\quad\times\left(\hat{\boldsymbol{P}}_{k-1}^{-1}\hat{\boldsymbol{x}}_{k-1}-\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{v}_k\right)+\boldsymbol{Q}_k^{-1}\boldsymbol{v}_k+\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{y}_k\\&=\underbrace{\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\hat{\boldsymbol{P}}_{k-1}^{-1}+\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\right)^{-1}\hat{\boldsymbol{P}}_{k-1}^{-1}}_{\text{由下文：}\check{\boldsymbol{P}}_k^{-1}\boldsymbol{A}_{k-1}}\hat{\boldsymbol{x}}_{k-1}\\&\quad+\underbrace{\left(\boldsymbol{Q}_k^{-1}-\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\hat{\boldsymbol{P}}_{k-1}^{-1}+\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\right)^{-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\right)}_{\check{\boldsymbol{P}}_k^{-1}}\boldsymbol{v}_k+\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{y}_k\\&=\check{\boldsymbol{P}}_k^{-1}\underbrace{\left(\boldsymbol{A}_{k-1}\hat{\boldsymbol{x}}_{k-1}+\boldsymbol{v}_k\right)}_{\check{\boldsymbol{x}}_k}+\boldsymbol{C}_k^{\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{y}_k\end{aligned}$$

$$\begin{aligned}&\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\underbrace{\left(\hat{\boldsymbol{P}}_{k-1}^{-1}+\boldsymbol{A}_{k-1}^{\mathrm{T}}\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\right)^{-1}}_{\text{再次由式（2.75）}}\hat{\boldsymbol{P}}_{k-1}^{-1}\\&=\boldsymbol{Q}_k^{-1}\boldsymbol{A}_{k-1}\left(\hat{\boldsymbol{P}}_{k-1}-\hat{\boldsymbol{P}}_{k-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}\underbrace{\left(\boldsymbol{Q}_k+\boldsymbol{A}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}\right)^{-1}}_{\check{\boldsymbol{P}}_k^{-1}}\right.\\&\qquad\left.\times\boldsymbol{A}_{k-1}\hat{\boldsymbol{P}}_{k-1}\right)\hat{\boldsymbol{P}}_{k-1}^{-1}\\&=\left(\boldsymbol{Q}_k^{-1}-\boldsymbol{Q}_k^{-1}\underbrace{\boldsymbol{A}_{k-1}\hat{\boldsymbol{P}}_{k-1}\boldsymbol{A}_{k-1}^{\mathrm{T}}}_{\check{\boldsymbol{P}}_k-\boldsymbol{Q}_k}\check{\boldsymbol{P}}_k^{-1}\right)\boldsymbol{A}_{k-1}\\&=\left(\boldsymbol{Q}_k^{-1}-\boldsymbol{Q}_k^{-1}+\check{\boldsymbol{P}}_k^{-1}\right)\boldsymbol{A}_{k-1}\\&=\check{\boldsymbol{P}}_k^{-1}\boldsymbol{A}_{k-1}\end{aligned}$$

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 上式即为信息形式的卡尔曼滤波器：

预测：
$$\check{P}_k = A_{k-1}\hat{P}_{k-1}A_{k-1}^{\mathrm{T}} + Q_k$$
$$\check{x}_k = A_{k-1}\hat{x}_{k-1} + v_k$$

更新：
$$\hat{P}_k^{-1} = \check{P}_k^{-1} + C_k^{\mathrm{T}}R_k^{-1}C_k$$
$$\hat{P}_k^{-1}\hat{x}_k = \check{P}_k^{-1}\check{x}_k + C_k^{\mathrm{T}}R_k^{-1}y_k$$

- 定义卡尔曼增益：$K_k = \hat{P}_k C_k^{\mathrm{T}} R_k^{-1}$
  - 我们已经推过它可以写为：$K_k = \check{P}_k C_k^{\mathrm{T}}\left(C_k\check{P}_k C_k^{\mathrm{T}} + R_k\right)^{-1}$

返回查看

- 于是把上面式子整理为：

预测：
$$\check{P}_k = A_{k-1}\hat{P}_{k-1}A_{k-1}^{\mathrm{T}} + Q_k$$
$$\check{x}_k = A_{k-1}\hat{x}_{k-1} + v_k$$

卡尔曼增益：
$$K_k = \check{P}_k C_k^{\mathrm{T}}\left(C_k\check{P}_k C_k^{\mathrm{T}} + R_k\right)^{-1}$$

更新：
$$\hat{P}_k = \left(1 - K_k C_k\right)\check{P}_k$$
$$\hat{x}_k = \check{x}_k + K_k \underbrace{\left(y_k - C_k\check{x}_k\right)}_{\text{更新量}}$$

- 这就是经典的卡尔曼滤波器
- 更新量可看成一个修正量，而卡尔曼增益就是它的权重系数
- 它等价于RTS的前向过程，或等价于MAP估计
- 实际上也可以不写卡尔曼增益，直接用上方式子推出

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 通过贝叶斯推断同样可以推出卡尔曼滤波器，过程更加简洁，下面推导之
- k-1时刻的后验分布 $p\left(x_{k-1} | \check{x}_0, v_{1:k-1}, y_{0:k-1}\right) = \mathcal{N}\left(\hat{x}_{k-1}, \hat{P}_{k-1}\right)$
- K时刻的先验由运动方程给出： $p\left(x_k | \check{x}_0, v_{1:k}, y_{0:k-1}\right) = \mathcal{N}\left(\check{x}_k, \check{P}_k\right)$
- 显然这是高斯分布的线性变换，于是有：

$$
\begin{aligned}
\check{P}_k &= A_{k-1} \hat{P}_{k-1} A_{k-1}^{\mathrm{T}} + Q_k \\
\check{x}_k &= A_{k-1} \hat{x}_{k-1} + v_k
\end{aligned}
$$

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 对于更新部分，使用高斯推断，联合分布为：

$$
\begin{aligned}
p\left(\boldsymbol{x}_k, \boldsymbol{y}_k \mid \check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:k}, \boldsymbol{y}_{0:k-1}\right) &= \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}\right) \\
&= \mathcal{N}\left(\begin{bmatrix} \check{\boldsymbol{x}}_k \\ \boldsymbol{C}_k \check{\boldsymbol{x}}_k \end{bmatrix}, \begin{bmatrix} \check{\boldsymbol{P}}_k & \check{\boldsymbol{P}}_k \boldsymbol{C}_k^{\mathrm{T}} \\ \boldsymbol{C}_k \check{\boldsymbol{P}}_k & \boldsymbol{C}_k \check{\boldsymbol{P}}_k \boldsymbol{C}_k^{\mathrm{T}} + \boldsymbol{R}_k \end{bmatrix}\right)
\end{aligned}
$$

- 那么条件分布为： $p\left(\boldsymbol{x}_k \mid \check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:k}, \boldsymbol{y}_{0:k}\right) = \mathcal{N}\Big(\underbrace{\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\left(\boldsymbol{y}_k - \boldsymbol{\mu}_y\right)}_{\hat{\boldsymbol{x}}_k}, \underbrace{\boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}}_{\hat{\boldsymbol{P}}_k}\Big)$

- 在结果中代入卡尔曼增益，易得：

$$
\begin{aligned}
\boldsymbol{K}_k &= \check{\boldsymbol{P}}_k \boldsymbol{C}_k^{\mathrm{T}}\left(\boldsymbol{C}_k \check{\boldsymbol{P}}_k \boldsymbol{C}_k^{\mathrm{T}} + \boldsymbol{R}_k\right)^{-1} \\
\hat{\boldsymbol{P}}_k &= \left(\mathbf{1} - \boldsymbol{K}_k \boldsymbol{C}_k\right) \check{\boldsymbol{P}}_k \\
\hat{\boldsymbol{x}}_k &= \check{\boldsymbol{x}}_k + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{C}_k \check{\boldsymbol{x}}_k\right)
\end{aligned}
$$

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 在滤波器问题中，我们看到了和批量问题中相同的结论：
  - MAP和贝叶斯推断给出了同样的结果，等价于卡尔曼滤波器
  - 原因依然是高斯分布的均值和模处在同一点上

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 关于卡尔曼滤波器的结论
  - 卡尔曼滤波器给出了LG系统下$\boxed{\text{最优线性无偏估计}}$（Best Linear Unbiased Estimate, BLUE）
  - 需要有初始状态
  - 卡尔曼滤波器即RTS Smoother的前向部分
  - 在非线性场合下，我们会使用扩展卡尔曼滤波器（EKF），但此时MAP、贝叶斯推断、EKF给出的结果均会不一样

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 本次课程略去的内容：
  - 3.4 连续时间批量估计问题
  - 3.3.4 从增益最优化角度来看卡尔曼滤波
  - 3.3.6 误差动态过程
  - 3.3.7 存在性、唯一性和能观性

深蓝学院
shenlanxueyuan.com

离散时间的滤波算法

- 习题：课本第1,2,6题