Linear-Gaussian Estimation

Words	Translate
recursive	递归的
deterministic	决定性的
sparse	稀疏的
exploit	利用

3.1 Batch Discrete-Time Estimation

译作：离散时间的批量估计

3.1.1 Problem Setup

这章节的大部分内容，我们考虑**离散时间、线性、时变的方程**。我们定义如下的运动以及观测模型：

$$motion\ model: x_k = A_{k-1}x_{k-1} + v_k + w_k,\ \ k = 1...K$$
$$observation\ model: y_k = C_k x_k + n_k,\ k = 0 ... K$$

其中，k是离散时间下标，最大值为K。上述符号意义如下：

$$\begin{aligned} \text{system state}: & \ \mathbf{x}_k \in \mathbb{R}^N \\ \text{initial state}: & \ \mathbf{x}_0 \in \mathbb{R}^N \sim \mathcal{N}\left(\check{\mathbf{x}}_0, \check{\mathbf{P}}_0\right) \\ \text{input}: & \ \mathbf{v}_k \in \mathbb{R}^N \\ \text{process noise}: & \ \mathbf{w}_k \in \mathbb{R}^N \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_k\right) \\ \text{measurement}: & \ \mathbf{y}_k \in \mathbb{R}^M \\ \text{measurement noise}: & \ \mathbf{n}_k \in \mathbb{R}^M \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_k\right) \end{aligned}$$

除了vk以外，这些都是随机变量。**噪声和初始状态我们均假设各不相关，而且不同时刻也不相关**。矩阵Ak称为**转移矩阵(transition matrix)**，矩阵Ck∈R^(M×N)被称为**观测矩阵(observation matrix)。**

状态估计目的

我们的首要目的：估计(得到)系统的所有时间的所有状态。

我们的已知：

1. 初始状态x0h以及其协方差矩阵P0h；我们有时候不知道初始信息，就需要在不知道的情况下进行推导。
2. 输入vk，来自我们的控制器输出，已知，而且其噪声协方差矩阵Qk。
3. 测量值yk,meas，是yk的一次**实现**，噪声协方差：Rk

状态估计问题

状态估计问题是指：在k个(一个或多个)时间点上，基于初始的状态信息x0h、一系列观测数据y(0:K,meas)、一系列输入v(1:K)，以及系统的运动和观测模型，来计算系统的真实状态的估计值x(k)h。

我们回到LG(Linear Gaussian)系统的估计问题：我们用两个不同的途径解决此问题：

1. **贝叶斯推断(Bayesian inference)：**我们从状态的先验概率密度函数出发，通过初始状态、输入、运动方程和观测数据，计算状态的后验概率密度函数。
2. **最大后验估计(Maximum A Posteriori MAP)**：我们用优化理论，来寻找给定信息下(初始状态、输入、观测)的最大后验估计。

3.1.2 Maximum A Posteriori

译作：最大后验估计

在批量估计问题中，我们的目标是解决如下MAP问题：

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

我们希望在给定先验信息和所有时刻的运动v、观测y的情况下，推测出所有时刻的最优状态xhat。

为此，我们定义一些数据：**我们用上帽子表示后验估计(包含了观测量的)，用下帽子表示先验估计(不含观测的)**

$$x = x_{0:K} = (x_0, \ldots, x_K)$$
$$v = (\check{x}_0, v_1, \ldots, v_K)$$
$$y = y_{0:K} = (y_0, \ldots, y_K)$$

我们首先利用贝叶斯公式重写MAP：过程详解

$$\hat{x} = argmax_x p(x|v,y) = argmax_x \frac{p(y|x,v)p(x|v)}{p(y|v)} = argmax_x p(y|x)p(x|v)$$

$$\begin{aligned} p(x|v,y) &= (\text{贝叶斯公式})\frac{p(v,y|x)p(x)}{p(v,y)} \\ &= (\text{条件概率公式})\frac{p(v,y|x)p(x)}{p(y|v)p(v)} \\ &= (\text{条件概率公式})\frac{p(v,y,x)}{p(y|v)p(v)} \\ &= (\text{条件概率公式})\frac{p(y|x,v)p(x,v)}{p(y|v)\frac{p(x,v)}{p(x|v)}} \\ (\text{化简}) &= \frac{p(y|x,v)p(x|v)}{p(y|v)} \end{aligned}$$

最后一个等式的解释：分母p(y|v)与变量x无关，可以当作常数；分子中，p(y|x,v)的v可以省略，因为如果x已知，v不会影响观测数据(观测方程与其无关)。

接下来，**我们做出一个重要假设：噪声wk和nk之间无关。**这样我们有：

$$p(y|x) = \prod_{k=0}^{K} p(y_k|x_k)(\text{根据观测方程可以得到})$$

另一方面，贝叶斯定理允许我们将p(x|v)分解为：

$$p(x|v) = p(x_0|\check{x_0})\prod_{k=1}^{K} p(x_k|x_{k-1}, v_k)$$
$$(\text{根据输入方程可得})$$

线性系统中，高斯密度函数可展开为：

$$p(x_0|\check{x_0}) = \frac{1}{\sqrt{(2\pi)^N det(\check{P}_0)}} \times exp(-\frac{1}{2}(x_0 - \check{x_0})\check{P_0}^{-1}(x_0 - \check{x_0}))$$

$$p(x_k|x_{k-1}, v_k) = (\text{根据运动方程})\frac{1}{\sqrt{}}$$

$$p(y_k|x_k) = (\text{根据观测方程})\frac{1}{\sqrt{}}$$

我们对优化的式子取对数(由于对数函数是**单调递增的(monotonically increasing)**，因此对我们的优化问题没有影响)。

$$ln(p(y|x)p(x|v)) = ln(p(x_0|\check{x_0})) + \sum_{k=1}^{K} p(x_k|x_{k-1}, v_k) + \sum_{k=0}^{K} p(y_k|x_k)$$

将上式代入，并且拿掉与x无关的常数项，我们定义：

$$J_{v,k}(x)=\begin{cases}\frac{1}{2}(x_0-\check{x}_0)^T\check{P}_0^{-1}(x_0-\check{x}_0) & , \quad k=0\\ \frac{1}{2}(x_k-A_{k-1}x_{k-1}-v_k)^TQ_k^{-1}(x_k-A_{k-1}x_{k-1}-v_k) & , \quad k=1\ldots K\end{cases}$$

$$J_{y,k}(x)=\frac{1}{2}(y_k-C_kx_k)^TR_k^{-1}(y_k-C_kx_k) \quad , \quad k=0\ldots K$$

这些都是平方马氏距离(Mahalanobis distances)。我们对整体定义**目标函数(objective function)**，寻找最小值

$$J(x)=\sum_{k=0}^{K}(J_{v,k}(x)+J_{y,k}(x))$$

我们还可以对Jx函数进行修改(添加限制或者惩罚项)，这些看情况而定。

综上所述，**我们把最大后验估计搞成了求Jx的最小值的问题。**也是无约束最优化问题。

$$\hat{x}=argmin_x J(x)$$

解此无约束最优化问题：

我们想办法把函数搞成x的平方的形式：

我们令所有已知量为z，所有待估计量为x：

$$z=[\check{x}_0,v_1,\ldots,v_K,y_0,\ldots,y_K]^T$$
$$x=[x_0,\ldots,x_K]^T$$

我们定义以下块矩阵：? ? ? ? ? ? ? ? ? ? ? ? ?

$$\mathbf{H}=\left[\begin{array}{ccccc}\mathbf{1} & & & & \\ -\mathbf{A}_0 & \mathbf{1} & & & \\ & \ddots & \ddots & & \\ & & -\mathbf{A}_{K-1} & \mathbf{1} & \\ \hline \mathbf{C}_0 & & & & \\ & \mathbf{C}_1 & & & \\ & & \ddots & & \\ & & & & \mathbf{C}_K\end{array}\right],$$

$$\mathbf{W}=\left[\begin{array}{cccc|cccc}\check{\mathbf{P}}_0 & & & & & & & \\ & \mathbf{Q}_1 & & & & & & \\ & & \ddots & & & & & \\ & & & \mathbf{Q}_K & & & & \\ \hline & & & & \mathbf{R}_0 & & & \\ & & & & & \mathbf{R}_1 & & \\ & & & & & & \ddots & \\ & & & & & & & \mathbf{R}_K\end{array}\right],$$

我们可以得到：

$$J(x)=\frac{1}{2}(z-Hx)^TW^{-1}(z-Hx)$$

同时，我们也有：

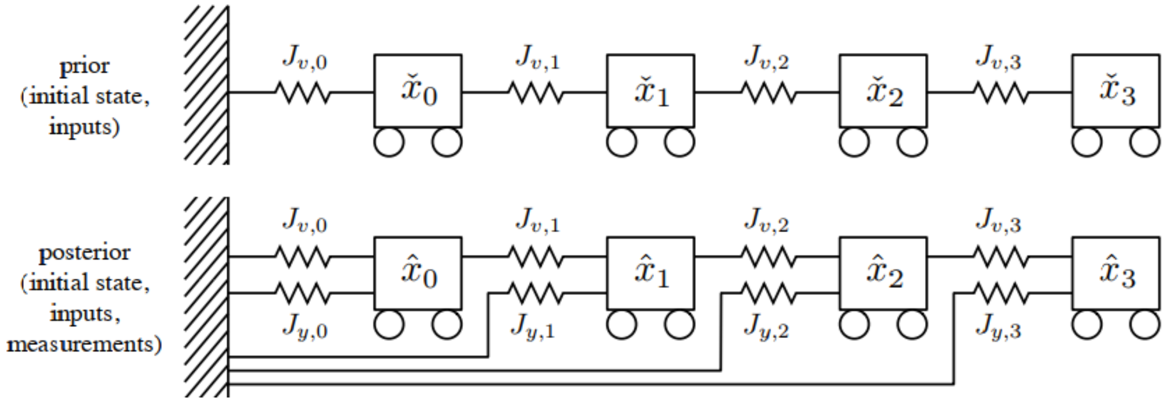
$$p(z|x)=\eta exp(-\frac{1}{2}(z-Hx)^TW^{-1}(z-Hx))$$

η为归一化常数。此时，我们对J(x)求导数：

$$\frac{dJ(x)}{dx}=-H^TW^{-1}(z-H\hat{x})=0$$
$$(H^Tw^{-1}H)\hat{x}=H^TW^{-1}z$$

不过这个求逆不需要硬求，可以利用矩阵的稀疏性，后续会讲解。

形象化



如图所示，批量估计问题的一个直观解释。函数的每一项可以认为是弹簧中储存的能量，根据运动方程可以得到先验，其中先验与上一个状态有关；而测量量直接与状态量相关，这样我们使系统能量最小即可得到后验估计。

3.1.3 贝叶斯推断

我们已经见到通过优化方法来进行批量线性高斯估计的方法，我们现在来通过贝叶斯法则，而不是最大化后验。**我们要从先验的概率密度出发，然后再用测量的数据来更新它。**

单独的k时刻方程，统一写成矩阵形式

先验：通过初始值和输入构建先验估计，利用运动方程得到：

$$x_k=A_{k-1}x_{k-1}+v_k+w_k$$

在**提升形式**下：我们有

$$x=A(v+w)$$

其中w是初始状态和运动噪声的提升形式：

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

提升形式推导:

我们注意：我们也要估计x0，且输入v0=x0已知，(x0先验(观测方程)=x0已知+wk噪声)

我们得到：

$$\begin{aligned} x_0 &= \check{x}_0 + w_0 \\ x_1 &= A_0 x_0 + v_1 + w_1 \\ &\cdots \\ x_k &= A_{k-1} x_{k-1} + v_k + w_k \end{aligned}$$

由于每一项都有v+w的部分，因此，A矩阵的对角线上全为1。其次，根据递推，上式的值代入下式总能得到规律，因此我们易得A矩阵的形式。

提升形式下，均值：（噪声w的均值为0）

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

协方差：

$$\begin{aligned} \check{P} &= E[(x - E[x])(x - E[x])^T] \\ &= E[(A(v+w) - Av)(A(v+w) - Av)^T] \\ &= E[Aww^T A^T](A\text{为已知常数矩阵}) \\ &= AQA^T \end{aligned}$$

其中，(QK为每次噪声的方差)

$$Q = E[ww^T] = diag(\check{P}_0, Q_1, \ldots, Q_K)$$

先验可以表示为：这么写的原因是运动方程，v的条件下x的值，p(x|v)

$$p(x|v) = N(\check{x}, \check{P}) = N(Av, AQA^T)$$

继续，测量模型：

$$y_k = C_k x_k + n_k$$

我们要写成**y=Cx+n**的形式，接下来求C。

$$C = diag(C_0, C_1, \ldots, C_K)$$

综上，我们有：

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

其中

$$R = E[nn^T] = diag(R_0, R_1, \ldots, R_k)$$

我们对上式进行推导:

均值比较好理解，现在看一下协方差矩阵：

$$
\begin{aligned}
p(y|v) &= N(C\check{x}, \check{P}_y) \\
\check{P}_y &= E[(y-E(y))(y-E(y))^T] \\
&= E[(Cx+n-C\check{x})(Cx+n-C\check{x})^T] \\
&= E[(CAw+n)(CAw+n)^T] \\
&= (n_k\text{之间相互独立，均值为零})E[CAww^TA^TC^T]+E[nn^T] \\
&= C\check{P}C^T+R
\end{aligned}
$$

$$
\begin{aligned}
&E[(x-E(x))(y-E(y))] \\
&= E[Aw(CAw+n)^T] \\
&= E[Aww^TA^TC^T]+E[n^T] \\
&= \check{P}C^T
\end{aligned}
$$

我们可以进行因式分解：

$$
\begin{aligned}
&p(x,y|v)(\text{条件概率公式}) \\
&= \frac{p(x,y,v)}{p(v)} \\
&= \frac{p(x,y,v)}{p(y,v)}\frac{p(y,v)}{p(v)} \\
&= p(x|v,y)p(y|v)
\end{aligned}
$$

我们只关心p(x|v,y)，这是全贝叶斯后验概率，观测y输入v的条件下，x的概率分布。

我们用(2.2.3)节的方法，写成：(2.2.3推导)

$$
\begin{aligned}
p(\mathbf{x}|\mathbf{v},\mathbf{y}) = \mathcal{N}\Big(&\check{\mathbf{x}} + \check{\mathbf{P}}\mathbf{C}^T(\mathbf{C}\check{\mathbf{P}}\mathbf{C}^T+\mathbf{R})^{-1}(\mathbf{y}-\mathbf{C}\check{\mathbf{x}}), \\
&\check{\mathbf{P}} - \check{\mathbf{P}}\mathbf{C}^T(\mathbf{C}\check{\mathbf{P}}\mathbf{C}^T+\mathbf{R})^{-1}\mathbf{C}\check{\mathbf{P}}\Big). \quad (3.28)
\end{aligned}
$$

附带上式推导：由2.2.3节中得到：

利用式子(2.75)中的SMW恒等式，可以将上式转换为：

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

附带公式：

我们为了便于优化，整理以下上式的均值项：

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

我们代入：

$$
\begin{aligned}
\check{x} &= Av \\
\check{P}^{-1} &= (AQA^T)^{-1} = A^{-T}Q^{-1}A^{-1}
\end{aligned}
$$

可以得到：

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

这里需要计算A的逆，不过，A的逆有一个很好看的形式：？？？？？

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

我们定义：

$$z=[v \quad y]^T$$
$$H=[A^{-1} \quad C]^T$$
$$W=\begin{bmatrix}Q & \\ & R\end{bmatrix}$$

我们就可以把系统写成：

$$(H^TW^{-1}H)\hat{x}=H^TW^{-1}z$$

3.1.4 Existence,Uniqueness and Observability

译作：存在性，唯一性，能观性

我们看线性高斯系统估计中有且仅有唯一解的情况：当且仅当H^TW^-1H可逆。有：

$$\hat{x}=(H^TW^{-1}H)^{-1}H^TW^{-1}z$$
$$rank(H^TW^{-1}H)=N(K+1)$$

由于W的逆为实对称正定矩阵，那么我们有：

$$rank(H^TH)=rank(H^T)=N(K+1)$$

我们现在有两种情况：

1. 我们有良好的初始先验 x0
2. 我们没有良好的初始先验

case1 Knowledge of initial state

我们直接看H^T即可：

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

很明显，矩阵满秩，因此，我们能够得到唯一解xhat，只需要：

$$
\begin{gathered}
\check{P}_0 > 0 \\
Q_k > 0(\text{正定})
\end{gathered}
$$

case2 No knowledge of initial state

没有良好的初始先验值，我们就只能去掉H^T的第一列，变为：

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

这个矩阵有K+1行，每块大小为N，将最上一行移到最下，得到：

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

我们为了将其变为阶梯型，我们用A0T乘第一行，然后加到最后一行上；再用A0TA1T乘第二行，加到最后一行上，。。。。。。。得到

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

那么，我们只需要右下角的分块矩阵的秩为N即可，即：

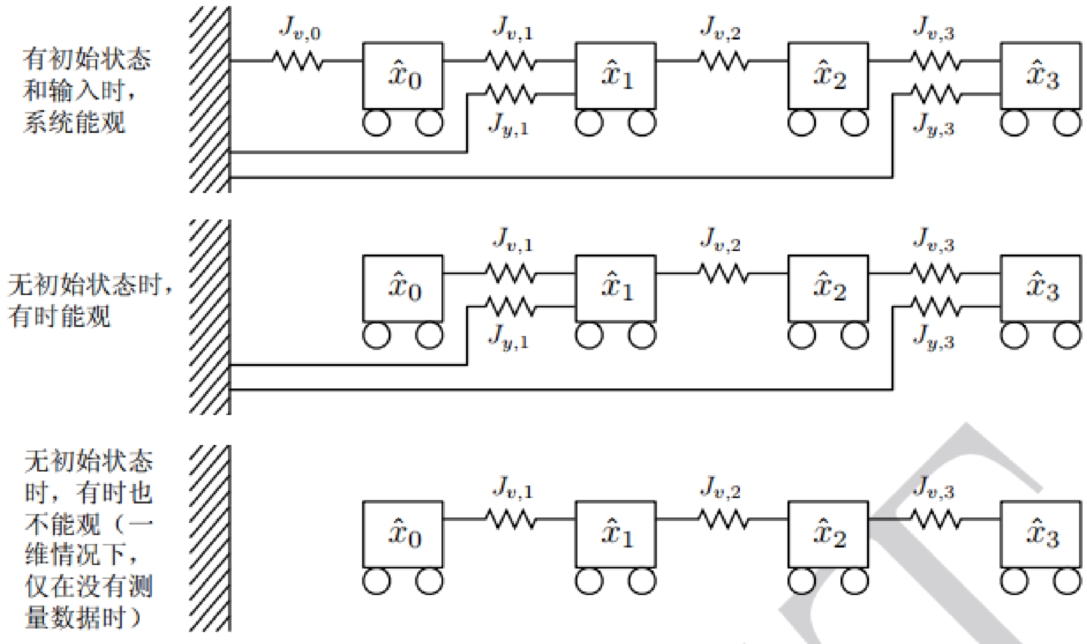
$$\operatorname{rank}\left[\begin{array}{lllll} \boldsymbol{C}_0^{\mathrm{T}} & \boldsymbol{A}_0^{\mathrm{T}} \boldsymbol{C}_1^{\mathrm{T}} & \boldsymbol{A}_0^{\mathrm{T}} \boldsymbol{A}_1^{\mathrm{T}} \boldsymbol{C}_2^{\mathrm{T}} & \cdots & \boldsymbol{A}_0^{\mathrm{T}} \cdots \boldsymbol{A}_{K-1}^{\mathrm{T}} \boldsymbol{C}_K^{\mathrm{T}} \end{array}\right]=N$$

根据克莱哈密顿定理，我们直接给出结论：

式(3.3.4)的解存在且唯一的条件是：

$$O=\left[\begin{array}{c} C \\ CA \\ \ldots \\ CA^{N-1} \end{array}\right]$$
$$Q_k>0, R_k>0, rank(O)=N$$

我们用弹簧模型来解释一下：如图所示



1. 当我们有初始状态时，系统总是能观的(无法移动任意一个(或多个)重物而不改变至少一个弹簧的长度)，这时候有一个唯一的最小能量状态。
2. 中间例子，就算不知道初始状态，也是能观的，有唯一解。
3. 最后一个就不行了，没有初始状态，也没有观测值，没有唯一解。

3.1.5 MAP的协方差

xhat代表x状态的最有估计，但是这个xhat有多少置信度?

3.2 Recursive Discrete-Time Smoothing

译作：离散时间的递归平滑算法

概述：离散时间的批量估计中，给出了良好的结论，也容易从最小二乘的角度来理解？？？？？？思考。但是，蛮力去解线性方程组是非常低效的。幸运的是，因为左侧的协方差矩阵的逆具有稀疏结构(对角块)，我们可以用这个性质有效地求解。

$$(H^T W^{-1} H)\hat{x} = H^T W^{-1} z$$

3.2.1 Exploiting Sparsity in the Batch Solution

译作：利用稀疏性求解批量估计

如之前所讨论，我们有如下结论：

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

其中，*代表非零块。可以采取稀疏**Cholesky分解**进行求解。Cholesky分解详见数值分析课程。

我们可以将上述矩阵分解为：

$$H^TW^{-1}H = LL^T$$

由于等式左侧为三对角块，因此L的形式如下：??????

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

然后，先解：

$$Ld = H^TW^{-1}z$$

再解：

$$L^T\hat{x} = d$$

3.2.2 Cholesky Smoother

译作：Cholesky 平滑算法

我们看一下L的形式：

$$
\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_0 & & & & & \\ \boldsymbol{L}_{10} & \boldsymbol{L}_1 & & & & \\ & \boldsymbol{L}_{21} & \boldsymbol{L}_2 & & & \\ & & \ddots & \ddots & & \\ & & & \boldsymbol{L}_{K-1,K-2} & \boldsymbol{L}_{K-1} & \\ & & & & \boldsymbol{L}_{K,K-1} & \boldsymbol{L}_K \end{bmatrix} \tag{3.60}
$$

我们展开并对比(这与矩阵的Cholesky分解过程较为类似)，一个位置一个位置的矩阵块对应相等即可，我们得到：

$$
\begin{aligned}
\boldsymbol{L}_0\boldsymbol{L}_0^{\mathrm{T}} &= \underbrace{\check{\boldsymbol{P}}_0^{-1} + \boldsymbol{C}_0^{\mathrm{T}}\boldsymbol{R}_0^{-1}\boldsymbol{C}_0}_{\boldsymbol{I}_0} + \boldsymbol{A}_0^{\mathrm{T}}\boldsymbol{Q}_1^{-1}\boldsymbol{A}_0 && \text{(3.61a)} \\
\boldsymbol{L}_{10}\boldsymbol{L}_0^{\mathrm{T}} &= -\boldsymbol{Q}_1^{-1}\boldsymbol{A}_0 && \text{(3.61b)} \\
\boldsymbol{L}_1\boldsymbol{L}_1^{\mathrm{T}} &= \underbrace{-\boldsymbol{L}_{10}\boldsymbol{L}_{10}^{\mathrm{T}} + \boldsymbol{Q}_1^{-1} + \boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}_1^{-1}\boldsymbol{C}_1}_{\boldsymbol{I}_1} + \boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{Q}_2^{-1}\boldsymbol{A}_1 && \text{(3.61c)} \\
\boldsymbol{L}_{21}\boldsymbol{L}_1^{\mathrm{T}} &= -\boldsymbol{Q}_2^{-1}\boldsymbol{A}_1 && \text{(3.61d)} \\
&\vdots \\
\boldsymbol{L}_{K-1}\boldsymbol{L}_{K-1}^{\mathrm{T}} &= \underbrace{-\boldsymbol{L}_{K-1,K-2}\boldsymbol{L}_{K-1,K-2}^{\mathrm{T}} + \boldsymbol{Q}_{K-1}^{-1} + \boldsymbol{C}_{K-1}^{\mathrm{T}}\boldsymbol{R}_{K-1}^{-1}\boldsymbol{C}_{K-1}}_{\boldsymbol{I}_{K-1}} + \boldsymbol{A}_{K-1}^{\mathrm{T}}\boldsymbol{Q}_K^{-1}\boldsymbol{A}_{K-1} && \text{(3.61e)} \\
\boldsymbol{L}_{K,K-1}\boldsymbol{L}_{K-1}^{\mathrm{T}} &= -\boldsymbol{Q}_K^{-1}\boldsymbol{A}_{K-1} && \text{(3.61f)} \\
\boldsymbol{L}_K\boldsymbol{L}_K^{\mathrm{T}} &= \underbrace{-\boldsymbol{L}_{K,K-1}\boldsymbol{L}_{K,K-1}^{\mathrm{T}} + \boldsymbol{Q}_K^{-1} + \boldsymbol{C}_K^{\mathrm{T}}\boldsymbol{R}_K^{-1}\boldsymbol{C}_K}_{\boldsymbol{I}_K} && \text{(3.61g)}
\end{aligned}
$$

如上所示，式子中我们标出了一些中间变量Ik。从这些式子可以看出，**我们先解L0(一步小型块稠密矩阵的分解)，然后将结果代入2式中得到L10，从上往下依次求解。**

接下来，我们从来解d：(方法与上类似)

$$
Ld = H^T W^{-1} z
$$

展开得到：

$$
\begin{aligned}
\boldsymbol{L}_0\boldsymbol{d}_0 &= \underbrace{\check{\boldsymbol{P}}_0^{-1}\check{\boldsymbol{x}}_0 + \boldsymbol{C}_0^{\mathrm{T}}\boldsymbol{R}_0^{-1}\boldsymbol{y}_0}_{\boldsymbol{q}_0} - \boldsymbol{A}_0^{\mathrm{T}}\boldsymbol{Q}_1^{-1}\boldsymbol{v}_1 \\
\boldsymbol{L}_1\boldsymbol{d}_1 &= \underbrace{-\boldsymbol{L}_{10}\boldsymbol{d}_0 + \boldsymbol{Q}_1^{-1}\boldsymbol{v}_1 + \boldsymbol{C}_1^{\mathrm{T}}\boldsymbol{R}_1^{-1}\boldsymbol{y}_1}_{\boldsymbol{q}_1} - \boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{Q}_2^{-1}\boldsymbol{v}_2 \\
&\vdots \\
\boldsymbol{L}_{K-1}\boldsymbol{d}_{K-1} &= \underbrace{-\boldsymbol{L}_{K-1,K-2}\boldsymbol{d}_{K-2} + \boldsymbol{Q}_{K-1}^{-1}\boldsymbol{v}_{K-1} + \boldsymbol{C}_{K-1}^{\mathrm{T}}\boldsymbol{R}_{K-1}^{-1}\boldsymbol{y}_{K-1}}_{\boldsymbol{q}_{K-1}} - \boldsymbol{A}_{K-1}^{\mathrm{T}}\boldsymbol{Q}_K^{-1}\boldsymbol{v}_K \\
\boldsymbol{L}_K\boldsymbol{d}_K &= \underbrace{-\boldsymbol{L}_{K,K-1}\boldsymbol{d}_{K-1} + \boldsymbol{Q}_K^{-1}\boldsymbol{v}_K + \boldsymbol{C}_K^{\mathrm{T}}\boldsymbol{R}_K^{-1}\boldsymbol{y}_K}_{\boldsymbol{q}_K}
\end{aligned}
$$

最后，我们来解xhat：

$$
L^T \hat{x} = d
$$

展开块矩阵运算，得到：

$$
\begin{aligned}
\boldsymbol{L}_K^{\mathrm{T}} \hat{\boldsymbol{x}}_K &= \boldsymbol{d}_K \\
\boldsymbol{L}_{K-1}^{\mathrm{T}} \hat{\boldsymbol{x}}_{K-1} &= -\boldsymbol{L}_{K,K-1}^{\mathrm{T}} \hat{\boldsymbol{x}}_K + \boldsymbol{d}_{K-1} \\
&\vdots \\
\boldsymbol{L}_1^{\mathrm{T}} \hat{\boldsymbol{x}}_1 &= -\boldsymbol{L}_{21}^{\mathrm{T}} \hat{\boldsymbol{x}}_2 + \boldsymbol{d}_1 \\
\boldsymbol{L}_0^{\mathrm{T}} \hat{\boldsymbol{x}}_0 &= -\boldsymbol{L}_{10}^{\mathrm{T}} \hat{\boldsymbol{x}}_1 + \boldsymbol{d}_0
\end{aligned}
$$

这样，我们就解完了。我们综合一下，整理得到：

对于中间量 $\boldsymbol{I}_k$ 和 $\boldsymbol{q}_k$，我们能够综合前向与后向两步（即解 $\boldsymbol{L}$ 和 $\boldsymbol{d}$ 的过程），将它们写成：

前向：$k = 1 \ldots K$

$$\boldsymbol{L}_{k-1} \boldsymbol{L}_{k-1}^{\mathrm{T}} = \boldsymbol{I}_{k-1} + \boldsymbol{A}_{k-1}^{\mathrm{T}} \boldsymbol{Q}_k^{-1} \boldsymbol{A}_{k-1} \tag{3.66a}$$

$$\boldsymbol{L}_{k-1} \boldsymbol{d}_{k-1} = \boldsymbol{q}_{k-1} - \boldsymbol{A}_{k-1}^{\mathrm{T}} \boldsymbol{Q}_k^{-1} \boldsymbol{v}_k \tag{3.66b}$$

$$\boldsymbol{L}_{k,k-1} \boldsymbol{L}_{k-1}^{\mathrm{T}} = -\boldsymbol{Q}_k^{-1} \boldsymbol{A}_{k-1} \tag{3.66c}$$

$$\boldsymbol{I}_k = -\boldsymbol{L}_{k,k-1} \boldsymbol{L}_{k,k-1}^{\mathrm{T}} + \boldsymbol{Q}_k^{-1} + \boldsymbol{C}_k^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{C}_k \tag{3.66d}$$

$$\boldsymbol{q}_k = -\boldsymbol{L}_{k,k-1} \boldsymbol{d}_{k-1} + \boldsymbol{Q}_k^{-1} \boldsymbol{v}_k + \boldsymbol{C}_k^{\mathrm{T}} \boldsymbol{R}_k^{-1} \boldsymbol{y}_k \tag{3.66e}$$

后向：$k = K \ldots 1$

$$\boldsymbol{L}_{k-1}^{\mathrm{T}} \hat{\boldsymbol{x}}_{k-1} = -\boldsymbol{L}_{k,k-1}^{\mathrm{T}} \hat{\boldsymbol{x}}_k + \boldsymbol{d}_{k-1} \tag{3.66f}$$

这些量的初始值为：

$$\boldsymbol{I}_0 = \check{\boldsymbol{P}}_0^{-1} + \boldsymbol{C}_0^{\mathrm{T}} \boldsymbol{R}_0^{-1} \boldsymbol{C}_0 \tag{3.67a}$$

$$\boldsymbol{q}_0 = \check{\boldsymbol{P}}_0^{-1} \check{\boldsymbol{x}}_0 + \boldsymbol{C}_0^{\mathrm{T}} \boldsymbol{R}_0^{-1} \boldsymbol{y}_0 \tag{3.67b}$$

$$\hat{\boldsymbol{x}}_K = \boldsymbol{L}_K^{-\mathrm{T}} \boldsymbol{d}_K \tag{3.67c}$$

3.2.3 Rauch-Tung-Striebel 平滑算法

我们求解上式3.66c中的L(k,k-1)，这需要式子a，b的结果。解完后将结果代入3.66d中得到Ik：

令：

$$
\begin{aligned}
\hat{P}_{k,f} &= I_k^{-1} \\
\check{P}_{k,f} &= (A_{k-1} I_{k-1}^{-1} + Q_k)^{-1}
\end{aligned}
$$

我们将上式写成：

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

这里，第一个式子表示预测的协方差，通过输入方程。而第二个式子表示修正的协方差，通过观测方程。

我们用卡尔曼增益Kk来表示：定义

$$K_k = \hat{P}_{k,f} C_k^T R_k^{-1}$$

$$\text{经过推导：（}P54\text{）}$$

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

综上所述，式（3.69a），式（3.71），式（3.76a），式（3.78），式（3.81）构成了 Rauch-Tung-Striebel 平滑算法：

前向：$k = 1 \ldots K$

$$\check{\boldsymbol{P}}_{k,f} = \boldsymbol{A}_{k-1}\hat{\boldsymbol{P}}_{k-1,f}\boldsymbol{A}_{k-1}^{\mathrm{T}} + \boldsymbol{Q}_k \tag{3.82a}$$

$$\check{\boldsymbol{x}}_{k,f} = \boldsymbol{A}_{k-1}\hat{\boldsymbol{x}}_{k-1,f} + \boldsymbol{v}_k \tag{3.82b}$$

$$\boldsymbol{K}_k = \check{\boldsymbol{P}}_{k,f}\boldsymbol{C}_k^{\mathrm{T}}\left(\boldsymbol{C}_k\check{\boldsymbol{P}}_{k,f}\boldsymbol{C}_k^{\mathrm{T}} + \boldsymbol{R}_k\right)^{-1} \tag{3.82c}$$

$$\hat{\boldsymbol{P}}_{k,f} = (\boldsymbol{1} - \boldsymbol{K}_k\boldsymbol{C}_k)\check{\boldsymbol{P}}_{k,f} \tag{3.82d}$$

$$\hat{\boldsymbol{x}}_{k,f} = \check{\boldsymbol{x}}_{k,f} + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{C}_k\check{\boldsymbol{x}}_{k,f}) \tag{3.82e}$$

后向：$k = K \ldots 1$

$$\hat{\boldsymbol{x}}_{k-1} = \hat{\boldsymbol{x}}_{k-1,f} + \hat{\boldsymbol{P}}_{k-1,f}\boldsymbol{A}_{k-1}^{\mathrm{T}}\check{\boldsymbol{P}}_{k,f}^{-1}(\hat{\boldsymbol{x}}_k - \check{\boldsymbol{x}}_{k,f}) \tag{3.82f}$$

它们的初始值为：

$$\check{\boldsymbol{P}}_{0,f} = \check{\boldsymbol{P}}_0 \tag{3.83a}$$

$$\check{\boldsymbol{x}}_{0,f} = \check{\boldsymbol{x}}_0 \tag{3.83b}$$

$$\hat{\boldsymbol{x}}_K = \hat{\boldsymbol{x}}_{K,f} \tag{3.83c}$$

前向的五个过程被称为卡尔曼滤波器，这六个公式表达的式RTS平滑算法。这件事情成立的原因，正式由于优化算法左侧矩阵的特殊三对角块的稀疏结构。

3.3 离散时间的递归滤波算法

本章其余部分，介绍其他几种推导和理解卡尔曼滤波器的方法。

这张图体现了批量估计算法和滤波器估计(卡尔曼滤波)的区别。

smoothers use all available information to estimate states

$$\overbrace{\underbrace{\check{\mathbf{x}}_0, \mathbf{y}_0, \mathbf{v}_1, \mathbf{y}_1, \mathbf{v}_2, \mathbf{y}_2, \ldots, \mathbf{v}_{k-1}, \mathbf{y}_{k-1}, \mathbf{v}_k, \mathbf{y}_k}_{\hat{\mathbf{x}}_{k,f}}, \mathbf{v}_{k+1}, \mathbf{y}_{k+1}, \ldots, \mathbf{v}_K, \mathbf{y}_K}^{\hat{\mathbf{x}}_k}$$

filters only use past/current information to estimate states

3.3.1 Factoring the Batch Solution

为了把结论写成递归的形式，我们重新排列一下批量优化中的变量，重新定义z、H、W(只是交换了矩阵行的位置)。

$$\mathbf{z} = \left[\begin{array}{c} \check{\mathbf{x}}_0 \\ \mathbf{y}_0 \\ \hline \mathbf{v}_1 \\ \mathbf{y}_1 \\ \hline \mathbf{v}_2 \\ \mathbf{y}_2 \\ \hline \vdots \\ \hline \mathbf{v}_K \\ \mathbf{y}_K \end{array}\right], \qquad \mathbf{H} = \left[\begin{array}{cccccc} \mathbf{1} & & & & & \\ \mathbf{C}_0 & & & & & \\ \hline -\mathbf{A}_0 & \mathbf{1} & & & & \\ & \mathbf{C}_1 & & & & \\ \hline & -\mathbf{A}_1 & \mathbf{1} & & & \\ & & \mathbf{C}_2 & & & \\ \hline & & \ddots & \ddots & & \\ \hline & & & & -\mathbf{A}_{K-1} & \mathbf{1} \\ & & & & & \mathbf{C}_K \end{array}\right],$$

$$\mathbf{W} = \left[\begin{array}{cc|cc|cc|c|cc} \check{\mathbf{P}}_0 & & & & & & & & \\ & \mathbf{R}_0 & & & & & & & \\ \hline & & \mathbf{Q}_1 & & & & & & \\ & & & \mathbf{R}_1 & & & & & \\ \hline & & & & \mathbf{Q}_2 & & & & \\ & & & & & \mathbf{R}_2 & & & \\ \hline & & & & & & \ddots & & \\ \hline & & & & & & & \mathbf{Q}_K & \\ & & & & & & & & \mathbf{R}_K \end{array}\right], \quad (3.84)$$

如上式，这些线分开了不同时间戳的变量。

现在，我们来考虑其在概率密度函数层面的分解(factorization)。我们现在想考虑k时刻的状态，**我们进行边缘化操作，把其他时刻的变量积分出去**

$$p(x_k|v, y) = \int_{x_i, \forall i \neq k} p(x_i|v, y) dx_{i, \forall i \neq k}$$

为了实现分解，我们要用到H的稀疏结构。先把H分解成12个块(仅有6个非零块)。如下所示：

$$\begin{array}{ll} \mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & & \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \\ & \mathbf{H}_{32} & \mathbf{H}_{33} \\ & & \mathbf{H}_{43} \end{bmatrix} & \begin{array}{l} \text{information from } 0 \ldots k-1 \\ \text{information from } k \\ \text{information from } k+1 \\ \text{information from } k+2 \ldots K \end{array} \\ \qquad\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ \text{states from } 0 \ldots k-1 & \text{states from } k & \text{states from } k+1 \ldots K \end{array} & \end{array}$$

举个k=2， K=4的例子：

$$
\mathbf{H}=\left[\begin{array}{cc|c|cc}
\mathbf{1} & & & & \\
\mathbf{C}_0 & & & & \\
-\mathbf{A}_0 & \mathbf{1} & & & \\
 & \mathbf{C}_1 & & & \\
\hline
 & -\mathbf{A}_1 & \mathbf{1} & & \\
 & & \mathbf{C}_2 & & \\
\hline
 & & -\mathbf{A}_2 & \mathbf{1} & \\
 & & & \mathbf{C}_3 & \\
\hline
 & & & -\mathbf{A}_3 & \mathbf{1} \\
 & & & & \mathbf{C}_4
\end{array}\right]. \tag{3.88}
$$

我们把z和W也转换成这个样式：

$$
\mathbf{z}=\begin{bmatrix}\mathbf{z}_1\\ \mathbf{z}_2\\ \mathbf{z}_3\\ \mathbf{z}_4\end{bmatrix},\quad \mathbf{W}=\begin{bmatrix}\mathbf{W}_1 & & & \\ & \mathbf{W}_2 & & \\ & & \mathbf{W}_3 & \\ & & & \mathbf{W}_4\end{bmatrix}. \tag{3.89}
$$

那么，具体的矩阵H^TW^-1H计算为：

其中，Lij为中间变量。

同样，我们计算H^TW^{-1}z计算为：

其中，ri为中间变量。

3.3.2 通过MAP推导卡尔曼滤波

讲解如何将上一节的前向估计转换为递归滤波器(卡尔曼滤波器的形式)。假设我们现在已经有了k-1时刻的前向估计：

$$\{\hat{x}_{k-1}, \hat{P}_{k-1}\}$$

我们要计算：

$$\{\hat{x}_k, \hat{P}_k\}$$

其实，我们无需从头开始，只需要使用k-1时刻的状态以及k时刻的vk(输入)，yk(观测)来估计k时刻的状态：

$$\{\hat{x}_{k-1}, \hat{P}_{k-1}, v_k, y_k\}->\{\hat{x}_k, \hat{P}_k\}$$

因此，我们直接写出k-1时刻的估计与k时刻估计的关系：

$$
\underbrace{\overbrace{\underbrace{\check{\boldsymbol{x}}_0, \boldsymbol{y}_0, \boldsymbol{v}_1, \boldsymbol{y}_1, \boldsymbol{v}_2, \boldsymbol{y}_2, \ldots, \boldsymbol{v}_{k-1}, \boldsymbol{y}_{k-1}}_{\hat{\boldsymbol{x}}_{k-1}}, \boldsymbol{v}_k}^{\hat{\boldsymbol{x}}'_{k-1}}, \boldsymbol{y}_k}_{\hat{\boldsymbol{x}}_k}, \boldsymbol{v}_{k+1}, \boldsymbol{y}_{k+1}, \ldots, \boldsymbol{v}_K, \boldsymbol{y}_K
$$

我们定义：

$$\hat{x} = \begin{bmatrix} \hat{x}'_{k-1} \\ \hat{x}_k \end{bmatrix}$$

其中，x'k-1是指利用了vk和yk估计得到的(利用了未来信息)。

3.3.3 通过贝叶斯推断推导卡尔曼滤波

设k-1时刻的高斯先验为：

$$p(x_{k-1}|\check{x}_0, v_{1:k-1}, y_{0,k-1}) = N(\hat{x}_{k-1}, \hat{P}_{k-1})$$

首先，**对于预测部分**，我们考虑最近时刻的输入vk，来计算k时刻的"先验"：

$$p(x_k|\check{x}_0, v_{1:k}, y_{0:k-1}) = N(\check{x}_k, \check{P}_k)$$

其中：

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

我们简单对上述两式进行一下推导：

$$\begin{gathered} \check{P}_k(\text{根据方差定义}) = E[(x_k - E[x_k])(x_k - E[x_k])^T] \\ = E[(A_{k-1}x_{k-1} + v_k + w_k - A_{k-1}\hat{x}_{k-1} - v_k)(A_{k-1}x_{k-1} + v_k + w_k - A_{k-1}\hat{x}_{k-1} - v_k)^T] \\ = A_{k-1}E[(x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})^T]A_{k-1}^T + E[w_k w_k^T] + E[w_k \times X] \\ (\text{矩阵}X\text{为由}xk\text{分量组成的随机变量矩阵，与}w_k\text{不相关}) \\ =\ldots\ldots + E[w_k]E[X] = \ldots\ldots + 0 \\ = A_{k-1}\hat{P}_{k-1}A_{k-1}^T + Q_k \end{gathered}$$

之后，对于更新部分(测量)，我们将状态与最近的依次测量写成联合高斯分布的形式：

$$\begin{aligned} p(\boldsymbol{x}_k, \boldsymbol{y}_k|\check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:k}, \boldsymbol{y}_{0:k-1}) &= \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}\right) \\ &= \mathcal{N}\left(\begin{bmatrix} \check{\boldsymbol{x}}_k \\ \boldsymbol{C}_k\check{\boldsymbol{x}}_k \end{bmatrix}, \begin{bmatrix} \check{\boldsymbol{P}}_k & \check{\boldsymbol{P}}_k\boldsymbol{C}_k^{\mathrm{T}} \\ \boldsymbol{C}_k\check{\boldsymbol{P}}_k & \boldsymbol{C}_k\check{\boldsymbol{P}}_k\boldsymbol{C}_k^{\mathrm{T}} + \boldsymbol{R}_k \end{bmatrix}\right) \end{aligned} \tag{3.126}$$

根据2.2.3节，我们总能将p(x,y)写成p(x|y)p(y)

其中：

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y}) \tag{2.53a}$$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y), \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}\right) \tag{2.53b}$$

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_{yy}) \tag{2.53c}$$

根据上述推断，我们直接得到xk的条件分布，即后验概率(上式把yk拿到了后面)：

$$p(\boldsymbol{x}_k|\check{\boldsymbol{x}}_0, \boldsymbol{v}_{1:k}, \boldsymbol{y}_{0:k}) = \mathcal{N}\Big(\underbrace{\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y}_k - \boldsymbol{\mu}_y)}_{\hat{\boldsymbol{x}}_k}, \underbrace{\boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}}_{\hat{\boldsymbol{P}}_k}\Big) \tag{3.127}$$

此处，我们定义xk_hat 为均值，Pk_hat为方差，代入之前的结果则有：

$$\boldsymbol{K}_k = \check{\boldsymbol{P}}_k \boldsymbol{C}_k^{\mathrm{T}} \left(\boldsymbol{C}_k \check{\boldsymbol{P}}_k \boldsymbol{C}_k^{\mathrm{T}} + \boldsymbol{R}_k\right)^{-1} \tag{3.128a}$$

$$\hat{\boldsymbol{P}}_k = \left(\mathbf{1} - \boldsymbol{K}_k \boldsymbol{C}_k\right) \check{\boldsymbol{P}}_k \tag{3.128b}$$

$$\hat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \boldsymbol{C}_k \check{\boldsymbol{x}}_k\right) \tag{3.128c}$$

3.3.4 从增益最优化的角度来看卡尔曼滤波

我们通常说**卡尔曼滤波是最优的**。同时，我们也确实从MAP中推导出了卡尔曼滤波与递归形式优化之间的关系。现从增益最优化的角度来看卡尔曼滤波。

假设一个估计器：

$$\hat{x}_k = \check{x}_k + K_k(y_k - C_k\check{x}_k)$$

但此时**不知道如何选取K的值**，我们定义状态误差为：

$$\hat{e}_k = \hat{x}_k - x_k$$

则有：

$$E[\hat{e}_k\hat{e}_k^T] = (1 - K_kC_k)\check{P}_k(1 - K_kC_k)^T + K_kR_kK_k^T$$

解释：协方差估计，第二项是，测量Y噪声带来的协方差。

我们定义出一个**代价函数**：

$$J(K_k) = \frac{1}{2}trE[\hat{e}_k\hat{e}_k^T] = E[\frac{1}{2}\hat{e}_k^T\hat{e}_k]$$

使这个误差最小化，求Kk即可。

因此，对K求导得到：

$$\text{前置公式：} \frac{\partial trXY}{\partial X} = Y^T, \frac{\partial trXZX^T}{\partial X} = 2XZ$$

$$\frac{\partial J(K_k)}{\partial K_k} = -(1 - K_kC_k)\check{P}_kC_k^T + K_kR_k$$

令偏导数=0即可求得：

$$K_k = \check{P}_kC_k^T(C_k\check{P}_kC_k^T + R_k)^{-1}$$

这就是卡尔曼增益的通常表达式。

3.3.5 关于卡尔曼滤波的讨论

以下是卡尔曼滤波的要点：

1. 对于高斯噪声的线性系统，卡尔曼滤波是**最优线性无偏估计**。这意味着它给出的解的协方差位于克拉美罗下界处。
2. 必须有初始状态 x0 P0
3. 协方差部分与均值部分可以独立地递推，有时可以计算一个固定的Kk，用于所有时刻的均值修正。这种做法称为**固定状态的卡尔曼滤波**。
4. 实现当中我们用实际的传感器读数，yk,meas
5. 从后向过程一样可以推导处类似的滤波方法，这种方法会沿着时间反向传递。

非线性场合下，我们需要使用——**扩展卡尔曼滤波器EKF**。

3.3.6 误差动态过程

3.3.7 存在性，唯一性以及能观性

3.4 连续时间的批量估计问题

3.4.1 高斯过程回归

3.4.2 一种稀疏的高斯过程先验

3.4.3 线性时不变情况

3.4.4 与批量离散时间情况的关系