Aerial Robotic Autonomy: Methods and Systems

Linear Gaussian Estimation

Kostas Alexis

Autonomous Robots Lab
Norwegian University of Science and Technology

Table of Contents

Table of Contents

Introduction

Topics to be covered

- Batch, discrete-time estimation
- Recursive approaches and their relation to batch estimation
- Estimation for continuous-time motion models and relevance to discrete-time results and Gaussian process regression.
- Classic results for linear models and Gaussian random variables, including the *Kalman Filter*

Main reference: Barfoot, T.D., 2017. State estimation for robotics. Cambridge University Press.

Table of Contents

Outline

We will introduce the problem setup and the methods used for its solution.
We shall explicitly discuss Maximum A Posteriori (MAP) and Bayesian Inference methods.

Problem Setup

We define the following motion and observation models

$$\text{motion model:} \quad \mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k, \quad k = 1 \ldots K \tag{1}$$

$$\text{observation model:} \quad \mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k, \quad k = 0 \ldots K \tag{2}$$

where k is the discrete-time index and:

- $\mathbf{x}_k \in \mathbb{R}^N$: system state (random variable)
- $\mathbf{x}_0 \in \mathbb{R}^N \sim \mathcal{N}(\check{\mathbf{x}}_0, \check{\mathbf{P}}_0)$: initial state (random variable)
- $\mathbf{v}_k \in \mathbb{R}^N$: input (deterministic variable)
- $\mathbf{w}_k \mathbb{R}^N \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$: process noise (random variable)
- $\mathbf{y}_k \in \mathbb{R}^M$: measurement (random variable)
- $\mathbf{n}_k \in \mathbb{R}^M \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$: measurement noise (random variable)

The noise variables and initial state state are assumed to be uncorrelated with one another and with themselves at different timesteps. $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ is the *transition matrix*, $\mathbf{C}_k \in \mathbb{R}^{M \times N}$ is the *observation matrix*.

Problem Setup

We will base our estimate, $\hat{\mathbf{x}}_k$, on

- Initial state knowledge, $\check{\mathbf{x}}_0$ and associated covariance matrix, $\check{\mathbf{P}}_0$. Sometimes we do not have this information and must practically overcome this issue.
- Inputs, $\mathbf{v}_k$ which are often the output of a controller and known. We also have the associated process noise covariance $\mathbf{Q}_k$.
- Measurements, $\mathbf{y}_{k,meas}$, which are *realizations* of the associated random variables, $\mathbf{y}_k$, and the associated covariance matrix, $\mathbf{R}_k$.

State Estimation Problem

The problem of state estimation is to derive an estimate, $\hat{\mathbf{x}}_k$, of the true state of a system, at one or more timesteps, k, given knowledge of the initial state, $\check{\mathbf{x}}_0$, a sequence of measurements, $\mathbf{y}_{0:K,meas}$, a sequence of inputs, $\mathbf{v}_{1:K}$, as well as knowledge of the system's motion and observation models.

Problem Setup

To show the relationship between various concepts in state estimation, we shall set up the batch Linear-Gaussian (LG) estimation problem using two different paradigms, namely

1. *Bayesian Inference;* we update a prior density over states (based on the initial state knowledge, inputs, and motion model) with our measurements, to produce a posterior (Gaussian) density over states.
2. *Maximum A Posteriori (MAP);* we employ optimization to find the most likely posterior state given the information we have (initial state knowledge, measurements, inputs).

LG Problem: Bayesian inference and MAP arrive at the same result

Although these approaches are different in nature, it can be shown that we arrive at the exact same answer for the LG problem because *the full Bayesian posterior is exactly Gaussian* $\Rightarrow$ the optimization approach will find the maximum (i.e., mode) of a Gaussian and this is the same as the mean.

When we move to nonlinear, non-Gaussian systems, the mean and mode of the posterior are no longer the same and the two methods will arrive to different results.

Maximum A Posteriori (MAP)

In batch estimation, our goal is to solve the following **MAP problem**:

$$\hat{\mathbf{x}} = \arg\max_{x} p(\mathbf{x}|\mathbf{v}, \mathbf{y}) \tag{3}$$

i.e., we want to find the best single estimate for the state of the system at all timesteps, $\hat{\mathbf{x}}$, given the prior information, $\mathbf{v}$, and measurements, $\mathbf{y}$ (*dropped* $_{meas}$ *from* $\mathbf{y}_{meas}$ *for simplicity*).

$$\mathbf{x} = \mathbf{x}_{0:K} = (\mathbf{x}_0, ..., \mathbf{x}_K), \ \mathbf{v} = (\check{\mathbf{x}}_0, \mathbf{v}_{1:K}) = (\check{\mathbf{x}}_0, \mathbf{v}_1, ..., \mathbf{v}_K), \mathbf{y} = \mathbf{y}_{0:K} = (\mathbf{y}_0, ..., \mathbf{y}_K) \tag{4}$$

where the timestemp range may be dropped for convenience of notation (when the range is the largest possible for that variable).

Maximum A Posteriori (MAP)

We begin by re-writting the MAP estimate using Bayes' rule

$$\hat{\mathbf{x}} = \arg\max_x p(\mathbf{x}|\mathbf{v},\mathbf{y}) = \arg\max_x \frac{p(\mathbf{y}|\mathbf{x},\mathbf{v})p(\mathbf{x}|\mathbf{v})}{p(\mathbf{y}|\mathbf{v})} = \arg\max_x p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\mathbf{v}) \quad (5)$$

where we drop the denominator because it is independent of $\mathbf{x}$ and we also drop $\mathbf{v}$ in $p(\mathbf{y}|\mathbf{x},\mathbf{v})$ since it does not affect $\mathbf{y}$ in our system if $\mathbf{x}$ is known.

Note that an assumption we make is that all the noise variables, $\mathbf{w}_k$ and $\mathbf{n}_k$ for $k = 0...K$, are uncorrelated. This allows to use Bayes' rule to factor $p(\mathbf{y}|\mathbf{x})$ as

$$p(\mathbf{y}|\mathbf{x}) = \prod_{k=0}^{K} p(\mathbf{y}_k|\mathbf{x}_k) \quad (6)$$

Maximum A Posteriori (MAP)

Furthermore, Bayes' rule allows us to factor $p(\mathbf{x}|\mathbf{v})$ as

$$p(\mathbf{x}|\mathbf{v}) = p(\mathbf{x}_0|\check{\mathbf{x}}_0) \prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{v}_k) \tag{7}$$

In this linear system the component (Gaussian) densities are given by

$$p(\mathbf{x}_0|\check{\mathbf{x}}_0) = \frac{1}{\sqrt{(2\pi)^N \det \check{\mathbf{P}}_0}} \exp\left(-\frac{1}{2}(\mathbf{x}_0 - \check{\mathbf{x}}_0)^T \check{\mathbf{P}}_0^{-1}(\mathbf{x}_0 - \check{\mathbf{x}}_0)\right) \tag{8}$$

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{v}_k) = \frac{1}{\sqrt{(2\pi)^N \det \mathbf{Q}_k}} \exp\left(-\frac{1}{2}(\mathbf{x}_k - \mathbf{A}_{k-1}\mathbf{x}_{k-1} - \mathbf{v}_k)^T \mathbf{Q}_k^{-1}(\mathbf{x}_k - \mathbf{A}_{k-1}\mathbf{x}_{k-1} - \mathbf{v}_k)\right) \tag{9}$$

$$p(\mathbf{y}_k|\mathbf{x}_k) = \frac{1}{\sqrt{(2\pi)^M \det \mathbf{R}_k}} \exp\left(-\frac{1}{2}(\mathbf{y}_k - \mathbf{C}_k\mathbf{x}_k)^T \mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{C}_k\mathbf{x}_k)\right) \tag{10}$$

Where $\check{\mathbf{P}}_0, \mathbf{Q}_k, \mathbf{R}_k$ are positive-definite by assumption and thus *invertible*.

Maximum A Posteriori (MAP)

We take the logarithm on both sides

$$\ln(p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\mathbf{v})) = \ln p(\mathbf{x}_0|\check{\mathbf{x}}_0) + \sum_{k=1}^{K} \ln p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{v}_k) + \sum_{k=0}^{K} \ln p(\mathbf{y}_k|\mathbf{x}_k) \tag{11}$$

where

$$\ln p(\mathbf{x}_0|\check{\mathbf{x}}_0) = -\frac{1}{2}(\mathbf{x}_0 - \check{\mathbf{x}}_0)^T \check{\mathbf{P}}_0^{-1}(\mathbf{x}_0 - \check{\mathbf{x}}_0) - \underbrace{\frac{1}{2}\left((2\pi)^N \det \check{\mathbf{P}}_0\right)}_{\text{independent of } \mathbf{x}} \tag{12}$$

$$\ln p(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{v}_k) = -\frac{1}{2}(\mathbf{x}_k - \mathbf{A}_{k-1}\mathbf{x}_{k-1} - \mathbf{v}_k)^T \mathbf{Q}_k^{-1}(\mathbf{x}_k - \mathbf{A}_{k-1}\mathbf{x}_{k-1} - \mathbf{v}_k) - \underbrace{\frac{1}{2}\ln\left((2\pi)^N \det \mathbf{Q}_k\right)}_{\text{independent of } \mathbf{x}} \tag{13}$$

$$\ln p(\mathbf{y}_k|\mathbf{x}_k) = -\frac{1}{2}(\mathbf{y}_k - \mathbf{C}_k\mathbf{x}_k)^T \mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{C}_k\mathbf{x}_k) - \underbrace{\frac{1}{2}\ln\left((2\pi)^M \det \mathbf{R}_k\right)}_{\text{independent of } \mathbf{x}} \tag{14}$$

Maximum A Posteriori (MAP)

Noticing that there are terms that do not depend on **x** we define

$$J_{\upsilon,k}(\mathbf{x}) = \begin{cases} \frac{1}{2}(\mathbf{x}_0 - \check{\mathbf{x}}_0)^T \check{\mathbf{P}}_0^{-1}(\mathbf{x}_0 - \check{\mathbf{x}}_0), \ k = 0 \\ \frac{1}{2}(\mathbf{x}_k - \mathbf{A}_{k-1}\mathbf{x}_{k-1} - \mathbf{v}_k)^T \mathbf{Q}_k^{-1}(\mathbf{x}_k - \mathbf{A}_{k-1}\mathbf{x}_{k-1} - \mathbf{v}_k), \ k = 1 ... K \end{cases} \tag{15}$$

$$J_{y,k}(\mathbf{x}) = \frac{1}{2}(\mathbf{y}_k - \mathbf{C}_k\mathbf{x}_k)^T \mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{C}_k\mathbf{x}_k), \ k = 0 ... K \tag{16}$$

which are all **squared Mahalanobis distances**.

We then define an overall **overall objective function**, $J(\mathbf{x})$, that we will seek to minimize wrt the *design parameter*, **x**

$$J(\mathbf{x}) = \sum_{k=0}^{K} (J_{\upsilon,k}(\mathbf{x}) + J_{y,k}(\mathbf{x})) \tag{17}$$

Maximum A Posteriori (MAP)

We seek to solve the following problem

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} J(\mathbf{x}) \tag{18}$$

which results in the same solution for the best estimate, $\hat{\mathbf{x}}$, as Eq. (3). In other words, we are still finding the optimal estimate that maximizes the likelihood of the state given the data. This is an **unconstrained optimization problem**.

We stack all the known data into a lifted column, $\mathbf{z}$ and recall that $\mathbf{x}$ is also a tall column consisting of all the states

$$\mathbf{z} = \begin{bmatrix} \check{\mathbf{x}}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \\ \hline \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{bmatrix} \tag{19}$$

Maximum A Posteriori (MAP)

We then define the block-matrix quantities

$$\mathbf{H} = \left[\begin{array}{ccccc} \mathbf{1} & & & & \\ -\mathbf{A}_0 & \mathbf{1} & & & \\ & \ddots & \ddots & & \\ & & -\mathbf{A}_{K-1} & \mathbf{1} \\ \hline \mathbf{C}_0 & & & & \\ & \mathbf{C}_1 & & & \\ & & \ddots & & \\ & & & & \mathbf{C}_K \end{array}\right], \quad \mathbf{W} = \left[\begin{array}{cccc|cccc} \check{\mathbf{P}}_0 & & & & & & & \\ & \mathbf{Q}_1 & & & & & & \\ & & \ddots & & & & & \\ & & & \mathbf{Q}_K & & & & \\ \hline & & & & \mathbf{R}_0 & & & \\ & & & & & \mathbf{R}_1 & & \\ & & & & & & \ddots & \\ & & & & & & & \mathbf{R}_K \end{array}\right] \tag{20}$$

where only non-zero blocks are shown.

Maximum A Posteriori (MAP)

Under these definitions, we find

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{z} - \mathbf{H}\mathbf{x})^T \mathbf{W}^{-1} (\mathbf{z} - \mathbf{H}\mathbf{x}) \tag{21}$$

which is exactly quadratic in $\mathbf{x}$. We also have

$$p(\mathbf{z}|\mathbf{x}) = \eta \exp\left(-\frac{1}{2}(\mathbf{z} - \mathbf{H}\mathbf{x})^T \mathbf{W}^{-1} (\mathbf{z} - \mathbf{H}\mathbf{x})\right) \tag{22}$$

where η is a normalization constant.

Since $J(\mathbf{x})$ is a paraboloid, we can find its minimum in closed form by setting the partial derivative wrt $\mathbf{x}$ to zero

$$\left.\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}^T}\right|_{\hat{\mathbf{x}}} = -\mathbf{H}^T \mathbf{W}^{-1} (\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) = 0 \Rightarrow (\mathbf{H}^T \mathbf{W}^{-1} \mathbf{H})\hat{\mathbf{x}} = \mathbf{H}^T \mathbf{W}^{-1} \mathbf{z} \tag{23}$$

Maximum A Posteriori (MAP)

The solution to Eq. (23) *(repeated here for convenience)*, $\hat{\mathbf{x}}$, is the classic **batch least-squares** solution aid it is equivalent to the **fixed-internal smoother**.

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

Pseudoinverse and sparse-equation solvers

The *batch least-squares* solution employes the **pseudoinverse**. Computationally, to solve this system of linear equations, we would never actually invert $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}$. This matrix presents a special *block-tridiagonal* structure and thus a sparse-equation solver can be exploited.

Bayesian Inference

Having looked at the optimization approach to batch LG estimation, we now look at the full Bayesian posterior, $p(\mathbf{x}|\mathbf{v}, \mathbf{y})$ and not only the maximum. This requires that we begin with a **prior density** over the states, which we will then update based on the measurements.

Bayesian Inference

Having looked at the optimization approach to batch LG estimation, we now look at the full Bayesian posterior, $p(\mathbf{x}|\mathbf{v}, \mathbf{y})$ and not only the maximum. This requires that we begin with a **prior density** over the states, which we will then update based on the measurements.

A prior can be built using knowledge of the initial state, as well as the inputs to the system: $p(\mathbf{x}|\mathbf{v})$. We will use just the motion model to derive this prior

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k \tag{24}$$

In *lifted matrix form*, we write

$$\mathbf{x} = \mathbf{A}(\mathbf{v} + \mathbf{w}) \tag{25}$$

where $\mathbf{w}$ is the lifted form of the initial state and process noise, and

$$\mathbf{A} = \begin{bmatrix} \mathbf{1} & & & & & \\ \mathbf{A}_0 & \mathbf{1} & & & & \\ \mathbf{A}_1\mathbf{A}_0 & \mathbf{A}_1 & \mathbf{1} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \mathbf{A}_{K-2}\cdots\mathbf{A}_0 & \mathbf{A}_{K-2}\cdots\mathbf{A}_1 & \mathbf{A}_{K-2}\cdots\mathbf{A}_2 & \cdots & \mathbf{1} & \\ \mathbf{A}_{K-1}\cdots\mathbf{A}_0 & \mathbf{A}_{K-1}\cdots\mathbf{A}_1 & \mathbf{A}_{K-1}\cdots\mathbf{A}_2 & \cdots & \mathbf{A}_{K-1} & \mathbf{1} \end{bmatrix} \tag{26}$$

is the lifted transition matrix and presents lower-triangular structure.

The lifted mean and covariance take the form

$$\text{lifted mean:} \quad \check{\mathbf{x}} = E[\mathbf{x}] = E[\mathbf{A}(\mathbf{v} + \mathbf{w})] = \mathbf{A}\mathbf{v} \tag{27}$$

$$\text{lifted covariance:} \quad \check{\mathbf{P}} = E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T] = \mathbf{A}\mathbf{Q}\mathbf{A}^T \tag{28}$$

where $\mathbf{Q} = E[\mathbf{w}\mathbf{w}^T] = \text{diag}(\check{\mathbf{P}}, \mathbf{Q}_1, ..., \mathbf{Q}_K)$. The prior can then be expressed as

$$\text{Prior:} \quad p(\mathbf{x}|\mathbf{v}) = \mathcal{N}(\check{\mathbf{x}}, \check{\mathbf{P}}) = \mathcal{N}(\mathbf{A}\mathbf{v}, \mathbf{A}\mathbf{Q}\mathbf{A}^T) \tag{29}$$

Turning to the measurements, we write for the measurement model

$$\mathbf{y}_k = \mathbf{C}_k \mathbf{x}_k + \mathbf{n}_k \tag{30}$$

and accordingly in lifted form

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{n}, \quad \mathbf{C} = \text{diag}(\mathbf{C}_0, \mathbf{C}_1, ..., \mathbf{C}_K) \tag{31}$$

where $\mathbf{n}$ is the lifted form of the measurement noise.

The joint density of the prior lifted state and the measurements can then be written as

$$p(\mathbf{x},\mathbf{y}|\mathbf{v}) = \mathcal{N}\left(\begin{bmatrix}\check{\mathbf{x}}\\ \mathbf{C}\check{\mathbf{x}}\end{bmatrix}, \begin{bmatrix}\check{\mathbf{P}} & \check{\mathbf{P}}\mathbf{C}^T\\ \mathbf{C}\check{\mathbf{P}} & \mathbf{C}\check{\mathbf{P}}\mathbf{C}^T + \mathbf{R}\end{bmatrix}\right) \tag{32}$$

where $\mathbf{R} = E[\mathbf{n}\mathbf{n}^T] = \mathrm{diag}(\mathbf{R}_0, \mathbf{R}_1, ..., \mathbf{R}_K)$. This can be factored based on $p(\mathbf{x},\mathbf{y}|\mathbf{v}) = p(\mathbf{x}|\mathbf{v},\mathbf{y})p(\mathbf{y},\mathbf{v})$. We care for the first factor, the **Bayesian posterior**, which we write

$$p(\mathbf{x}|\mathbf{v},\mathbf{y}) = \mathcal{N}\left(\check{\mathbf{x}} + \check{\mathbf{P}}\mathbf{C}^T(\mathbf{C}\check{\mathbf{P}}\mathbf{C}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathbf{C}\check{\mathbf{x}}), \check{\mathbf{P}} - \check{\mathbf{P}}(\mathbf{C}\check{\mathbf{P}}\mathbf{C}^T + \mathbf{R})^{-1}\mathbf{C}\check{\mathbf{P}}\right) \tag{33}$$

Using the SMW identities we can rewrite for the **Bayesian solution**

$$p(\mathbf{x}|\mathbf{v},\mathbf{y}) = \mathcal{N}\left(\underbrace{(\check{\mathbf{P}}^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C})^{-1}(\check{\mathbf{P}}^{-1}\check{\mathbf{x}} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y})}_{\hat{\mathbf{x}},\ \text{mean}}, \underbrace{(\check{\mathbf{P}}^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C})^{-1}}_{\hat{\mathbf{P}},\ \text{covariance}}\right) \tag{34}$$

To see the connection to the optimization approach we rearrange the mean expression to arrive at a linear system for $\hat{\mathbf{x}}$

$$\underbrace{\left(\check{\mathbf{P}}^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)}_{\hat{\mathbf{P}}^{-1}}\hat{\mathbf{x}} = \check{\mathbf{P}}^{-1}\check{\mathbf{x}} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y} \tag{35}$$

and see that the inverse covariance appears on the left-hand side. Substituting in $\check{\mathbf{x}} = \mathbf{A}\mathbf{v}$ and $\check{\mathbf{P}}^{-1} = (\mathbf{A}\mathbf{Q}\mathbf{A}^T)^{-1} = \mathbf{A}^{-T}\mathbf{Q}^{-1}\mathbf{A}^{-1}$ we rewrite as

$$\underbrace{\left(\mathbf{A}^{-T}\mathbf{Q}^{-1}\mathbf{A}^{-1} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}\right)}_{\hat{\mathbf{P}}^{-1}}\hat{\mathbf{x}} = \mathbf{A}^{-T}\mathbf{Q}^{-1}\mathbf{v} + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y} \tag{36}$$

Bayesian Inference

In turn, we see that this requires computing $\mathbf{A}^{-1}$ which however has a neat *lower-triangular and very sparse structure*

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{1} & & & & & \\ -\mathbf{A}_0 & \mathbf{1} & & & & \\ & -\mathbf{A}_1 & \mathbf{1} & & & \\ & & -\mathbf{A}_2 & \ddots & & \\ & & & \ddots & \mathbf{1} & \\ & & & & -\mathbf{A}_{K-1} & \mathbf{1} \end{bmatrix} \tag{37}$$

If we define

$$\mathbf{z} = \begin{bmatrix} \mathbf{v} \\ \mathbf{y} \end{bmatrix}, \ \mathbf{H} = \begin{bmatrix} \mathbf{A}^{-1} \\ \mathbf{C} \end{bmatrix}, \ \mathbf{W} = \begin{bmatrix} \mathbf{Q} & \\ & \mathbf{R} \end{bmatrix} \tag{38}$$

we then can rewrite our system of equations as

$$(\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H})\hat{\mathbf{x}} = \mathbf{H}^T\mathbf{W}^{-1}\mathbf{z} \tag{39}$$

which is **identical to the optimization-based solution discussed earlier**.

Existence, Uniqueness, and Observability

Most LG estimation results can be viewed as a special case of Eq. (39). Examining the equation it holds that $\hat{\mathbf{x}}$ will **exist** and be a **unique solution** if and only if $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}$ is **invertible**. Then

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}^{-1}\mathbf{z} \tag{40}$$

For $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}$ to be invertible it must hold that $\mathrm{rank}(\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}) = N(K+1)$ because $\dim \mathbf{x} = N(K+1)$. Since $\mathbf{W}^{-1}$ is real symmetric positive definite, the uniqueness requirement becomes

$$\mathrm{rank}(\mathbf{H}^T\mathbf{H}) = \mathrm{rank}(\mathbf{H}^T) = N(K+1) \tag{41}$$

or in other words, we need $N(K+1)$ linearly independent rows (or columns) in $\mathbf{H}^T$.

We can have two cases for our problem, namely to (i) have a good prior $\check{\mathbf{x}}_0$, or (ii) to note have such knowledge of the initial state.

For the case of knowledge of the initial state, we write out $\mathbf{H}^T$ and our rank test takes the form

$$\operatorname{rank}(\mathbf{H}^T) = \operatorname{rank} \left[\begin{array}{ccccc|ccccc} \mathbf{1} & -\mathbf{A}_0^T & & & & \mathbf{C}_0^T & & & & \\ & \mathbf{1} & -\mathbf{A}_1^T & & & & \mathbf{C}_1^T & & & \\ & & \mathbf{1} & \ddots & & & & \mathbf{C}_2^T & & \\ & & & \ddots & -\mathbf{A}_{K-1}^T & & & & \ddots & \\ & & & & \mathbf{1} & & & & & \mathbf{C}_K^T \end{array} \right] \tag{42}$$

which is *row-echelon form*. This entails that the matrix is full rank, $N(K+1)$, since all the block-rows are linearly indepdendent. Accordingly, we always have a unique solution for $\hat{\mathbf{x}}$ as long as

$$\hat{\mathbf{P}}_0 \succ 0, \quad \mathbf{Q}_k \succ 0 \tag{43}$$

where $\succ$ means that the matrix is positive-definite and thus invertible.

Existence, Uniqueness, and Observability

Row-echelon form

A matrix being in row-echelon form means that Gaussian elimination has operated on the rows, and column-echelon form means that Gaussian elimination has operated on the columns. A matrix is in column-echelon form if its transpose is in row-echelon form. The test for a matrix being in row-echelon form is

- All rows consisting of only zeroes are at the bottom.
- The leading coefficient (or *pivot*) of a nonzero row is always strictly to the right of the leading coefficient of the row above it.

The additional test rule of the leading coefficient being 1 may be added.
The conditions imply that all entries in a column below a leading coefficient are zeros.
The row-echelon form of a matrix is - generally - not unique.

Existence, Uniqueness, and Observability

Reduced row-echelon form

A matrix is in reduced row-echelon form if it satisfies the following conditions

- It is in row-echelon form.
- The leading entry in each nonzero is a 1
- Each column containing a leading 1 has zeros in all its other entries.

The reduced row-echelon form of a matrix can be computed via *Gauss-Jordan elimination*.
The reduced row-echelon form of a matrix is *unique* and does not depend on the algorithm used to compute it.

For the case of not having any knowledge of the initial state, it can be shown that the requirement for solution uniqueness is that the matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{(N-1)} \end{bmatrix} \tag{44}$$

has rank N, i.e.,

$$\text{rank}\mathcal{O} = N \tag{45}$$

Which in turn is the same requirement as that for observability in LTI systems. Thus, we can see the connection between observability and invertibility of $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}$.

Overall the conditions for solution existence and uniqueness in Eq. (39) are

$$\mathbf{Q}_k \succ 0, \ \mathbf{R}_k \succ 0, \ \ \text{rank}\mathcal{O} = N \tag{46}$$

Note: These are sufficient but not necessary conditions.

MAP Covariance

One important question is how confident are we for $\hat{\mathbf{x}}$ in Eq. (39). It turns out that we can re-interpret the least-squares solution as a Gaussian estimate for $\mathbf{x}$

$$\underbrace{(\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H})}_{\substack{\text{inverse}\\ \text{covariance}}}\underbrace{\hat{\mathbf{x}}}_{\text{mean}} = \underbrace{\mathbf{H}^T\mathbf{W}^{-1}\mathbf{z}}_{\substack{\text{information}\\ \text{vector}}} \tag{47}$$

where the right-hand side is referred to as the *information vector*. To understand further, let us employ Bayes' rule to rewrite Eq. (22) as

$$p(\mathbf{x}|\mathbf{z}) = \beta \exp\left(-\frac{1}{2}(\mathbf{H}\mathbf{x}-\mathbf{z})^T\mathbf{W}^{-1}(\mathbf{H}\mathbf{x}-\mathbf{z})\right) \tag{48}$$

where β is a new normalization constant. We then substitute Eq. (39) and find that

$$p(\mathbf{x}|\hat{\mathbf{x}}) = \kappa \exp\left(-\frac{1}{2}(\mathbf{x}-\hat{\mathbf{x}})^T(\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H})(\mathbf{x}-\hat{\mathbf{x}})\right) \tag{49}$$

where κ a new normalization constant.

MAP Covariance

Accordingly, we see that $\mathcal{N}(\hat{\mathbf{x}}, \hat{\mathbf{P}})$ is a Gaussian estimator for $\mathbf{x}$ whose mean is the optimization solution and whose covariance is $\hat{\mathbf{P}} = (\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H})^{-1}$.

Table of Contents

Recursive Discrete-Time Smoothing

- The batch solution is appealing in that it is easy to understand, yet brute-force solving is not realistic for many situations.
- Since the inverse covariance matrix on the left side is sparse, we can solve the system of equations in an efficient manner employing a forward recursion followed by a backward recursion.
- When the equations are solved this manner, the method is referred to as a *fixed-interval smoother*.
- Smoothers implement efficiently the full batch solution, with no approximation.
- We shall discuss the *Cholesky smoother* and the *Rauch-Tung-Striebel smoother*.

Exploiting Sparsity in the Batch Solution

Consider

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

The left hand side of Eq. (39) (repeated above for convenience), $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}$, is block-tridiagonal (under our chronological variable ordering for $\mathbf{x}$)

$$\begin{bmatrix} \star & \star & & & & \\ \star & \star & \star & & & \\ & \star & \star & \star & & \\ & & \ddots & \ddots & \ddots & \\ & & & \star & \star & \star \\ & & & & \star & \star \end{bmatrix} \tag{50}$$

where $\star$ indicates a non-zero block. Importantly, there exist solvers that can *exploit this sparse structure* and thus solve for $\hat{\mathbf{x}}$ efficiently.

Exploiting Sparsity in the Batch Solution

Such batch equations can be solved through a sparse *Cholesky decomposition* followed by forward and backward passes. We can efficiently factor $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}$ into

$$\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H} = \mathbf{L}\mathbf{L}^T \tag{51}$$

where $\mathbf{L}$ is called the Cholesky factor and is a block-lower-triangular matrix with form

$$\mathbf{L} = \begin{bmatrix} \star & & & & \\ \star & \star & & & \\ & \star & \star & & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ & & & & \star & \star \end{bmatrix} \tag{52}$$

and the decomposition can be computed in $O(N(K+1))$ time.

Exploiting Sparsity in the Batch Solution

Next we solve for $\mathbf{d}$ in

$$\mathbf{L}\mathbf{d} = \mathbf{H}^T\mathbf{W}^{-1}\mathbf{z} \tag{53}$$

This is called the **forward pass** and is done again in $O(N(K+1))$ through forward substitution owing to the sparse lower-triangular form of $\mathbf{L}$.

Finally, we solve for $\hat{\mathbf{x}}$ in

$$\mathbf{L}^T\hat{\mathbf{x}} = \mathbf{d} \tag{54}$$

This is called the **backward pass** and is done again in $O(N(K+1))$ through backward substitution owing to the sparse upper-triangular form of $\mathbf{L}^T$.

Overall, the batch equations can be solved in computational time that scales linearly with the size of the state.

Exploiting Sparsity in the Batch Solution

Cholesky Decomposition

The Cholesky decomposition of a Hermitian positive-definite matrix $\mathbf{A}$ is a decomposition of the form

$$\mathbf{A} = \mathbf{L}\mathbf{L}^{\star} \tag{55}$$

where $\mathbf{L}$ is a lower-diagonal matrix with ral and positive diagonal entries, and $\mathbf{L}^{\star}$ is its conjugate transpose. Every Hermitian positive-definite matrix (and thus also every real-valued symmetric positive-definite matrix) has a unique Cholesky decomposition.

When $\mathbf{A}$ is a real matrix (thus symmetric positive-definite), we write

$$\mathbf{A} = \mathbf{L}\mathbf{L}^{T} \tag{56}$$

where $\mathbf{L}$ is a real lower-triangular matrix with positive diagonal entries.

The Cholesky solution to the batch estimation problem, involving a set of forward-backward recursions, is called the *Cholesky smoother*.

Let $\mathbf{L}$ which takes the form

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_0 & & & & & \\ \mathbf{L}_{10} & \mathbf{L}_1 & & & & \\ & \mathbf{L}_{21} & \mathbf{L}_2 & & & \\ & & \ddots & \ddots & & \\ & & & \mathbf{L}_{K-1,K-2} & \mathbf{L}_{K-1} & \\ & & & & \mathbf{L}_{K,K-1} & \mathbf{L}_K \end{bmatrix} \tag{57}$$

Cholesky Smoother

Using the definitions of $\mathbf{H}, \mathbf{W}$, when we multiply out $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H} = \mathbf{L}\mathbf{L}^T$ and compare at the block level we have

$$\mathbf{L}_0\mathbf{L}_0^T = \underbrace{\check{\mathbf{P}}_0^{-1} + \mathbf{C}_0^T\mathbf{R}_0^{-1}\mathbf{C}_0}_{\mathbf{I}_0} + \mathbf{A}_0^T\mathbf{Q}_1^{-1}\mathbf{A}_0 \tag{58}$$

$$\mathbf{L}_{10}\mathbf{L}_0^T = -\mathbf{Q}_1^{-1}\mathbf{A}_0 \tag{59}$$

$$\mathbf{L}_1\mathbf{L}_1^T = \underbrace{-\mathbf{L}_{10}\mathbf{L}_{10}^T + \mathbf{Q}_1^{-1} + \mathbf{C}_1^T\mathbf{R}_1^{-1}\mathbf{C}_1}_{\mathbf{I}_1} + \mathbf{A}_1^T\mathbf{Q}_2^{-1}\mathbf{A}_1 \tag{60}$$

$$\mathbf{L}_{21}\mathbf{L}_1^T = -\mathbf{Q}_2^{-1}\mathbf{A}_1 \tag{61}$$

$$\vdots$$

$$\mathbf{L}_{K-1}\mathbf{L}_{K-1}^T = \underbrace{-\mathbf{L}_{K-1,K-2}\mathbf{L}_{K-1,K-2}^T + \mathbf{Q}_{K-1}^{-1} + \mathbf{C}_{K-1}^T\mathbf{R}_{K-1}^{-1}\mathbf{C}_{K-1}}_{\mathbf{I}_{K-1}} + \mathbf{A}_{K-1}^T\mathbf{Q}_K^{-1}\mathbf{A}_{K-1} \tag{62}$$

$$\mathbf{L}_{K,K-1}\mathbf{L}_{K-1}^T = -\mathbf{Q}_K^{-1}\mathbf{A}_{K-1} \tag{63}$$

$$\mathbf{L}_K\mathbf{L}_K^T = \underbrace{-\mathbf{L}_{K,K-1}\mathbf{L}_{K,K-1}^T + \mathbf{Q}_K^{-1} + \mathbf{C}_K^T\mathbf{R}_K^{-1}\mathbf{C}_K}_{\mathbf{I}_K} \tag{64}$$

where the underbraces allow us to define the $\mathbf{I}_k$ quantities.

Cholesky Smoother

From the above equations, we can first solve for $\mathbf{L}_0$ by doing a small (dense) Cholesky decomposition in the first equation, then substitute into the second to solve for $\mathbf{L}_{10}$ - and so on up to $\mathbf{L}_K$ thus confirming that we can work out all the blocks of $\mathbf{L}$ in a single forward pass with $O(N(K+1))$ complexity.

Next we solve $\mathbf{L}\mathbf{d} = \mathbf{H}^T\mathbf{W}^{-1}\mathbf{z}$ for $\mathbf{d}$, where

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_K \end{bmatrix} \tag{65}$$

Cholesky Smoother

Multiplying out and comparing at the block level

$$\mathbf{L}_0\mathbf{d}_0 = \underbrace{\check{\mathbf{P}}_0^{-1}\check{\mathbf{x}}_0 + \mathbf{C}_0^T\mathbf{R}_0^{-1}\mathbf{y}_0}_{\mathbf{q}_0} - \mathbf{A}_0^T\mathbf{Q}_1^{-1}\mathbf{v}_1 \quad (66)$$

$$\mathbf{L}_1\mathbf{d}_1 = \underbrace{-\mathbf{L}_{10}\mathbf{d}_0 + \mathbf{Q}_1^{-1}\mathbf{v}_1 + \mathbf{C}_1^T\mathbf{R}_1^{-1}\mathbf{y}_1}_{\mathbf{q}_1} - \mathbf{A}_1^T\mathbf{Q}_2^{-1}\mathbf{v}_2 \quad (67)$$

$$\vdots$$

$$\mathbf{L}_{K-1}\mathbf{d}_{K-1} = \underbrace{-\mathbf{L}_{K-1,K-2}\mathbf{d}_{K-2} + \mathbf{Q}_{K-1}^{-1}\mathbf{v}_{K-1} + \mathbf{C}_{K-1}^T\mathbf{R}_{K-1}^{-1}\mathbf{y}_{K-1}}_{\mathbf{q}_{K-1}} - \mathbf{A}_{K-1}^T\mathbf{Q}_K^{-1}\mathbf{v}_K \quad (68)$$

$$\mathbf{L}_K\mathbf{d}_K = \underbrace{-\mathbf{L}_{K,K-1}\mathbf{d}_{K-1} + \mathbf{Q}_K^{-1}\mathbf{v}_K + \mathbf{C}_K^T\mathbf{R}_K^{-1}\mathbf{y}_K}_{\mathbf{q}_K} \quad (69)$$

where the underbraces allow to define the $\mathbf{q}_k$ terms.

Cholesky Smoother

In the above equations we can solve for $\mathbf{d}_0$, then for $\mathbf{d}_1$ and so on until $\mathbf{d}_K$ which confirms that we can wok out all the blocks of $\mathbf{d}$ in a single forward pass in $O(N(K+1))$.

The last step is to solve $\mathbf{L}^T\hat{\mathbf{x}} = \mathbf{d}$ for $\hat{\mathbf{x}}$, where

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_0 \\ \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_K \end{bmatrix} \tag{70}$$

Cholesky Smoother

Multiplying out and comparing at the block level, we have

$$\mathbf{L}_K^T \hat{\mathbf{x}}_K = \mathbf{d}_K \tag{71}$$

$$\mathbf{L}_{K-1}^T \hat{\mathbf{x}}_{K-1} = -\mathbf{L}_{K,K-1}^T \hat{\mathbf{x}}_K + \mathbf{d}_{K-1} \tag{72}$$

$$\vdots$$

$$\mathbf{L}_1^T \hat{\mathbf{x}}_1 = -\mathbf{L}_{21}^T \hat{\mathbf{x}}_2 + \mathbf{d}_1 \tag{73}$$

$$\mathbf{L}_0^T \hat{\mathbf{x}}_0 = -\mathbf{L}_{10}^T \hat{\mathbf{x}}_1 + \mathbf{d}_0 \tag{74}$$

From these equations, we can solve for $\hat{\mathbf{x}}_K$ in the first equation, then substitute to derive $\hat{\mathbf{x}}_{K-1}$ and so on up to $\hat{\mathbf{x}}_0$. This confirms that we can work out all the blocks of $\hat{\mathbf{x}}$ in a single backward pass in $O(N(K+1))$ time.

Cholesky Smoother

In terms of the $\mathbf{I}_k, \mathbf{q}_k$ quantities, we can combine the two forward passes (solving for $\mathbf{L}, \mathbf{d}$) and also write the backwards pass as

forward:
$k = 1 \ldots K$

$$\mathbf{L}_{k-1}\mathbf{L}_{k-1}^T = \mathbf{I}_{k-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \tag{75}$$

$$\mathbf{L}_{k-1}\mathbf{d}_{k-1} = \mathbf{q}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k \tag{76}$$

$$\mathbf{L}_{k,k-1}\mathbf{L}_{k-1}^T = -\mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \tag{77}$$

$$\mathbf{I}_k = -\mathbf{L}_{k,k-1}\mathbf{L}_{k,k-1}^T + \mathbf{Q}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \tag{78}$$

$$\mathbf{q}_k = -\mathbf{L}_{k,k-1}\mathbf{d}_{k-1} + \mathbf{Q}_k^{-1}\mathbf{v}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \tag{79}$$

backward:
$k = K \ldots 1$

$$\mathbf{L}_{k-1}^T \hat{\mathbf{x}}_{k-1} = -\mathbf{L}_{k,k-1}^T \hat{\mathbf{x}}_k + \mathbf{d}_{k-1} \tag{80}$$

which are initialized with

$$\mathbf{I}_0 = \check{\mathbf{P}}_0^{-1} + \mathbf{C}_0^T \mathbf{R}_0^{-1} \mathbf{C}_0 \tag{81}$$

$$\mathbf{q}_0 = \check{\mathbf{P}}_p^{-1} \check{\mathbf{x}}_0 + \mathbf{C}_0^T \mathbf{R}_0^{-1} \mathbf{y}_0 \tag{82}$$

$$\hat{\mathbf{x}}_K = \mathbf{L}_K^{-T} \mathbf{d}_K \tag{83}$$

The forward pass maps $\{\mathbf{q}_{k-1}, \mathbf{I}_{k-1}\}$ to the same pair at the next time, $\{\mathbf{q}_k, \mathbf{I}_k\}$. The backward pass maps $\hat{\mathbf{x}}_k$ to the same quantity at the previous timestep, $\hat{\mathbf{x}}_{k-1}$. In the process, we solve for all blocks of $\mathbf{L}, \mathbf{d}$. *The only linear algebra operations required the Cholesky smoother are Cholesky decomposition, multiplication, addition, and solving a linear system via forward/backward substitution.*

Table of Contents

Recursive Discrete-Time Filtering

The batch solution - and the corresponding smoother implementations - give us the best solution but cannot be used online. For an online solution, the estimate of the current state can only employ data up to the current timestep. The classical solution to this problem is the *Kalman Filter*.

$$\overbrace{\underbrace{\check{\mathbf{x}}_0, \mathbf{y}_0, \mathbf{v}_1, \mathbf{y}_1, \mathbf{v}_2, \mathbf{y}_2, \ldots, \mathbf{v}_{k-1}, \mathbf{y}_{k-1}, \mathbf{v}_k, \mathbf{y}_k,}_{\hat{\mathbf{x}}_{k,f}} \mathbf{v}_{k+1}, \mathbf{y}_{k+1}, \ldots, \mathbf{v}_K, \mathbf{y}_K}^{\hat{\mathbf{x}}_k} \tag{84}$$

Recursive Discrete-Time Filtering

To prepare things, we reorder some of the variables from the batch solution and redefine $\mathbf{z}, \mathbf{H}, \mathbf{W}$ as

$$
\mathbf{z} = \left[\begin{array}{c} \check{\mathbf{x}}_0 \\ \mathbf{y}_0 \\ \hline \mathbf{v}_1 \\ \mathbf{y}_1 \\ \hline \mathbf{v}_2 \\ \mathbf{y}_2 \\ \hline \vdots \\ \hline \mathbf{v}_K \\ \mathbf{y}_K \end{array}\right], \quad
\mathbf{H} = \left[\begin{array}{ccccc} \mathbf{1} & & & & \\ \mathbf{C}_0 & & & & \\ \hline -\mathbf{A}_0 & \mathbf{1} & & & \\ & \mathbf{C}_1 & & & \\ \hline & -\mathbf{A}_1 & \mathbf{1} & & \\ & & \mathbf{C}_2 & & \\ \hline & & \ddots & \ddots & \\ \hline & & & -\mathbf{A}_{K-1} & \mathbf{1} \\ & & & & \mathbf{C}_K \end{array}\right], \quad
\mathbf{W} = \left[\begin{array}{cc|cc|cc|c|cc} \check{\mathbf{P}}_0 & & & & & & & & \\ & \mathbf{R}_0 & & & & & & & \\ \hline & & \mathbf{Q}_1 & & & & & & \\ & & & \mathbf{R}_1 & & & & & \\ \hline & & & & \mathbf{Q}_2 & & & & \\ & & & & & \mathbf{R}_2 & & & \\ \hline & & & & & & \ddots & & \\ \hline & & & & & & & \mathbf{Q}_K & \\ & & & & & & & & \mathbf{R}_K \end{array}\right] \tag{85}
$$

where the partition lines now show divisions between timesteps. As this reordering does not change the order in $\mathbf{x}$, the matrix $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}$ is still block-tridiagonal.

Recursive Discrete-Time Filtering

Let us consider the factorization at the probability level. Considering the previous derivations on $p(\mathbf{x}|\mathbf{v},\mathbf{y})$ and if we want to consider only the state at time k, we can marginalize out the other states by integrating over all possible values

$$p(\mathbf{x}_k|\mathbf{v},\mathbf{y}) = \int_{\mathbf{x}_i,\forall i\neq k} p(\mathbf{x}_0,\ldots,\mathbf{x}_K|\mathbf{v},\mathbf{y})d\mathbf{x}_{i,\forall i\neq k} \tag{86}$$

It turns out that **we can factor this probability density into two parts**

$$p(\mathbf{x}_k|\mathbf{v},\mathbf{y}) = \eta p(\mathbf{x}_k|\check{\mathbf{x}}_0,\mathbf{v}_{1:k},\mathbf{y}_{0:k})p(\mathbf{x}_k|\mathbf{v}_{k+1:K},\mathbf{y}_{k+1:K}) \tag{87}$$

where η is a normalization constant.

- **We can take our batch solution and factor it into the normalized product of two Gaussian PDFs.**

Recursive Discrete-Time Filtering

To carry out this factorization we exploit the sparse structure in $\mathbf{H}$ and begin by partitioning it into 12 blocks, of which 6 are non-zero

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & & \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \\ & \mathbf{H}_{32} & \mathbf{H}_{33} \\ & & \mathbf{H}_{43} \end{bmatrix}, \quad \begin{matrix} \text{information from } 0 \ldots k-1 \\ \text{information from } k \\ \text{information from } k+1 \\ \text{information from } k+2 \ldots K \end{matrix} \tag{88}$$

whereas the first column refers to states from $0 \ldots k-1$, the second to states from k, and the third to states from $k+1 \ldots K$.

Recursive Discrete-Time Filtering

For example, for $k = 2$, $K = 4$ the partitions of $\mathbf{H}$ and compatible $\mathbf{z}, \mathbf{W}$ partitions are

$$\mathbf{H} = \left[\begin{array}{cc|c|c|c} \mathbf{1} & & & & \\ \mathbf{C}_0 & & & & \\ -\mathbf{A}_0 & \mathbf{1} & & & \\ & \mathbf{C}_1 & & & \\ \hline & -\mathbf{A}_1 & \mathbf{1} & & \\ & & \mathbf{C}_2 & & \\ \hline & & -\mathbf{A}_2 & \mathbf{1} & \\ & & & \mathbf{C}_3 & \\ \hline & & & -\mathbf{A}_3 & \mathbf{1} \\ & & & & \mathbf{C}_4 \end{array}\right], \quad \mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & & & \\ & \mathbf{W}_2 & & \\ & & \mathbf{W}_3 & \\ & & & \mathbf{W}_4 \end{bmatrix} \tag{89}$$

Recursive Discrete-Time Filtering

For $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}$ we then have

$$\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11}^T\mathbf{W}_1^{-1}\mathbf{H}_{11} + \mathbf{H}_{21}^T\mathbf{W}_2^{-1}\mathbf{H}_{21} & \mathbf{H}_{21}^T\mathbf{W}_2^{-1}\mathbf{H}_{22} & \\ \mathbf{H}_{22}^T\mathbf{W}_2^{-1}\mathbf{H}_{21} & \mathbf{H}_{22}^T\mathbf{W}_2^{-1}\mathbf{H}_{22} + \mathbf{H}_{32}^T\mathbf{W}_3^{-1}\mathbf{H}_{32} & \mathbf{H}_{32}^T\mathbf{W}_3^{-1}\mathbf{H}_{33} \\ & \mathbf{H}_{33}^T\mathbf{W}_3^{-1}\mathbf{H}_{32} & \mathbf{H}_{33}^T\mathbf{W}_3^{-1}\mathbf{H}_{33} + \mathbf{H}_{43}^T\mathbf{W}_4^{-1}\mathbf{H}_{43} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} & \\ \mathbf{L}_{12}^T & \mathbf{L}_{22} & \mathbf{L}_{32}^T \\ & \mathbf{L}_{32} & \mathbf{L}_{33} \end{bmatrix} \tag{90}$$

where the intermediate variables $\mathbf{L}_{ij}$ are used to assign the blocks of the matrix.

Recursive Discrete-Time Filtering

For $\mathbf{H}^T\mathbf{W}^{-1}\mathbf{z}$ we have

$$\mathbf{H}^T\mathbf{W}^{-1}\mathbf{z} = \begin{bmatrix} \mathbf{H}_{11}^T\mathbf{W}_1^{-1}\mathbf{z}_1 + \mathbf{H}_{21}^T\mathbf{W}_2^{-1}\mathbf{z}_2 \\ \mathbf{H}_{22}^T\mathbf{W}_2^{-1}\mathbf{z}_2 + \mathbf{H}_{32}^T\mathbf{W}_3^{-1}\mathbf{z}_3 \\ \mathbf{H}_{33}^T\mathbf{W}_3^{-1}\mathbf{z}_3 + \mathbf{H}_{43}^T\mathbf{W}_4^{-1}\mathbf{z}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix} \quad (91)$$

where again we have assigned the blocks to some intermediate variables $\mathbf{r}_i$.

Next we **partition the states**, $\mathbf{x}$ as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{0:k-1} \\ \mathbf{x}_k \\ \mathbf{x}_{k+1:K} \end{bmatrix} \begin{matrix} \text{states from } 0 \ldots k-1 \\ \text{states from } k \\ \text{states from } k+1 \ldots K \end{matrix} \quad (92)$$

The overall batch system of equations now takes the form

$$\begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} & \\ \mathbf{L}_{12}^T & \mathbf{L}_{22} & \mathbf{L}_{32}^T \\ & \mathbf{L}_{32} & \mathbf{L}_{33} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{0:k-1} \\ \hat{\mathbf{x}}_k \\ \hat{\mathbf{x}}_{k+1:K} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{bmatrix} \tag{93}$$

where $(\hat{\cdot})$ is added to indicate that this is the solution to the optimization problem discussed earlier. We proceed with steps towards a recursive LG estimator to solve $\hat{\mathbf{x}}_k$.

To isolate $\hat{\mathbf{x}}_k$ we left multiply both sides of the above equation by

$$\begin{bmatrix} \mathbf{1} & & \\ -\mathbf{L}_{12}^T\mathbf{L}_{11}^{-1} & \mathbf{1} & -\mathbf{L}_{32}^T\mathbf{L}_{33}^{-1} \\ & & \mathbf{1} \end{bmatrix} \tag{94}$$

Recursive Discrete-Time Filtering

The resulting system of equations is

$$\begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} & \\ & \mathbf{L}_{22} - \mathbf{L}_{12}^T \mathbf{L}_{11}^{-1} \mathbf{L}_{12} - \mathbf{L}_{32}^T \mathbf{L}_{33}^{-1} \mathbf{L}_{32} & \\ & \mathbf{L}_{32} & \mathbf{L}_{33} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{0:k-1} \\ \hat{\mathbf{x}}_k \\ \hat{\mathbf{x}}_{k+1:K} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 - \mathbf{L}_{12}^T \mathbf{L}_{11}^{-1} \mathbf{r}_1 - \mathbf{L}_{32}^T \mathbf{L}_{33}^{-1} \mathbf{r}_3 \\ \mathbf{r}_3 \end{bmatrix} \quad (95)$$

and the solution for $\hat{\mathbf{x}}_k$ is thus

$$\underbrace{\left(\mathbf{L}_{22} - \mathbf{L}_{12}^T \mathbf{L}_{11}^{-1} \mathbf{L}_{12} - \mathbf{L}_{32}^T \mathbf{L}_{33}^{-1} \mathbf{L}_{32}\right)}_{\hat{\mathbf{P}}_k^{-1}} \hat{\mathbf{x}}_k = \underbrace{\left(\mathbf{r}_2 - \mathbf{L}_{12}^T \mathbf{L}_{11}^{-1} \mathbf{r}_1 - \mathbf{L}_{32}^T \mathbf{L}_{33}^{-1} \mathbf{r}_3\right)}_{\mathbf{q}_k} \quad (96)$$

where $\hat{\mathbf{P}}_k, \mathbf{q}_k$ are defined. Essentially, *we have marginalized out $\hat{\mathbf{x}}_{0:k-1}$ and $\hat{\mathbf{x}}_{k+1:K}$.*

Substituting the values of the $\mathbf{L}_{ij}$ blocks back into $\hat{\mathbf{P}}_k^{-1}$ we can see that

$$\hat{\mathbf{P}}_k^{-1} = \mathbf{L}_{22} - \mathbf{L}_{12}^T\mathbf{L}_{11}^{-1}\mathbf{L}_{12} - \mathbf{L}_{32}^T\mathbf{L}_{33}^{-1}\mathbf{L}_{32}$$

$$= \underbrace{\mathbf{H}_{22}^T\left(\mathbf{W}_2^{-1} - \mathbf{W}_2^{-1}\mathbf{H}_{21}(\mathbf{H}_{11}^T\mathbf{W}_1^{-1}\mathbf{H}_{11} + \mathbf{H}_{21}^T\mathbf{W}_2^{-1}\mathbf{H}_{21})^{-1}\mathbf{H}_{21}^T\mathbf{W}_2^{-1}\right)\mathbf{H}_{22}}_{\hat{\mathbf{P}}_{k,f}^{-1}=\mathbf{H}_{22}^T\left(\mathbf{W}_2+\mathbf{H}_{21}(\mathbf{H}_{11}^T\mathbf{W}_1^{-1}\mathbf{H}_{11})^{-1}\mathbf{H}_{21}^T\right)^{-1}\mathbf{H}_{22} \text{ by SMW}} \quad (97)$$

$$+ \underbrace{\mathbf{H}_{32}^T\left(\mathbf{W}_3^{-1} - \mathbf{W}_3^{-1}\mathbf{H}_{33}(\mathbf{H}_{33}^T\mathbf{W}_3^{-1}\mathbf{H}_{33} + \mathbf{H}_{43}^T\mathbf{W}_4^{-1}\mathbf{H}_{43})^{-1}\mathbf{H}_{33}^T\mathbf{W}_3^{-1}\right)\mathbf{H}_{32}}_{\hat{\mathbf{P}}_{k,b}^{-1}=\mathbf{H}_{32}^T\left(\mathbf{W}_3+\mathbf{H}_{33}(\mathbf{H}_{43}^T\mathbf{W}_4^{-1}\mathbf{H}_{43})^{-1}\mathbf{H}_{33}^T\right)^{-1}\mathbf{H}_{32} \text{ by SMW}} \quad (98)$$

$$= \underbrace{\hat{\mathbf{P}}_{k,f}^{-1}}_{\text{forward}} + \underbrace{\hat{\mathbf{P}}_{k,b}^{-1}}_{\text{backward}} \quad (99)$$

where the term labelled **'forward'** depends only on the blocks of $\mathbf{H}$ and $\mathbf{W}$ up to time k and the term **'backward'** depends only on the blocks of $\mathbf{H},\mathbf{W}$ from $k+1$ to K.

Turning into $\mathbf{q}_k$ we substitute the values of $\mathbf{L}_{ij}$ and $\mathbf{r}_i$ blocks

$$\begin{aligned}\mathbf{q}_k &= \mathbf{r}_2 - \mathbf{L}_{12}^T\mathbf{L}_{11}^{-1}\mathbf{r}_1 - \mathbf{L}_{32}^T\mathbf{L}_{33}^{-1}\mathbf{r}_3 \\ &= \underbrace{\mathbf{q}_{k,f}}_{\text{forward}} + \underbrace{\mathbf{q}_{k,b}}_{\text{backward}}\end{aligned} \tag{100}$$

where again 'forward' depends only on quantities up to k and 'backward' depends only on quantities from $k+1$ to K, while

$$\begin{aligned}\mathbf{q}_{k,f} =& -\mathbf{H}_{22}^T\mathbf{W}_2^{-1}\mathbf{H}_{21}(\mathbf{H}_{11}^T\mathbf{W}_1^{-1}\mathbf{H}_{11} + \mathbf{H}_{21}^T\mathbf{W}_2^{-1}\mathbf{H}_{21})^{-1}\mathbf{H}_{11}^T\mathbf{W}_1^{-1}\mathbf{z}_1 \\ &+ \mathbf{H}_{22}^T(\mathbf{W}_2^{-1} - \mathbf{W}_2^{-1}\mathbf{H}_{21}(\mathbf{H}_{11}^T\mathbf{W}_1^{-1}\mathbf{H}_{11} + \mathbf{H}_{21}^T\mathbf{W}_2^{-1}\mathbf{H}_{21})^{-1}\mathbf{H}_{21}^T\mathbf{W}_2^{-1})\mathbf{z}_2\end{aligned} \tag{101}$$

$$\begin{aligned}\mathbf{q}_{k,b} =& \mathbf{H}_{32}^T(\mathbf{W}_3^{-1} - \mathbf{W}_3^{-1}\mathbf{H}_{33}(\mathbf{H}_{33}^T\mathbf{W}_3^{-1}\mathbf{H}_{33} + \mathbf{H}_{43}^T\mathbf{W}_4^{-1}\mathbf{H}_{43})^{-1}\mathbf{H}_{33}^T\mathbf{W}_3^{-1})\mathbf{z}_3 \\ &- \mathbf{H}_{32}^T\mathbf{W}_3^{-1}\mathbf{H}_{33}(\mathbf{H}_{43}^T\mathbf{W}_4^{-1}\mathbf{H}_{43} + \mathbf{H}_{33}^T\mathbf{W}_3^{-1}\mathbf{H}_{33})^{-1}\mathbf{H}_{43}^T\mathbf{W}_4^{-1}\mathbf{z}_4\end{aligned} \tag{102}$$

Recursive Discrete-Time Filtering

Let us define the following two 'forward' and 'backward' estimators $\hat{\mathbf{x}}_{k,f}$ and $\hat{\mathbf{x}}_{k,b}$ respectively

$$\hat{\mathbf{P}}_{k,f}^{-1}\hat{\mathbf{x}}_{k,f} = \mathbf{q}_{k,f} \quad (103)$$

$$\hat{\mathbf{P}}_{k,b}^{-1}\hat{\mathbf{x}}_{k,b} = \mathbf{q}_{k,b} \quad (104)$$

where $\hat{\mathbf{x}}_{k,f}$ depends only on quantities up to time k and $\hat{\mathbf{x}}_{k,b}$ depends only on quantities from time $k+1$ to K. Under these definitions we have that

$$\hat{\mathbf{P}}_{k}^{-1} = \hat{\mathbf{P}}_{k,f}^{-1} + \hat{\mathbf{P}}_{k,b}^{-1} \quad (105)$$

$$\hat{\mathbf{P}}_{k}^{-1}\hat{\mathbf{x}}_{k} = \hat{\mathbf{P}}_{k,f}^{-1}\hat{\mathbf{x}}_{k,f} + \hat{\mathbf{P}}_{k,b}^{-1}\hat{\mathbf{x}}_{k,b} \quad (106)$$

which is the *normalized product of two Gaussian PDFs*.

Referring to Eq. (87) we have that

$$p(\mathbf{x}_k|\mathbf{v},\mathbf{y}) \rightarrow \mathcal{N}(\hat{\mathbf{x}}_k, \hat{\mathbf{P}}_k) \tag{107}$$

$$p(\mathbf{x}_k|\check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k}) \rightarrow \mathcal{N}(\hat{\mathbf{x}}_{k,f}, \hat{\mathbf{P}}_{k,f}) \tag{108}$$

$$p(\mathbf{x}_k|\mathbf{v}_{k+1:K}, \mathbf{y}_{k+1:K}) \rightarrow \mathcal{N}(\hat{\mathbf{x}}_{k,b}, \hat{\mathbf{P}}_{k,b}) \tag{109}$$

where $\hat{\mathbf{P}}_k, \hat{\mathbf{P}}_{k,f}, \hat{\mathbf{P}}_{k,b}$ are the covariance matrices associated with $\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_{k,f}, \hat{\mathbf{x}}_{k,b}$.

$\Rightarrow$ **We have Gaussian estimators with the MAP estimators as the means.**

Next we seek how we can turn the forward Gaussian estimator, $\hat{\mathbf{x}}_{k,f}$, into a recursive filter.

Kalman Filter via MAP

- We shall show how to turn the forward estimator above into a *recursive filter*, the *Kalman Filter* uing the MAP approach.
- Notation is simplified: $\hat{\mathbf{x}}_{k,f} \rightarrow \hat{\mathbf{x}}_k$, $\hat{\mathbf{P}}_{k,f} \rightarrow \hat{\mathbf{P}}_k$ not to be confused with the batch/smoothed estimates presented earlier.

Kalman Filter via MAP

Let us assume that we already have a forwards estimate and its covariance at time $k-1$

$$\{\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}\} \tag{110}$$

These estimates are based on al the data up to and including those at time $k-1$. The goal is to compute

$$\{\hat{\mathbf{x}}_{k}, \hat{\mathbf{P}}_{k}\} \tag{111}$$

using all the data up to and including those at time k.

We do not need to start all over again. We can simply incorporate the new data at time k, $\mathbf{v}_k$ and $\mathbf{y}_k$ into the estimate at time $k-1$

$$\{\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}, \mathbf{v}_k, \mathbf{y}_k\} \rightarrow \{\hat{\mathbf{x}}_{k}, \hat{\mathbf{P}}_{k}\} \tag{112}$$

To see this, let us define

$$\mathbf{z} = \begin{bmatrix} \hat{\mathbf{x}}_{k-1} \\ \mathbf{v}_k \\ \mathbf{y}_k \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{1} & \\ -\mathbf{A}_{k-1} & \mathbf{1} \\ & \mathbf{C}_k \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \hat{\mathbf{P}}_{k-1} & & \\ & \mathbf{Q}_k & \\ & & \mathbf{R}_k \end{bmatrix} \tag{113}$$

where $\{\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}\}$ serves as substitutes for all the data up to time $k-1$.

Kalman Filter via MAP

Our usual MAP solution to the problem is $\hat{\mathbf{x}}$ given by

$$(\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H})\hat{\mathbf{x}} = \mathbf{H}^T\mathbf{W}^{-1}\mathbf{z} \tag{114}$$

We then define

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}'_{k-1} \\ \hat{\mathbf{x}}_k \end{bmatrix} \tag{115}$$

where we distinguish $\hat{\mathbf{x}}'_{k-1}$ from $\hat{\mathbf{x}}_{k-1}$.

- $\hat{\mathbf{x}}'_{k-1}$ is the estimate at time $k-1$ incorporating data up to and including time k
- $\hat{\mathbf{x}}_{k-1}$ is the estimate at time $k-1$ using data up to and including $k-1$.

Kalman Filter via MAP

Substituting in the quantities from Eq. (113) to the least-squares solutiion

$$\begin{bmatrix} \hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} & -\mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \\ -\mathbf{Q}_k^{-1} \mathbf{A}_{k-1} & \mathbf{Q}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}'_{k-1} \\ \hat{\mathbf{x}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k \\ \mathbf{Q}_k^{-1} \mathbf{v}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \end{bmatrix} \quad (116)$$

We can **marginalize** $\hat{\mathbf{x}}'_{k-1}$ as we do not care for it since we aim for a recursive estimator to be used onine and $\hat{\mathbf{x}}'_{k-1}$ incorporates future data. We marginalize by left-multiplying both sides by

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} \right)^{-1} & \mathbf{1} \end{bmatrix} \quad (117)$$

this is an elemantary row operation and thus does not alter the solution to the linear system of equations.

Kalman Filter via MAP

Eq. (116) becomes

$$\begin{bmatrix} \hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} & -\mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \\ \mathbf{0} & \mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} (\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1})^{-1} \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}'_{k-1} \\ \hat{\mathbf{x}}_k \end{bmatrix}$$

$$= \begin{bmatrix} \hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k \\ \mathbf{Q}_k^{-1} \mathbf{A}_{k-1} (\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{A}_{k-1})^{-1} (\hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k) + \mathbf{Q}_k^{-1} \mathbf{v}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \end{bmatrix} \quad (118)$$

The solution for $\hat{\mathbf{x}}_k$ takes the form

$$\left(\underbrace{\mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1}\mathbf{A}_{k-1}(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{A}_{k-1})^{-1}\mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}}_{(\mathbf{Q}_k+\mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T)^{-1}(by\ \ SMW)} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\right)\hat{\mathbf{x}}_k$$

$$= \left(\mathbf{Q}_k^{-1}\mathbf{A}_{k-1}(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{A}_{k-1})^{-1}(\hat{\mathbf{P}}_{k-1}^{-1}\hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{v}_k) + \mathbf{Q}_k^{-1}\mathbf{v}_k + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k\right) \quad (119)$$

Let us define the following quantities

$$\check{\mathbf{P}}_k = \mathbf{Q}_k + \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T \quad (120)$$

$$\hat{\mathbf{P}}_k = (\check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k)^{-1} \quad (121)$$

Kalman Filter via MAP

Eq. (119) becomes

$$\hat{\mathbf{P}}_k^{-1}\mathbf{x}_k = \underbrace{\mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\right)^{-1}\hat{\mathbf{P}}_{k-1}^{-1}}_{\check{\mathbf{P}}_k^{-1}\mathbf{A}_{k-1}}\hat{\mathbf{x}}_{k-1}$$

$$+\underbrace{\left(\mathbf{Q}_k^{-1} - \mathbf{Q}_k^{-1}\mathbf{A}_{k-1}(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{A}_{k-1})^{-1}\mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\right)}_{\check{\mathbf{P}}_k^{-1}}\mathbf{v}_k + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k \tag{122}$$

$$=\check{\mathbf{P}}_k^{-1}\underbrace{\left(\mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k\right)}_{\check{\mathbf{x}}_k} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k \tag{123}$$

where $\check{\mathbf{x}}_k$ is defined as the **'predicted'** value of the state.

Kalman Filter via MAP

Furthermore, in the above we used that

$$
\begin{aligned}
&\mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\underbrace{\left(\hat{\mathbf{P}}_{k-1}^{-1}+\mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\right)^{-1}}_{\text{apply SMW}}\hat{\mathbf{P}}_{k-1}^{-1} \\
&=\mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\left(\hat{\mathbf{P}}_{k-1}-\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T\underbrace{\left(\mathbf{Q}_k+\mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T\right)^{-1}}_{\check{\mathbf{P}}_k^{-1}}\mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\right)\hat{\mathbf{P}}_{k-1}^{-1} \\
&=\left(\mathbf{Q}_k^{-1}-\mathbf{Q}_k^{-1}\underbrace{\mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T}_{\check{\mathbf{P}}_k-\mathbf{Q}_k}\check{\mathbf{P}}_k^{-1}\right)\mathbf{A}_{k-1} \\
&=\left(\mathbf{Q}_k^{-1}-\mathbf{Q}_k^{-1}+\check{\mathbf{P}}_k^{-1}\right)\mathbf{A}_{k-1} \\
&=\check{\mathbf{P}}_k^{-1}\mathbf{A}_{k-1}
\end{aligned}
\tag{124}
$$

Kalman Filter via MAP

Bringing the above together, leads to a recursive filter update formulation as follows

$$\begin{aligned}
\text{predictor:} \quad & \check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k && (125)\\
& \check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k && (126)\\
\text{corrector:} \quad & \hat{\mathbf{P}}_k^{-1} = \check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k && (127)\\
& \hat{\mathbf{P}}_k^{-1}\hat{\mathbf{x}}_k = \check{\mathbf{P}}_k^{-1}\check{\mathbf{x}}_k + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k && (128)
\end{aligned}$$

which represents the *inverse covariance* or *information form* of the *Kalman Filter*.

Kalman Filter via MAP

To get to the *canonical form* of the *Kalman Filter* some algebraic manipulation is required. Let us define the **Kalman gain $\mathbf{K}_k$** as

$$\mathbf{K}_k = \hat{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} \tag{129}$$

Then we perform the following operations

$$\begin{aligned} \mathbf{1} &= \hat{\mathbf{P}}_k (\check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k) \\ &= \hat{\mathbf{P}}_k \check{\mathbf{P}}_k^{-1} + \mathbf{K}_k \mathbf{C}_k && (130) \\ \hat{\mathbf{P}}_k &= (\mathbf{1} - \mathbf{K}_k \mathbf{C}_k) \check{\mathbf{P}}_k && (131) \\ \underbrace{\hat{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1}}_{\mathbf{K}_k} &= (\mathbf{1} - \mathbf{K}_k \mathbf{C}_k) \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} && (132) \\ \mathbf{K}_k (\mathbf{1} + \mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1}) &= \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} && (133) \end{aligned}$$

Kalman Filter via MAP

Solving for $\mathbf{K}_k$ in the last equation, we write the recursive filter equations

$$\begin{aligned}
\text{predictor:} \quad & \hat{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k && (134)\\
& \check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k && (135)\\
\text{Kalman gain:} \quad & \mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{C}_k^T(\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{R}_k)^{-1} && (136)\\
\text{corrector:} \quad & \hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\check{\mathbf{P}}_k && (137)\\
& \hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k \underbrace{(\mathbf{y}_k - \mathbf{C}_k\check{\mathbf{x}}_k)}_{\text{innovation}} && (138)
\end{aligned}$$

where the innovation term represents the difference between the actual and the expected measurements and the role of the Kalman gain is to properly weight the innovation's contribution to the estimate as compared to the prediction step.

In the above form, the Kalman Filter has been the workhorse of state estimation and are identical to the forward pass Rauch-Tung-Striebel smoother.

Kalman Filter via Bayesian Inference

A simpler derivation of the Kalman Filter can be derived using the Bayesian inference approach. The Gaussian prior estimate at $k-1$ is

$$p(\mathbf{x}_{k-1}|\check{\mathbf{x}}_0, \mathbf{v}_{1:k-1}, \mathbf{y}_{0:k-1}) = \mathcal{N}(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}) \tag{139}$$

First, for the **prediction step**, we incorporate the latest input, $\mathbf{v}_k$, to write a 'priori' at time k

$$p(\mathbf{x}_k|\check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k-1}) = \mathcal{N}(\check{\mathbf{x}}_k, \check{\mathbf{P}}_k) \tag{140}$$

where

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k \tag{141}$$

$$\check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k \tag{142}$$

which are identical to the prediction equations from the previous derivation. The last two equations can in fact be found by exactly passing the prior at $k-1$ through the *linear motion model*.

Kalman Filter via Bayesian Inference

For the mean we have

$$\check{\mathbf{x}}_k = E[\mathbf{x}_k] = E[\mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k]$$
$$= \mathbf{A}_{k-1} \underbrace{E[\mathbf{x}_{k-1}]}_{\hat{\mathbf{x}}_{k-1}} + \mathbf{v}_k + \underbrace{E[\mathbf{w}_k]}_{\mathbf{0}} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k \quad (143)$$

And for the covariance

$$\check{\mathbf{P}}_k = E[(\mathbf{x}_k - E[\mathbf{x}_k])(\mathbf{x}_k - E[\mathbf{x}_k])^T]$$
$$= E[(\mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k - \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} - \mathbf{v}_k)(\mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k - \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} - \mathbf{v}_k)^T]$$
$$= \mathbf{A}_{k-1} \underbrace{E[(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})^T]}_{\hat{\mathbf{P}}_{k-1}} \mathbf{A}_{k-1}^T + \underbrace{E[\mathbf{w}_k\mathbf{w}_k^T]}_{\mathbf{Q}_k}$$
$$= \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k \quad (144)$$

and finally for the latest measurement, at time k, we write

$$p(\mathbf{x}_k, \mathbf{y}_k | \check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k-1}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \check{\mathbf{P}}_k & \check{\mathbf{P}}_k \mathbf{C}_k^T \\ \mathbf{C}_k \check{\mathbf{P}}_k & \mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T + \mathbf{R}_k \end{bmatrix} \right) \tag{145}$$

Based on Bayesian inference theory, we can directly write the conditional density for $\mathbf{x}_k$ - or in other terms the **posterior** - as

$$p(\mathbf{x}_k | \check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k}) = \mathcal{N}(\underbrace{(\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} (\mathbf{y}_k - \boldsymbol{\mu}_y)}_{\hat{\mathbf{x}}_k}, \underbrace{\boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{yx}}_{\hat{\mathbf{P}}_k}) \tag{146}$$

where $\hat{\mathbf{x}}_k$ is the mean and $\hat{\mathbf{P}}_k$ is the covariance

Substituting in the moments derived above we derive the **correction equations** identical to those from the MAP-based derivation.

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{C}_k^T (\mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T + \mathbf{R}_k)^{-1} \tag{147}$$

$$\hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k \mathbf{C}_k) \check{\mathbf{P}}_k \tag{148}$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \check{\mathbf{x}}_k) \tag{149}$$

The two results are identical as both the *motion* and *measurement* models are *linear* and the *noises* and *prior* are *Gaussian*.

- Under these conditions, the posterior density is exactly Gaussian. Thus the *mean* and *mode* of the posterior are the same.

Error Dynamics

To understand filter performance and stability properties, it is useful to look at the difference between the estimated and the actual state. Let us define the following errors

$$\check{\mathbf{e}}_k = \check{\mathbf{x}}_k - \mathbf{x}_k \tag{150}$$

$$\hat{\mathbf{e}}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k \tag{151}$$

using the system dynamics and the equations of the Kalman filter we can rewrite

$$\check{\mathbf{e}}_k = \mathbf{A}_{k-1}\hat{\mathbf{e}}_{k-1} - \mathbf{w}_k \tag{152}$$

$$\hat{\mathbf{e}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\check{\mathbf{e}}_k + \mathbf{K}_k\mathbf{n}_k \tag{153}$$

where $\hat{\mathbf{e}}_0 = \hat{\mathbf{x}}_0 - \mathbf{x}_0$.

Error Dynamics

As can be shown by the above equations, if $E[\hat{\mathbf{e}}_0] = \mathbf{0}$ then $E[\hat{\mathbf{e}}_k] = \mathbf{0}$.
Proof by induction: It is true for $k = 0$ - assume it is also true for $k - 1$, then

$$E[\check{\mathbf{e}}_k] = \mathbf{A}_{k-1} \underbrace{E[\hat{\mathbf{e}}_{k-1}]}_{\mathbf{0}} - \underbrace{E[\mathbf{w}_k]}_{\mathbf{0}} = \mathbf{0} \tag{154}$$

$$E[\hat{\mathbf{e}}_k] = (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k) \underbrace{E[\check{\mathbf{e}}_k]}_{\mathbf{0}} + \mathbf{K}_k \underbrace{E[\mathbf{n}_k]}_{\mathbf{0}} = \mathbf{0} \tag{155}$$

Thus it is true for all k. *This means that our estimator is* **unbiased**.

It is also true that as long as $E[\hat{\mathbf{e}}_0 \hat{\mathbf{e}}_0^T] = \hat{\mathbf{P}}_0$,

$$E[\check{\mathbf{e}}_k \check{\mathbf{e}}_k^T] = \check{\mathbf{P}}_k \qquad (156)$$

$$E[\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T] = \hat{\mathbf{P}}_k \qquad (157)$$

for $k > 0$. *This means that our estimator is* **consistent**.

Proof by induction: It is true for $k = 0$ by assertion - assume $E[\hat{\mathbf{e}}_{k-1} \hat{\mathbf{e}}_{k-1}^T] = \hat{\mathbf{P}}_{k-1}$

$$\begin{aligned} E[\check{\mathbf{e}}_k \check{\mathbf{e}}_k^T] &= E\left[(\mathbf{A}_{k-1}\hat{\mathbf{e}}_{k-1} - \mathbf{w}_k)(\mathbf{A}_{k-1}\hat{\mathbf{e}}_{k-1} - \mathbf{w}_k)^T\right] \\ &= \mathbf{A}_{k-1} \underbrace{E[\hat{\mathbf{e}}_{k-1}\hat{\mathbf{e}}_{k-1}^T]}_{\hat{\mathbf{P}}_{k-1}} \mathbf{A}_{k-1}^T - \mathbf{A}_{k-1} \underbrace{E[\hat{\mathbf{e}}_{k-1}\mathbf{w}_k^T]}_{\mathbf{0} \text{ by independence}} - \underbrace{E[\mathbf{w}_k \hat{\mathbf{e}}_{k-1}^T]}_{\mathbf{0} \text{ by independence}} \mathbf{A}_{k-1}^T \underbrace{E[\mathbf{w}_k \mathbf{w}_k^T]}_{\mathbf{Q}_k} \\ &= \check{\mathbf{P}}_k \end{aligned} \qquad (158)$$

and

$$
\begin{aligned}
E[\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T] &= E\left[((\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\check{\mathbf{e}}_k + \mathbf{K}_k\mathbf{n}_k)((\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\check{\mathbf{e}}_k + \mathbf{K}_k\mathbf{n}_k)^T\right] \\
&= (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\underbrace{E[\check{\mathbf{e}}_k\check{\mathbf{e}}_k^T]}_{\check{\mathbf{P}}_k}(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)^T + (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\underbrace{E[\check{\mathbf{e}}_k\mathbf{n}_k^T]}_{\mathbf{0}\text{ by independence}}\mathbf{K}_k^T \\
&\quad + \mathbf{K}_k\underbrace{E[\mathbf{n}_k\check{\mathbf{e}}_k^T]}_{\mathbf{0}\text{ by independence}}(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)^T + \mathbf{K}_k\underbrace{E[\mathbf{n}_k\mathbf{n}_k^T]}_{\mathbf{R}_k}\mathbf{K}_k^T \\
&= (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\check{\mathbf{P}}_k \underbrace{-\hat{\mathbf{P}}_k\mathbf{C}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T}_{\mathbf{0}\text{ because }\mathbf{K}_k=\hat{\mathbf{P}}_k\mathbf{C}_k^T\mathbf{R}_k^{-1}} \\
&= \hat{\mathbf{P}}_k
\end{aligned}
\tag{159}
$$

It is therefore ture for all k.

Error Dynamics

- The true uncertainty in the system (i.e., the covariance of the error, $E[\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T]$) is perfectly modelled by our estimate of the covariance, $\hat{\mathbf{P}}_k$.
- The Kalman filter is an optimal filter.
- The Kalman filter is called the *Best Linear Unbiased Estimate (BLUE)*.
- The covariance of the Kalman filter is exactly at the Cramér-Rao Lower Bound (CRLB). In other words we cannot be any more certain in our estimate given the uncertainty in the measurements we have used in that estimate.
- The expectations employed here are overall *all possible outcmes of the random variables*! They are not time averages. The average performance over an infinite number of trials will be that of zero error **but** this does not imply that within a single trial the error will be zero or decay to zero over time.

Existence, Uniqueness, and Observability

- Below we present the key steps to be taken towards a proof of the stability of the Kalman Filter.
- We consider the LTI case.

To avoid confusion with the lifted form of equations we use italics in the following notation simplification $\mathbf{A}_k = A, \mathbf{C}_k = C, \mathbf{Q}_k = Q, \mathbf{R}_k = R$

`Step 1:` The covariance equation of the KF can be iterated to convergence prior. A question is whether the covariance will converge to its steady-state value and, if so, whether it will be unique. Writing P for the steady-state value for $\check{\mathbf{P}}_k$ we have that the following holds at steady-state:

$$P = A(1 - KC)P(1 - KC)^T A^T + AKRK^T A^T + Q \tag{160}$$

which is one form of the Discrete Algebraic Riccati Equation (DARE).

Existence, Uniqueness, and Observability

The DARE has a unique positive-semidefinite solution, P, if and only if the following conditions hold

- $R \succ 0$ as already assumed in the batch LG case
- $Q \succeq 0$ while in the batch LG we have assumed $Q \succ 0$
- (A, V) is stabilizable with $V^T V = Q$ - this condition is redundant when $Q \succ 0$
- (A, C) is detectable which is the same as observable except any ubobservable eigenvalues are stable.

Existence, Uniqueness, and Observability

`Step 2:` Once the covariance evolves to its steady-state value, P, so does the Kalman gain. Let K be the steady-state value of K_k. We have

$$K = PC^T(CPC^T + R)^{-1} \tag{161}$$

for the steady-state Kalman gain.

Existence, Uniqueness, and Observability

`Step 3:` The errpr dynamics of the filter are then stable

$$E[\check{\mathbf{e}}_k] = \underbrace{A(1 - KC)}_{\text{eigs.}<1 \text{ in mag.}} E[\check{\mathbf{e}}_{k-1}] \tag{162}$$

we can see this by noting that for any eigenvector, $\mathbf{v}$, corresponding to an eigenvalue, λ, of $(1 - KC)^T A^T$, we have

$$\mathbf{v}^T P \mathbf{v} = \underbrace{\mathbf{v}^T A(1 - KC)}_{\lambda \mathbf{v}^T} P \underbrace{(1 - KC)^T A^T \mathbf{v}}_{\lambda \mathbf{v}} + \mathbf{v}^T (AKRK^T A^T + Q)\mathbf{v} \tag{163}$$

$$(1 - \lambda^2) \underbrace{\mathbf{v}^T P \mathbf{v}}_{>0} = \underbrace{\mathbf{v}^T (AKRK^T A^T + Q)\mathbf{v}}_{>0} \tag{164}$$

which means that it must $|\lambda| < 1$ and thus the steady-state error dynamics are stable.

Thank you

Q&A

Assignments

Other matters