Lecture 4: Recursive Linear-Gaussian Estimation

AER1513: State Estimation

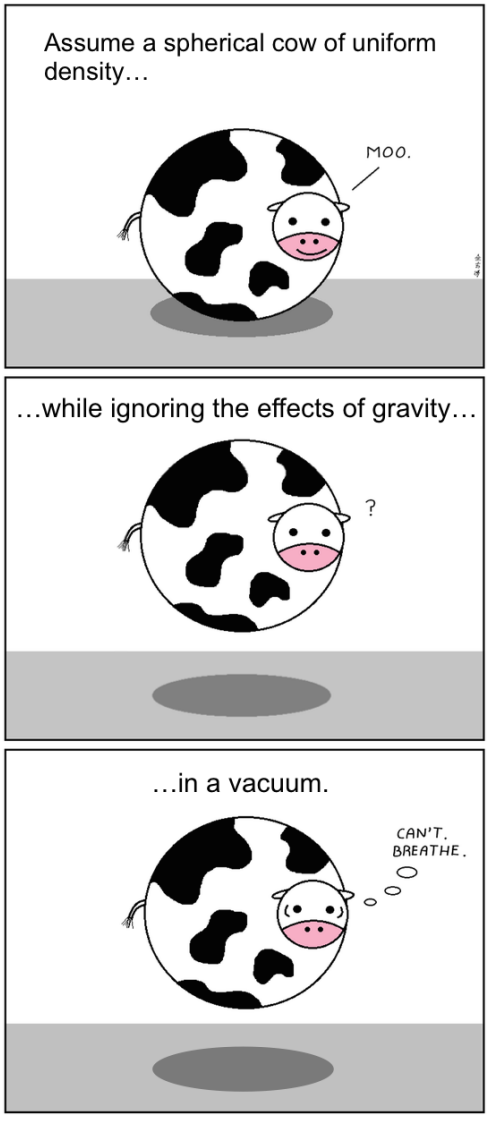
Timothy D. Barfoot

University of Toronto

UNIVERSITY OF TORONTO

Outline

Why linear-Gaussian?



- most of the classical results are for linear-Gaussian systems
- we can often find results without approximation
- we can get some intuition for real systems by studying them

"Equation (1.2-9) is a second order, nonlinear, vector, differential equation which has defied solution in its present form. It is here therefore we depart from the realities of nature to make some simplifying assumptions..."

– Bate et al., Fundamentals of Astrodynamics (1971)

"Linear systems are very important because we can solve them exactly!"

– Sami Mikhail (heard third hand via Raja Mukherji)

System

– we define our system using the following linear, time-varying models:

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

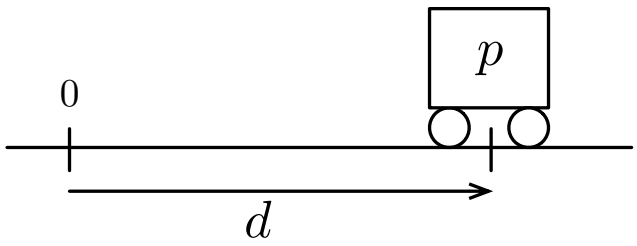
$$\text{observation model:} \quad \mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k, \quad k = 0 \dots K \tag{1b}$$

where k is the discrete-time index and K its maximum

– the variables have the following meanings:

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

System example



$$\ddot{p}(t) = a(t) + w(t)$$

$$\dot{p}(t) = v(t)$$
$$\dot{v}(t) = a(t) + w(t)$$

$$p_k \approx p_{k-1} + Tv_{k-1} + \frac{1}{2}T^2(a_k + w_k)$$
$$v_k \approx v_{k-1} + T\,(a_k + w_k)$$

$$d_k = p_k + n_k$$

$$\underbrace{d_k}_{\mathbf{y}_k} = \underbrace{\begin{bmatrix}1 & 0\end{bmatrix}}_{\mathbf{C}_k} \underbrace{\begin{bmatrix}p_k \\ v_k\end{bmatrix}}_{\mathbf{x}_k} + \underbrace{n_k}_{\mathbf{n}_k}$$

observation model

$$\underbrace{\begin{bmatrix}p_k \\ v_k\end{bmatrix}}_{\mathbf{x}_k} = \underbrace{\begin{bmatrix}1 & T \\ 0 & 1\end{bmatrix}}_{\mathbf{A}_{k-1}} \underbrace{\begin{bmatrix}p_{k-1} \\ v_{k-1}\end{bmatrix}}_{\mathbf{x}_{k-1}} + \underbrace{\begin{bmatrix}\frac{1}{2}T^2 \\ T\end{bmatrix} a_k}_{\mathbf{v}_k} + \underbrace{\begin{bmatrix}\frac{1}{2}T^2 \\ T\end{bmatrix} w_k}_{\mathbf{w}_k}$$

motion model

UNIVERSITY OF TORONTO

What do we know?

- although we want to know the state of the system (at all times), we only have access to the following quantities, and must base our estimate, $\hat{\mathbf{x}}_k$, on just this information:
  - (i) the initial state knowledge, $\check{\mathbf{x}}_0$, and the associated covariance matrix, $\check{\mathbf{P}}_0$; sometimes we do not have this piece of information and must do without
  - (ii) the inputs, $\mathbf{v}_k$, which typically come from the output of our controller and so are known; we also have the associated process noise covariance, $\mathbf{Q}_k$
  - (iii) the measurements, $\mathbf{y}_{k,\text{meas}}$, which are realizations of the associated random variables, $\mathbf{y}_k$, and the associated covariance matrix, $\mathbf{R}_k$
- we will use $\hat{(\cdot)}$ to indicate posterior estimates (incorporating measurements) and $\check{(\cdot)}$ to indicate prior estimates (not incorporating measurements)

Problem statement

– we define the state estimation problem as follows:

Definition

The problem of state estimation is to come up with 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, $\mathbf{y}_{0:K,\text{meas}}$, a sequence of inputs, $\mathbf{v}_{1:K}$, as well as knowledge of the system's motion and observation models.

– the rest of the linear-Gaussian section will investigate a suite of techniques for addressing this state estimation problem

– our approach will always be to attempt to come up with not only a state estimate, but also to quantify the uncertainty in that estimate

Batch is best, but recursive works online

- in the last lecture, we investigated batch linear-Gaussian techniques (sometimes called smoothers)
- the batch solution is very useful for computing state estimates after the fact because it uses all the measurements in the estimation of all the states at once (hence the usage of 'batch')
- a batch method cannot be used online since we cannot employ future measurements to estimate past states
- in this lecture, we'll look at recursive state estimators (called filters), which can be used online

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

What density do we want to know?

- the batch solution estimated the whole trajectory at once, but now we only care about the last timestep so we can marginalize out the other states:

$$p(\mathbf{x}_k|\mathbf{v},\mathbf{y}) = \int_{\mathbf{x}_{i,\forall i\neq k}} \underbrace{p(\mathbf{x}_0,\ldots,\mathbf{x}_K|\mathbf{v},\mathbf{y})}_{\text{batch finds this}}\, d\mathbf{x}_{i,\forall i\neq k} \tag{2}$$

- we can factor this probability density into two parts:

$$p(\mathbf{x}_k|\mathbf{v},\mathbf{y}) = \underbrace{p(\mathbf{x}_k|\check{\mathbf{x}}_0,\mathbf{v}_{1:k},\mathbf{y}_{0:k})}_{\text{this is what we want}} p(\mathbf{x}_k|\mathbf{v}_{k+1:K},\mathbf{y}_{k+1:K}) \tag{3}$$

- we can't rely on future measurements, that would be acausal, so we just want the first factor

All roads lead to Rome

- our goal is to compute the density,

$$p(\mathbf{x}_k|\check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k}) = \mathcal{N}(\hat{\mathbf{x}}_{k,f}, \hat{\mathbf{P}}_{k,f}) \tag{4}$$

- the solution should be recursive, allowing us to incorporate the latest input/measurement incrementally:

$$\{\hat{\mathbf{x}}_{k-1,f}, \mathbf{v}_k, \mathbf{y}_k\} \mapsto \hat{\mathbf{x}}_{k,f} \tag{5}$$

- we already know one way to do this – the forwards pass of the RTS smoother does this and is called the Kalman filter (KF)
- we can also derive the KF equations in several other ways, and we'll look at three:
  - (i) maximum a posteriori
  - (ii) Bayesian inference
  - (iii) gain optimization

Approach (i): maximum a posteriori

- we want to find a recursive approach that allows the following update:

$$\{\hat{\mathbf{x}}_{k-1}, \mathbf{v}_k, \mathbf{y}_k\} \mapsto \hat{\mathbf{x}}_k \tag{6}$$

  where we have dropped the subscript f from the last slide to keep the notation simpler

- this is possible because the old estimate, $\hat{\mathbf{x}}_{k-1}$, summarizes the initial state knowledge, inputs, and measurements up to $k-1$:

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

- we can do this exactly in the linear-Gaussian case because of the Markov property – more on this later

MAP

- we can think of $\hat{\mathbf{x}}_{k-1}$ as a pseudomeasurement, taking the place of the initial state knowledge, inputs, and measurements up to $k-1$
- taking the MAP approach, we can set up a small batch estimation problem as follows:

$$\left(\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}\right)\hat{\mathbf{x}} = \mathbf{H}^T\mathbf{W}^{-1}\mathbf{z} \tag{7}$$

where

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}'_{k-1} \\ \hat{\mathbf{x}}_k \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} \hat{\mathbf{x}}_{k-1} \\ \mathbf{v}_k \\ \mathbf{y}_k \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{1} & \\ -\mathbf{A}_{k-1} & \mathbf{1} \\ & \mathbf{C}_k \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \hat{\mathbf{P}}_{k-1} & & \\ & \mathbf{Q}_k & \\ & & \mathbf{R}_k \end{bmatrix} \tag{8}$$

- need to be careful to understand what $\hat{\mathbf{x}}'_{k-1}$ represents:

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

$\hat{\mathbf{x}}_k$

MAP

– multiplying out, the system of equations becomes

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

$$= \underbrace{\begin{bmatrix} \hat{\mathbf{P}}_{k-1}^{-1} \hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T \mathbf{Q}_k^{-1} \mathbf{v}_k \\ \mathbf{Q}_k^{-1} \mathbf{v}_k + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \end{bmatrix}}_{\mathbf{H}^T \mathbf{W}^{-1} \mathbf{z}} \tag{9}$$

– it was convenient to define $\hat{\mathbf{x}}$ as we did in order to set up the mini batch problem, but we only really care about $\hat{\mathbf{x}}_k$

– unfortunately, the two equations are coupled

– luckily, we can marginalize out $\hat{\mathbf{x}}'_{k-1}$ using the Schur complement trick

MAP: Schur complement

– we can marginalize out $\hat{\mathbf{x}}'_{k-1}$ by left-multiplying both sides by

$$\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\right)^{-1} & \mathbf{1} \end{bmatrix} \tag{10}$$

– the system of equations becomes

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

$$= \begin{bmatrix} \hat{\mathbf{P}}_{k-1}^{-1}\hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{v}_k \\ \mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\left(\hat{\mathbf{P}}_{k-1}^{-1} + \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{A}_{k-1}\right)^{-1}\left(\hat{\mathbf{P}}_{k-1}^{-1}\hat{\mathbf{x}}_{k-1} - \mathbf{A}_{k-1}^T\mathbf{Q}_k^{-1}\mathbf{v}_k\right) \\ +\mathbf{Q}_k^{-1}\mathbf{v}_k + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k \end{bmatrix} \tag{11}$$

– the second equation now does not depend on $\hat{\mathbf{x}}'_{k-1}$

MAP

– pulling out only the second equation we have

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

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

$$\left. + \quad \mathbf{Q}_k^{-1}\mathbf{v}_k + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k\right) \quad (12)$$

– we see that the SMW identity can be used to make the LHS look much more compact

MAP

- next, we define the following helpful quantities,

$$\check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k \tag{13a}$$
$$\check{\mathbf{P}}_k = \mathbf{Q}_k + \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T \tag{13b}$$
$$\hat{\mathbf{P}}_k = \left(\check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k\right)^{-1} \tag{13c}$$

- plugging these in to our system of equations from the last slide, after a bit of manipulation it becomes

$$\hat{\mathbf{P}}_k^{-1}\hat{\mathbf{x}}_k = \check{\mathbf{P}}_k^{-1}\check{\mathbf{x}}_k + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k \tag{14}$$

- together, these four equations are the inverse covariance or information form of the KF

MAP

– to get to the canonical form of the KF we define the Kalman gain

$$\mathbf{K}_k = \hat{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} \tag{15}$$

and then manipulate into a different form:

$$\begin{aligned}
\mathbf{1} &= \hat{\mathbf{P}}_k \left(\check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \right) \\
&= \hat{\mathbf{P}}_k \check{\mathbf{P}}_k^{-1} + \mathbf{K}_k \mathbf{C}_k && \text{(16a)} \\
\hat{\mathbf{P}}_k &= \left(\mathbf{1} - \mathbf{K}_k \mathbf{C}_k \right) \check{\mathbf{P}}_k && \text{(16b)} \\
\underbrace{\hat{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1}}_{\mathbf{K}_k} &= \left(\mathbf{1} - \mathbf{K}_k \mathbf{C}_k \right) \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} && \text{(16c)} \\
\mathbf{K}_k \left(\mathbf{1} + \mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} \right) &= \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} && \text{(16d)} \\
\mathbf{K}_k &= \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} \left(\mathbf{1} + \mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T \mathbf{R}_k^{-1} \right)^{-1} && \text{(16e)}
\end{aligned}$$

– we could have equivalently started with this last version as our definition of $\mathbf{K}_k$

MAP

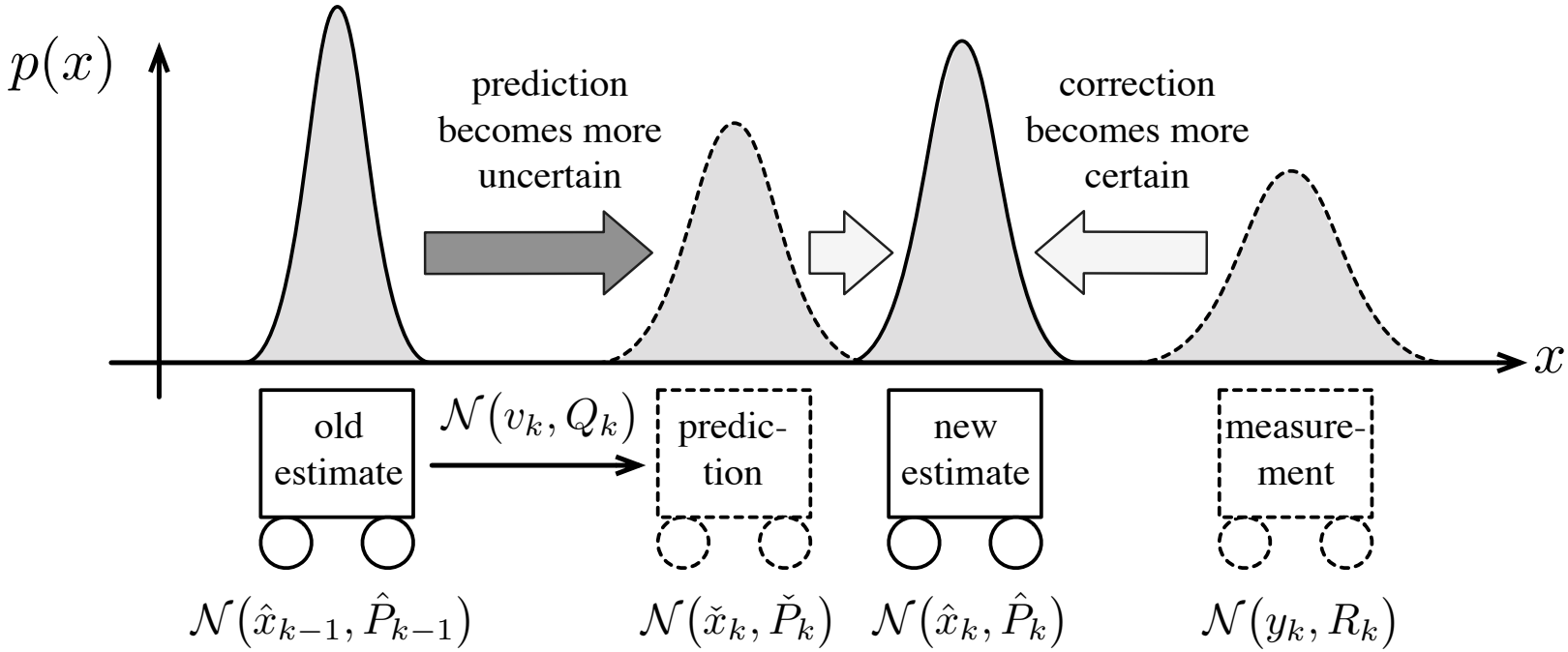
– we can now restate the filter equations as

$$\begin{aligned}
\text{predictor:} \quad \check{\mathbf{P}}_k &= \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k && \text{(17a)}\\
\check{\mathbf{x}}_k &= \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k && \text{(17b)}\\
\text{Kalman gain:} \quad \mathbf{K}_k &= \check{\mathbf{P}}_k\mathbf{C}_k^T\left(\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{R}_k\right)^{-1} && \text{(17c)}\\
\text{corrector:} \quad \hat{\mathbf{P}}_k &= \left(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{P}}_k && \text{(17d)}\\
\hat{\mathbf{x}}_k &= \check{\mathbf{x}}_k + \mathbf{K}_k\underbrace{\left(\mathbf{y}_k - \mathbf{C}_k\check{\mathbf{x}}_k\right)}_{\text{innovation}} && \text{(17e)}
\end{aligned}$$

– these five equations represent the canonical Kalman filter and have been the workhorse of estimation since 1960

– they are identical to the forwards pass of the RTS smoother that we saw earlier (with the subscript f dropped)

Predictor Corrector



Approach (ii): Bayesian inference

- we can get to the KF equations yet another way that is perhaps the simplest to understand
- our Gaussian prior estimate at $k-1$ is

$$p(\mathbf{x}_{k-1}|\check{\mathbf{x}}_0, \mathbf{v}_{1:k-1}, \mathbf{y}_{0:k-1}) = \mathcal{N}\left(\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{P}}_{k-1}\right) \tag{18}$$

- we carry out the prediction step directly, by incorporating the latest input, $\mathbf{v}_k$, to write a prior at time k:

$$p(\mathbf{x}_k|\check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k-1}) = \mathcal{N}\left(\check{\mathbf{x}}_k, \check{\mathbf{P}}_k\right) \tag{19}$$

where

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k \tag{20a}$$

$$\check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k \tag{20b}$$

- these are just the prediction equations from the MAP method

Bayesian inference

- the prediction expressions can be found by exactly passing the prior at $k-1$ through the linear motion model
- for the mean we have

$$\begin{aligned}\check{\mathbf{x}}_k = E[\mathbf{x}_k] &= E[\mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k] \\ &= \mathbf{A}_{k-1} \underbrace{E[\mathbf{x}_{k-1}]}_{\hat{\mathbf{x}}_{k-1}} + \mathbf{v}_k + \underbrace{E[\mathbf{w}_k]}_{\mathbf{0}} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k \end{aligned} \tag{21}$$

- for the covariance we have

$$\begin{aligned}\check{\mathbf{P}}_k &= E\left[(\mathbf{x}_k - E[\mathbf{x}_k])(\mathbf{x}_k - E[\mathbf{x}_k])^T\right] \\ &= E[(\mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k - \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} - \mathbf{v}_k) \\ &\qquad \times (\mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k - \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} - \mathbf{v}_k)^T\Big] \\ &= \mathbf{A}_{k-1} \underbrace{E\left[(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1})^T\right]}_{\hat{\mathbf{P}}_{k-1}} \mathbf{A}_{k-1}^T + \underbrace{E\left[\mathbf{w}_k\mathbf{w}_k^T\right]}_{\mathbf{Q}_k} \\ &= \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k \end{aligned} \tag{22}$$

Bayesian inference

- for the correction step, we express the joint density of our state and latest measurement, at time k, as a Gaussian:

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{y}_k | \check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k-1}) &= \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix} \right) \\ &= \mathcal{N}\left(\begin{bmatrix} \check{\mathbf{x}}_k \\ \mathbf{C}_k \check{\mathbf{x}}_k \end{bmatrix}, \begin{bmatrix} \check{\mathbf{P}}_k & \check{\mathbf{P}}_k \mathbf{C}_k^T \\ \mathbf{C}_k \check{\mathbf{P}}_k & \mathbf{C}_k \check{\mathbf{P}}_k \mathbf{C}_k^T + \mathbf{R}_k \end{bmatrix} \right) \end{aligned} \tag{23}$$

- we then recall that we can factor this as

$$p(\mathbf{x}_k, \mathbf{y}_k | \check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k-1}) = \underbrace{p(\mathbf{x}_k | \check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k})}_{\text{posterior}} p(\mathbf{y}_k | \check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k-1}) \tag{24}$$

- we only care about the first factor since is the final density that we want at time k

Bayesian inference

– looking back to the lecture on probability where we introduced Bayesian inference, we can directly write the conditional density for $\mathbf{x}_k$ (i.e., the posterior) as

$$p(\mathbf{x}_k|\check{\mathbf{x}}_0, \mathbf{v}_{1:k}, \mathbf{y}_{0:k}) = \mathcal{N}\Big(\underbrace{\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_y)}_{\hat{\mathbf{x}}_k}, \underbrace{\boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}}_{\hat{\mathbf{P}}_k}\Big) \tag{25}$$

where we have defined $\hat{\mathbf{x}}_k$ as the mean and $\hat{\mathbf{P}}_k$ as the covariance

– substituting in $\boldsymbol{\Sigma}_{xx}$, $\boldsymbol{\Sigma}_{xy}$, $\boldsymbol{\Sigma}_{yy}$, we have

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{C}_k^T\left(\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{R}_k\right)^{-1} \tag{26a}$$

$$\hat{\mathbf{P}}_k = \left(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{P}}_k \tag{26b}$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{C}_k\check{\mathbf{x}}_k\right) \tag{26c}$$

– this is the same as the correction step from the MAP approach

Approach (iii): gain optimization

- the Kalman filter is often referred to as being optimal
- we did perform an optimization to come up with the recursive relations above in the MAP derivation but, there are other ways to look at the optimality of the KF
- one way is to assume we have an estimator with the correction step taking the form

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{C}_k \check{\mathbf{x}}_k \right) \tag{27}$$

- we then try to find the optimal gain matrix, $\mathbf{K}_k$, to blend the corrective measurements with the prediction

Gain optimization

- if we define the error in the state estimate to be

$$\hat{\mathbf{e}}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k \tag{28}$$

  then we have

$$E[\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T] = (\mathbf{1} - \mathbf{K}_k \mathbf{C}_k)\, \check{\mathbf{P}}_k\, (\mathbf{1} - \mathbf{K}_k \mathbf{C}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \tag{29}$$

- we then define a cost function of the form

$$J(\mathbf{K}_k) = \frac{1}{2} \text{tr}\, E[\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T] = E\left[\frac{1}{2} \hat{\mathbf{e}}_k^T \hat{\mathbf{e}}_k\right] \tag{30}$$

  which quantifies the magnitude of the covariance of $\hat{\mathbf{e}}_k$
- we can minimize this cost directly with respect to $\mathbf{K}_k$, to generate the minimum variance estimate

Gain optimization

– we will make use of the identities

$$\frac{\partial \text{tr}\,\mathbf{XY}}{\partial \mathbf{X}} \equiv \mathbf{Y}^T, \qquad \frac{\partial \text{tr}\,\mathbf{XZX}^T}{\partial \mathbf{X}} \equiv 2\mathbf{XZ} \tag{31}$$

where $\mathbf{Z}$ is symmetric

– then we have

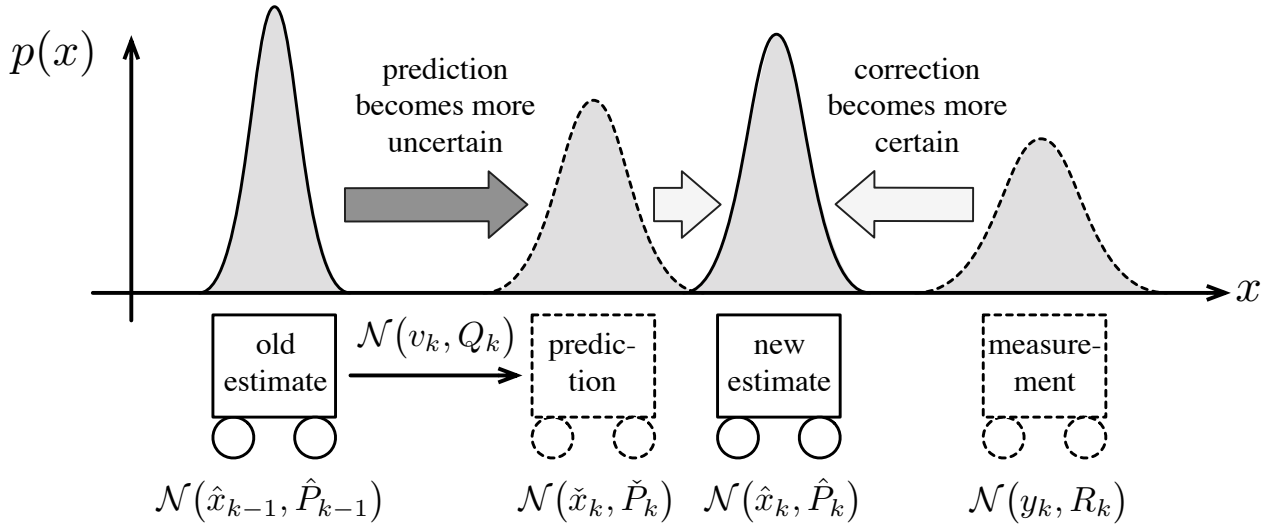
$$\frac{\partial J(\mathbf{K}_k)}{\partial \mathbf{K}_k} = -\left(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{K}_k\mathbf{R}_k \tag{32a}$$

– setting this to zero and solving for $\mathbf{K}_k$ we have

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{C}_k^T\left(\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{R}_k\right)^{-1} \tag{33}$$

which is our usual expression for the Kalman gain

Kalman filter summary



predictor:

$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k \quad (34a)$$

$$\check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k \quad (34b)$$

Kalman gain:

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{C}_k^T\left(\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{R}_k\right)^{-1} \quad (34c)$$

corrector:

$$\hat{\mathbf{P}}_k = \left(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{P}}_k \quad (34d)$$

$$\hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k\underbrace{\left(\mathbf{y}_k - \mathbf{C}_k\check{\mathbf{x}}_k\right)}_{\text{innovation}} \quad (34e)$$

Why are MAP and full Bayesian approaches the same?

- this is because 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 (i.e., maximum) of the posterior are one and the same
- this property does not hold if we switch to a nonlinear measurement model, which we will discuss later

KF discussion

- for a linear system with Gaussian noise, the Kalman filter equations are the best linear unbiased estimate (BLUE); this means they are performing right at the Cramér-Rao lower bound
- initial conditions must be provided:

$$\left\{\check{\mathbf{x}}_0, \check{\mathbf{P}}_0\right\} \tag{35}$$

- the covariance equations can be propagated independently of the mean equations – sometimes a steady-state value of $\mathbf{K}_k$ is computed and used for all time-steps to propagate the mean; this is known as the steady-state Kalman filter
- at implementation, we must use $\mathbf{y}_{k,\text{meas}}$, the actual readings we receive from our sensors, in the filter
- a similar set of equations can be developed for the backwards estimator that runs backwards in time

Error dynamics

- it is useful to look at the difference between the estimated state and the actual state
- we define the following errors

$$\check{\mathbf{e}}_k = \check{\mathbf{x}}_k - \mathbf{x}_k \tag{36a}$$
$$\hat{\mathbf{e}}_k = \hat{\mathbf{x}}_k - \mathbf{x}_k \tag{36b}$$

- plugging in the estimator and the system model we have

$$\check{\mathbf{e}}_k = \mathbf{A}_{k-1}\hat{\mathbf{e}}_{k-1} - \mathbf{w}_k \tag{37a}$$
$$\hat{\mathbf{e}}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k)\,\check{\mathbf{e}}_k + \mathbf{K}_k\mathbf{n}_k \tag{37b}$$

where we note that $\hat{\mathbf{e}}_0 = \hat{\mathbf{x}}_0 - \mathbf{x}_0$

Error dynamics: unbiased

- from the error dynamics, it is not hard to see that $E\left[\hat{\mathbf{e}}_k\right] = \mathbf{0}$ for $k > 0$ so long as $E\left[\hat{\mathbf{e}}_0\right] = \mathbf{0}$
- this means our estimator is unbiased
- to show it, we can use proof by induction: it is true for $k = 0$ by assertion; assume it is also true for $k - 1$; then

$$E\left[\check{\mathbf{e}}_k\right] = \mathbf{A}_{k-1} \underbrace{E\left[\hat{\mathbf{e}}_{k-1}\right]}_{\mathbf{0}} - \underbrace{E\left[\mathbf{w}_k\right]}_{\mathbf{0}} = \mathbf{0} \tag{38a}$$

$$E\left[\hat{\mathbf{e}}_k\right] = (\mathbf{1} - \mathbf{K}_k\mathbf{C}_k) \underbrace{E\left[\check{\mathbf{e}}_k\right]}_{\mathbf{0}} + \mathbf{K}_k \underbrace{E\left[\mathbf{n}_k\right]}_{\mathbf{0}} = \mathbf{0} \tag{38b}$$

- it is therefore true for all k

Error dynamics: consistent

- it is less obvious that

$$E\left[\check{\mathbf{e}}_k \check{\mathbf{e}}_k^T\right] = \check{\mathbf{P}}_k \tag{39a}$$

$$E\left[\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T\right] = \hat{\mathbf{P}}_k \tag{39b}$$

  for $k > 0$ so long as $E\left[\hat{\mathbf{e}}_0 \hat{\mathbf{e}}_0^T\right] = \hat{\mathbf{P}}_0$
- this means our estimator is consistent
- this means that the true uncertainty in the system is perfectly modelled by our estimate of the covariance, $\hat{\mathbf{P}}_k$
- in this sense, the Kalman filter is an optimal filter and why it is sometimes referred to as BLUE
- yet another way of saying this is that the covariance of the Kalman filter is right at the CRLB; we cannot be any more certain

Error dynamics: consistent

– to show this, we again use proof by induction: it is true for $k = 0$ by assertion; assume $E\left[\hat{\mathbf{e}}_{k-1}\hat{\mathbf{e}}_{k-1}^T\right] = \hat{\mathbf{P}}_{k-1}$; then

$$
\begin{aligned}
E\left[\check{\mathbf{e}}_k\check{\mathbf{e}}_k^T\right] &= E\left[\left(\mathbf{A}_{k-1}\hat{\mathbf{e}}_{k-1} - \mathbf{w}_k\right)\left(\mathbf{A}_{k-1}\hat{\mathbf{e}}_{k-1} - \mathbf{w}_k\right)^T\right] \\
&= \mathbf{A}_{k-1}\underbrace{E\left[\hat{\mathbf{e}}_{k-1}\hat{\mathbf{e}}_{k-1}^T\right]}_{\hat{\mathbf{P}}_{k-1}}\mathbf{A}_{k-1}^T - \mathbf{A}_{k-1}\underbrace{E\left[\hat{\mathbf{e}}_{k-1}\mathbf{w}_k^T\right]}_{\mathbf{0}\text{ by independence}} \\
&\quad - \underbrace{E\left[\mathbf{w}_k\hat{\mathbf{e}}_{k-1}^T\right]}_{\mathbf{0}\text{ by independence}}\mathbf{A}_{k-1}^T + \underbrace{E\left[\mathbf{w}_k\mathbf{w}_k^T\right]}_{\mathbf{Q}_k} \\
&= \check{\mathbf{P}}_k
\end{aligned}
\tag{40}
$$

Error dynamics: consistent

– and then

$$
\begin{aligned}
E\left[\hat{\mathbf{e}}_k \hat{\mathbf{e}}_k^T\right] &= E\left[\left(\left(\mathbf{1}-\mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{e}}_k + \mathbf{K}_k\mathbf{n}_k\right)\left(\left(\mathbf{1}-\mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{e}}_k + \mathbf{K}_k\mathbf{n}_k\right)^T\right] \\
&= \left(\mathbf{1}-\mathbf{K}_k\mathbf{C}_k\right)\underbrace{E\left[\check{\mathbf{e}}_k\check{\mathbf{e}}_k^T\right]}_{\check{\mathbf{P}}_k}\left(\mathbf{1}-\mathbf{K}_k\mathbf{C}_k\right)^T \\
&\quad + \left(\mathbf{1}-\mathbf{K}_k\mathbf{C}_k\right)\underbrace{E\left[\check{\mathbf{e}}_k\mathbf{n}_k^T\right]}_{\mathbf{0}\text{ by independence}}\mathbf{K}_k^T \\
&\quad + \mathbf{K}_k\underbrace{E\left[\mathbf{n}_k\check{\mathbf{e}}_k^T\right]}_{\mathbf{0}\text{ by independence}}\left(\mathbf{1}-\mathbf{K}_k\mathbf{C}_k\right)^T + \mathbf{K}_k\underbrace{E\left[\mathbf{n}_k\mathbf{n}_k^T\right]}_{\mathbf{R}_k}\mathbf{K}_k^T \\
&= \left(\mathbf{1}-\mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{P}}_k\underbrace{-\hat{\mathbf{P}}_k\mathbf{C}_k^T\mathbf{K}_k^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T}_{\mathbf{0}\text{ because }\mathbf{K}_k=\hat{\mathbf{P}}_k\mathbf{C}_k^T\mathbf{R}_k^{-1}} \\
&= \hat{\mathbf{P}}_k
\end{aligned}
\tag{41}
$$

– it is therefore true for all k

DARE we hope that the KF converges?

- on average, the KF converges to a unique solution under specific conditions – this implies the initial condition is forgotten over time
- consider the time-invariant case so that $\mathbf{A}_k = \boldsymbol{A}, \mathbf{C}_k = \boldsymbol{C}, \mathbf{Q}_k = \boldsymbol{Q}, \mathbf{R}_k = \boldsymbol{R}$
- at steady-state, the covariance, $\boldsymbol{P}$, must satisfy the Discrete Algebraic Riccati Equation (DARE)

$$\boldsymbol{P} = \boldsymbol{A}\left(1 - \boldsymbol{K}\boldsymbol{C}\right)\boldsymbol{P}\left(1 - \boldsymbol{K}\boldsymbol{C}\right)^T\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T\boldsymbol{A}^T + \boldsymbol{Q} \quad (42)$$

which comes from combining the prediction and correction equations for covariance

KF convergence

- the DARE has a unique positive-semidefinite solution, $\boldsymbol{P}$, if and only if the following conditions hold:
  - $\boldsymbol{R} > 0$; note, we already assume this in the batch linear-Gaussian case
  - $\boldsymbol{Q} \geq 0$; in the batch linear-Gaussian case we actually assumed that $\boldsymbol{Q} > 0$, whereupon the next condition is redundant
  - $(\boldsymbol{A}, \boldsymbol{V})$ is stabilizable with $\boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{Q}$; this condition is redundant when $\boldsymbol{Q} > 0$
  - $(\boldsymbol{A}, \boldsymbol{C})$ is detectable; same as observable except any unobservable eigenvalues are stable; we saw a similar observability condition in the batch linear-Gaussian case
- the proof of the above statement is beyond the scope of our course

KF convergence

- once the covariance evolves to its steady-state value, $\boldsymbol{P}$, so does the Kalman gain
- let $\boldsymbol{K}$ be the steady-state value of $\boldsymbol{K}_k$, which becomes

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{C}^T \left(\boldsymbol{C}\boldsymbol{P}\boldsymbol{C}^T + \boldsymbol{R}\right)^{-1} \tag{43}$$

- the error dynamics of the filter are then stable:

$$E[\check{\mathbf{e}}_k] = \underbrace{\boldsymbol{A}\left(\mathbf{1} - \boldsymbol{K}\boldsymbol{C}\right)}_{\text{eigs. } < 1 \text{ in mag.}} E[\check{\mathbf{e}}_{k-1}] \tag{44}$$

KF convergence

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

$$\begin{aligned} \mathbf{v}^T \boldsymbol{P} \mathbf{v} &= \underbrace{\mathbf{v}^T \boldsymbol{A} (\mathbf{1} - \boldsymbol{KC})}_{\lambda \mathbf{v}^T} \boldsymbol{P} \underbrace{(\mathbf{1} - \boldsymbol{KC})^T \boldsymbol{A}^T \mathbf{v}}_{\lambda \mathbf{v}} \\ &\quad + \mathbf{v}^T \left(\boldsymbol{AKRK}^T \boldsymbol{A}^T + \boldsymbol{Q} \right) \mathbf{v} \end{aligned} \tag{45a}$$

$$(1 - \lambda^2) \underbrace{\mathbf{v}^T \boldsymbol{P} \mathbf{v}}_{>0} = \underbrace{\mathbf{v}^T \left(\boldsymbol{AKRK}^T \boldsymbol{A}^T + \boldsymbol{Q} \right) \mathbf{v}}_{>0} \tag{45b}$$

which means that we must have $|\lambda| < 1$, and thus the steady-state error dynamics are stable (easy to see when $\mathbf{Q} > 0$)

Summary

- we have looked at the Kalman filter, a recursive solution to the linear-Gaussian estimation problem that is appropriate to online use
- there are several different ways to get to the KF: RTS forwards pass, MAP, full Bayesian, gain optimization
- all these methods result in the KF because the true posterior is exactly Gaussian and the mean and mode are one and the same
- the KF can be shown to converge under certain conditions (similar to batch uniqueness test)
- the KF is BLUE, it performs right at the CRLB