

Recursive Estimation

Shizheng Wen

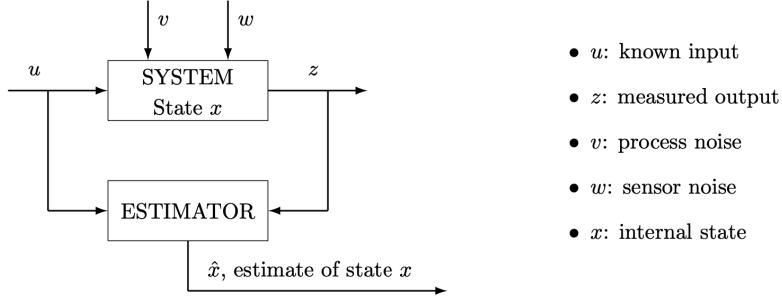
shiwen@student.ethz.ch

- 1 Introduction to State Estimation - Probability Review
- 2 Probability Review
- 3 Baye's Theorem
- 4 Bayesian Tracking
- 5 Extracting Estimates from Probability Distributions
 - 5.1 Maximum Likelihood (ML)
 - 5.2 Maximum a posteriori (MAP)
 - 5.3 Recursive Least Squares (RLS)
- 6 Kalman Filter I
 - 6.1 Model
 - 6.2 Gaussian Random Variable (GRV)
 - 6.2.1 Important of the Gaussian Property
 - 6.3 Problem Formulation and Auxiliary Variables
 - 6.3.1 Auxiliary variables
 - 6.4 Kalman Filter Equations
 - 6.4.1 Recap: Auxiliary variables
 - 6.4.2 Auxiliary variables are GRVs
 - 6.4.3 Mean and variance of auxiliary variables
 - 6.4.4 Summary (Kalman Filter equations)
 - 6.4.5 Alternative Equations
 - 6.5 Remarks
 - 6.6 Example
- 7 The Kalman Filter as State Observer
- 8 Extended Kalman Filter (unfinished)
- 9 Particle Filter I & II (unfinished)
- 10 Observer-Based Control and the Separation Principle (unfinished)
- 11 Probability
- 12 Basics of Probability
- 13 Bayes' Theorem
- 14 Bayesian Tracking
- 15 Extracting Estimates from Probability Distributions
- 16 The Kalman Filter
- 17 The Kalman Filter as State Observer
- 18 Particles Filter
- 19 Observer-Based Control and the Separation Principle

1 Introduction to State Estimation - Probability Review

Goal

Estimation of the state of a dynamic system based on a model and observations (sensor measurements), in a computationally efficient way.



Introduce two important rules (for Two):

- Marginalization (sum rule): 一起发生的概率(联合概率Joint PDF) 与单独发生的概率之间的关系

$$p_t(i) = \sum_{j=1}^L p_{ts}(i, j) \text{ and } p_s(j) = \sum_{i=1}^M p_{ts}(i, j)$$

- Conditioning (product rule): 一起发生的概率，一个发生另一个发生的概率（条件概率Conditional PDF），单独发生的概率三者之间的关系

$$p_{t|s}(i|j) := n_{ts}(i, j)/n_s(j), \text{ the likelihood of wearing trousers } i \text{ given that he is wearing shirt } j$$

$$p_{s|t}(j|i) := n_{ts}(i, j)/n_t(i), \text{ the likelihood of wearing shirt } j \text{ given that he is wearing trousers } i$$

Then it holds that

$$\begin{aligned} p_{ts}(i, j) &= \frac{n_{ts}(i, j)}{N} = \frac{n_{ts}(i, j)}{n_t(i)} \frac{n_t(i)}{N} = p_{s|t}(j|i) p_t(i) \\ &= \frac{n_{ts}(i, j)}{n_s(j)} \frac{n_s(j)}{N} = p_{t|s}(i|j) p_s(j). \end{aligned}$$

By the way, we can easily generalize this formula to multi-variable cases. And the case for continuous situation.

2 Proability Review

对这一问题讨论的前提是我们的世界上有很多的随机事件，对于每个随机事件我们都用一个随机变量去刻画，而描述这个随机变量我们用统计学事件讨论其发生的可能性。注意一个随机事件当然可以多次发生。其对应就是你从这个随机变量的分布中多次采样。

We have discussed the marginalization and conditioning for conditional PDFs of three random variables x, y, z .

- Marginalization: $p_{x|z}(\bar{x}|\bar{z}) = \sum_{\bar{y} \in \mathcal{Y}} p_{xy|z}(\bar{x}, \bar{y}|\bar{z})$ (Or integral if CRV)
- Conditioning: $p_{x|yz}(\bar{x}|\bar{y}, \bar{z}) = \frac{p_{xy|z}(\bar{x}, \bar{y}|\bar{z})}{p_{yz}(\bar{y}|\bar{z})}$

理解时基于intuition即可，用上衣，裤子，皮带当作三个随机变量来去理解。z永远在|后面，其实理解时可以去掉它，从two random variable的角度理解。

Notation:

- $p_{x|yz}(\bar{x}|\bar{y}, \bar{z})$: 在 y, z 发生的前提下， x 的发生概率。
 - intuition: = x, y, z 同时发生的概率 $\frac{1}{\text{所有可能的}} y, z$ 同时发生的概率

Next, we discussed the concept of **Independence**

- Random variables x, y are said to be independent. Two independent events之间的独立
- two variables x, y are said to be conditionally independent given z if $p(x|y, z) = p(x|z)$. 两个随机变量在 z 发生的情况下的独立：也就是在 z 发生的情况下，两个变量的发生不受影响。但是 z 不发生时，两个变量的发生是可能会受到影响的： $p(x, y|z) = p(x|z)p(y|z)$, but $p(x, y) \neq p(x)p(y)$ 。
 - **Conditional Independence \rightarrow independence** $p_{xy|z}(\bar{x}, \bar{y}|\bar{z}) = p_{x|z}(\bar{x}|\bar{z})p_{y|z}(\bar{y}|\bar{z}) \nrightarrow p_{xy}(\bar{x}, \bar{y}) = p_x(\bar{x})p_y(\bar{y})$
 $x = z + w_1, y = z + w_2$ where z, w_1, w_2 are mutually independent RVs.
 It holds: $p_{xy|z}(\bar{x}, \bar{y}|\bar{z}) = p_{x|z}(\bar{x}|\bar{z})p_{y|z}(\bar{y}|\bar{z})$
 However, $p_{xy}(\bar{x}, \bar{y}) \neq p_x(\bar{x})p_y(\bar{y})$, as both x, y depend on z .
 - **Conditional Independence \leftarrow independence** $p_{xy}(\bar{x}, \bar{y}) = p_x(\bar{x})p_y(\bar{y}) \nrightarrow p_{xy|z}(\bar{x}, \bar{y}|\bar{z}) = p_{x|z}(\bar{x}|\bar{z})p_{y|z}(\bar{y}|\bar{z})$
 $\mathcal{X} = \{0, 1\}$ and $\mathcal{Y} = \{0, 1\}$ are results of 2 independent coin tosses.
 $\rightarrow p_{xy}(\bar{x}, \bar{y}) = p_x(\bar{x})p_y(\bar{y})$
 However, given $\mathcal{Z} = \mathcal{X} + \mathcal{Y}$,
 $p_{xy|z}(\bar{x}, \bar{y}|\bar{z}) \neq p_{x|z}(\bar{x}|\bar{z})p_{y|z}(\bar{y}|\bar{z})$

The reason why we discussed independence is because it really simplifies algorithms and allows us to decouple information and processes.

Then we discussed the definition of **Expected Value**: 也就是对于满足一定分布的随机变量而言。你从中进行采样，你抱有的期待能是多少。weighted sum of values, with probability being the weight. (期望不能每次采样都实现，在无数次采样中求平均就是我们的期望。)

- If $y = g(x)$ and we know the distribution of X , then we can calculate $E_y[y]$ without calculating the PDF of y
 $\therefore E_y[y] = \sum_{\bar{y} \in \mathcal{Y}} \bar{y} p_y(\bar{y}) = \sum_{\bar{x} \in \mathcal{X}} g(\bar{x}) p_x(\bar{x})$ (or integral if CRV)

Mean and Variance

- $E[x]$ is also called the **mean**, generally a vector (x is a vector.)
- $\text{Var}[x] := E[(x - E[x])(x - E[x])^T]$: variance就是刻画单个采样相比于均值的差异

Sample a distribution

Most math libraries have functions that can generate uniformly distributed, random real numbers in the range $(0, 1)$ ¹. Here, we present four algorithms for generating samples of random variables with arbitrary PDFs using samples of CRVs with uniform distributions.

- One discrete random variable:

- $\hat{F}_x(\bar{x}) := \sum_{i=-\infty}^{\bar{x}} \hat{p}_x(i)$: cumulative distribution function (CDF).

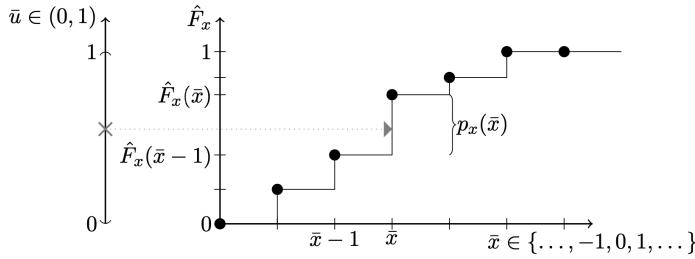


Figure 2.1: Sampling a discrete distribution.

- - randomly generate u . Solve for \bar{x} such that $\hat{F}_x(\bar{x}-1) < u$ and $u \leq \hat{F}_x(\bar{x})$.

Proof:

First, notice that we can always solve for such an \bar{x} (see Fig. 2.1).

For a fixed \bar{x} , determine the values of u for which $\hat{F}_x(\bar{x}-1) < u$ and $u \leq \hat{F}_x(\bar{x})$: We have

$$u \leq \hat{F}_x(\bar{x}) = \hat{F}_x(\bar{x}-1) + \hat{p}_x(\bar{x}) \quad \text{and, thus, } \hat{F}_x(\bar{x}-1) < u \leq \hat{F}_x(\bar{x}-1) + \hat{p}_x(\bar{x}).$$

The probability of drawing a sample \bar{x} is equal to the probability of u being in $(\hat{F}_x(\bar{x}-1), \hat{F}_x(\bar{x}))$, that is

$$p_x(\bar{x}) = \int_{\hat{F}_x(\bar{x}-1)}^{\hat{F}_x(\bar{x})} p_u(\lambda) d\lambda = \int_{\hat{F}_x(\bar{x}-1)}^{\hat{F}_x(\bar{x})} 1 d\lambda = \int_{\hat{F}_x(\bar{x}-1)}^{\hat{F}_x(\bar{x}-1) + \hat{p}_x(\bar{x})} 1 d\lambda = \hat{p}_x(\bar{x})$$

as desired. That is, the random variable $x \in \mathcal{X}$ has PDF $p_x = \hat{p}_x$.

- 证明的关键就是知道如果按照上述公式取 \bar{x} , 那么The probability of drawing a sample \bar{x} is equal to the probability of u in $(\hat{F}_x(\bar{x}-1), \hat{F}_x(\bar{x}))$. 然后我们在该范围上积分一做推导发现就是想要的desired propersity。

- More than one discrete random variable: **有两种方法**

- Option 1: only for \mathcal{X}, \mathcal{Y} are finite. $\hat{p}_z(1) = \hat{p}_{xy}(1, 1), \hat{p}_z(2) = \hat{p}_{xy}(1, 2), \dots, \hat{p}_z(N_x N_y) = \hat{p}_{xy}(N_x, N_y)$
 - 将一个多维的采样问题拉成一个一维的采样问题, 然后再用上面单采样的方法
- Option 2: works for both finite and infinite number of elements.
 -

Option 2: This works for both finite and infinite number of elements. Decompose $\hat{p}_{xy}(\bar{x}, \bar{y}) = \hat{p}_{x|y}(\bar{x}|\bar{y}) \hat{p}_y(\bar{y})$. Apply A1 to first get a sample \bar{y} for y via $\hat{p}_y(\bar{y})$, then with \bar{y} fixed, apply A1 again to get a sample \bar{x} for x via $\hat{p}_{x|y}(\bar{x}|\bar{y})$. Note that the independence of the uniform number generator between successive calls is key (**PSET 1: P25**).

- One continuous random variable: $\hat{F}_x(\bar{x}) := \int_{-\infty}^{\bar{x}} \hat{p}_x(\lambda) d\lambda, \bar{u} = \hat{F}_x(\bar{x})$.
- More than one continuous random variable: same as before.

Change of variables

We often encounter situations where we have to calculate the PDFs of functions of random variables. We will focus on one random variable; similar formulas can be derived for joint random variables. **!**: 容易和上文Expect value的概念混淆: $E[y] = \sum_{\bar{y} \in \mathcal{Y}} \bar{y} p_y(\bar{y}) = \sum_{\bar{x} \in \mathcal{X}} g(\bar{x}) p_x(\bar{x})$ (or integral if CRV)。此外, 连续随机变量的结论和离散变量的结论完全不同, 不能类比。

- Discrete Random Variable

—
Let p_y be given. Consider the function $x = g(y)$ – the goal is to calculate p_x . Let $\mathcal{X} = g(\mathcal{Y})$. For each $\bar{x} \in \mathcal{X}$, let $\mathcal{Y}_{\bar{x}} = \{\bar{y}_i | \bar{y}_i \in \mathcal{Y}, g(\bar{y}_i) = \bar{x}\}$. We claim that

$$p_x(\bar{x}) = \sum_{\bar{y} \in \mathcal{Y}: g(\bar{y})=\bar{x}} p_y(\bar{y}).$$

- Continuous Random Variable

—
This is a bit more complicated. Consider the function $x = g(y)$ – the goal is to calculate p_x . We will assume that $g(y)$ is continuously differentiable and strictly monotonic, and that p_y is continuous. Strictly monotonic means that $\frac{dg}{dy}(\bar{y}) > 0$ for all \bar{y} or that $\frac{dg}{dy}(\bar{y}) < 0$ for all \bar{y} . Assume that $\frac{dg}{dy}(\bar{y}) > 0$ for all \bar{y} , the other case is similar (**PSET 1: P8**). We claim that

$$p_x(\bar{x}) = \frac{p_y(\bar{y})}{\frac{dg}{dy}(\bar{y})},$$

where $\bar{y} = g^{-1}(\bar{x})$.

This gives us another way for generating samples of a PDF:

- Given a desired PDF \hat{p}_x , and a method for sampling p_y , find a function $x = g(y)$ such that

$$\frac{dg}{dy}(\bar{y}) = \frac{p_y(\bar{y})}{\hat{p}_x(g(\bar{y}))}.$$

Equivalently, solve $\frac{d\bar{x}}{d\bar{y}} = \frac{p_y(\bar{y})}{\hat{p}_x(\bar{x})}$, a differential equation.

- 这种方法不是扯淡吗，还要解微分方程，根本用不上。

Example: Let y be uniformly distributed on $[0, 1]$. Then $p_y(\bar{y}) = 1$ for all $\bar{y} \in [0, 1]$. Solve

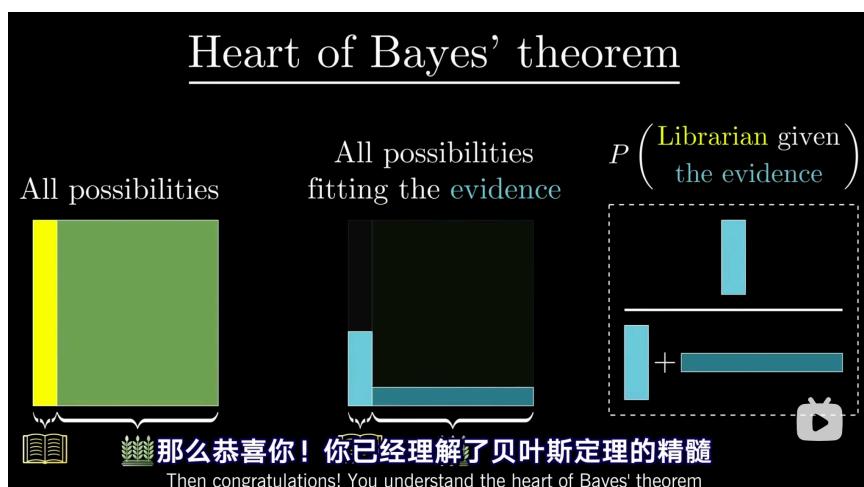
$$\frac{d\bar{x}}{d\bar{y}} = 1/\hat{p}_x(\bar{x}) \Rightarrow d\bar{y} = d\bar{x} \hat{p}_x(\bar{x}) \Rightarrow \bar{y} = \int_{-\infty}^{\bar{x}} \hat{p}_x(\lambda) d\lambda = \hat{F}_x(\bar{x}).$$

This is algorithm **A3!**

- — 解微分方程是为了得到 x ，其实就是上面的方法。

3 Baye's Theorem

New evidence does not completely determine your beliefs in a vacuum; it should update prior beliefs.



When to use Bayes' rule

You have a hypothesis

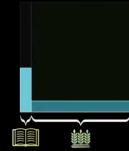


You've observed some evidence

Steve is very shy and withdrawn, invariably helpful but with very little interest in people or in the world of reality. A meek and tidy soul, he has a need for order and structure, and a passion for detail.

You want $P(H|E)$

$P(\text{Hypothesis given the evidence})$



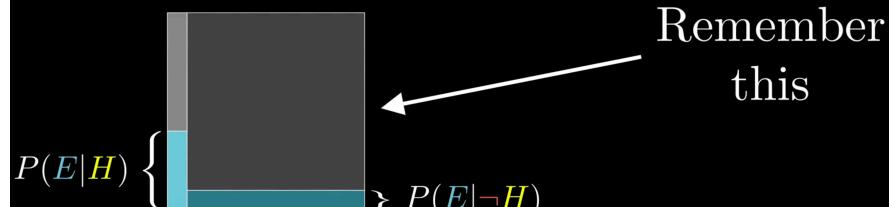
- Hypothesis: 假设
- Evidence: 证据
- $P(H|E)$ 在基于这些证据的前提下，我们假设成立的可能性。需要注意的是，新的证据并不能直接决定你的 beliefs，新的证据只是用来更新prior knowledge
- Prior $P(H)$ 考虑新证据evidence前，假设成立的可能性。
- likelihood $P(E|H)$ 在假设成立的前提下，我们看到证据的概率
- $P(E|\neg H)$ 在假设不成立的前提下，我们看到evidence的概率

$$\text{Bayes' theorem}$$

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} = \frac{P(H)P(E|H)}{P(H)P(E|H) + P(\neg H)P(E|\neg H)}$$

- $P(E)$: 我们看到所有证据的概率
- $P(H|E)$: Posterior 后验概率：基于evidence后，对prior做出的修正
- THE theorem lets you quantify and systematize the idea of changing beliefs
-

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} = \frac{P(H)P(E|H)}{P(H)P(E|H) + P(\neg H)P(E|\neg H)}$$



$P(E|H)$ { } $\rightarrow P(E|\neg H)$

but to, instead, draw out this diagram as needed



Proportions → Probability → Uncertainty

背景问题改变的是先验概率，而关于性格和印象的问题改变的是其对应的似然概率likelihoods（可能性）

4 Bayesian Tracking

Bayesian is a method used in many fields to track the state of an object, system, or process over time using Bayes' theorem. The key idea is updating our belief or estimate about the state based on new evidence or observations.

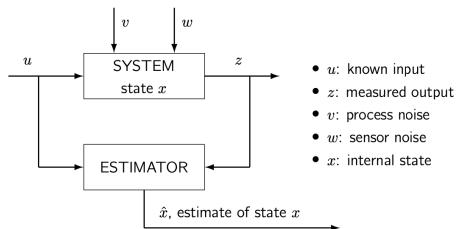
The algorithm works in two steps: prediction and update.

1. **Prediction:** Using the previous state and a model of how the state evolves over time (known as the transition model), a prediction is made about the current state.
2. **Update (or Correction):** Once an actual observation is made, this predicted state is updated to reflect the observed data. This is done using Bayes' theorem, hence the name "Bayesian tracking".

An important thing to note is that each prediction or estimate includes an associated uncertainty. As more observations are made, the estimates are adjusted and their uncertainties are typically reduced.

One common application of Bayesian tracking is in robotics for localization (determining a robot's position in an environment), and for tracking objects over time in computer vision. A famous example of Bayesian tracking is the Kalman Filter, which is used in many fields such as navigation, computer vision, and even economics.

你的Goal是得到系统在任意离散时间上的state。但是你不能直接对state做估计，虽然你有process model。主要原因是process model中会引入process error，以及 u_0 也是受到噪声影响，因此想直接通过理论的process model得知state不太准确。因此我们想引入对系统的一些measurement，然后这些measurement如果能直接测量state，那就再好不过了，我们直接得知state，但如果不能直接测量state，而是测量与state有关的量，那我们需要做一些推导，这就是所谓的measurement model，同时，在measurement的时候，我们也会受到measurement noise的影响。因此我们也需要将这个考虑进来。



Nonlinear, discrete-time system:

- $x(k) = q_{k-1}(x(k-1), u(k-1), v(k-1)), k = 1, 2, \dots$
- $z(k) = h_k(x(k), w(k))$ (when you do the measurement, you will introduce the measurement noise.)

Where $x(0), \{v(\cdot)\}, \{w(\cdot)\}$ have a probabilistic description.

Estimator has its own state: $\hat{x}(k)$, the estimate of $x(k)$ at time k .

Will compute $\hat{x}(k)$ from $\hat{x}(k-1)$, $u(k-1)$, $z(k)$, and model knowledge (dynamic model and probabilistic noise models).

No need to keep track of the complete time history $\{u(\cdot)\}$, $\{z(\cdot)\}$.

Recursive state estimation - finite state space

- The prior update, where the state estimate is predicted forward using the **process model**.

–

$$p(x(k)|z(1:k-1)) = \sum_{x(k-1) \in \mathcal{X}} p(x(k)|z(1:k-1), x(k-1)) \underbrace{p(x(k-1)|z(1:k-1))}_{\text{assumed to be known}}.$$

- As a result, it can be concluded that $x(k)$ and $z(1 : k - 1)$ are conditionally independent, given $x(k - 1)$.
- The prior update is therefore given by: $p(x(k)|z(1:k-1)) = \sum_{x(k-1) \in \mathcal{X}} p(x(k)|x(k-1)) p(x(k-1)|z(1:k-1))$
- This is an intuitive result: we use the process model to push our estimate forward in time. Note that conditional independence is crucial.

- The measurement update, where the prior is combined with observations/measurements (using Bayes' theorem).

$$\begin{aligned} p(x(k)|z(1:k)) &= p(x(k)|z(k), z(1:k-1)) \\ \text{– By Baye's rule, } &= \frac{p(z(k)|x(k), z(1:k-1)) p(x(k)|z(1:k-1))}{p(z(k)|z(1:k-1))}. \end{aligned}$$

- Using the above, the measurement update reads:

◦

$$p_{x(k)|z(1:k)}(\bar{x}(k)|\bar{z}(1:k)) = \frac{p_{z(k)|x(k)}(\bar{z}(k)|\bar{x}(k)) p_{x(k)|z(1:k-1)}(\bar{x}(k)|\bar{z}(1:k-1))}{\sum_{i \in \mathcal{X}} p_{z(k)|x(k)}(\bar{z}(k)|i) p_{x(k)|z(1:k-1)}(i|\bar{z}(1:k-1))}$$

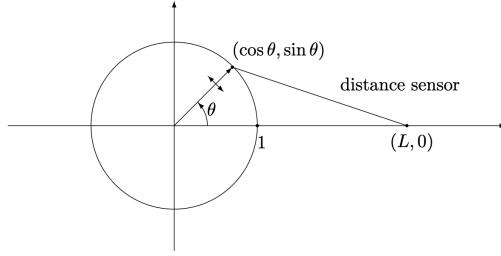
Summary:

$$\text{Prior update (state prediction): } p(x(k)|z(1:k-1)) = \sum_{x(k-1) \in \mathcal{X}} \overbrace{p(x(k)|x(k-1))}^{\text{process model}} \overbrace{p(x(k-1)|z(1:k-1))}^{\text{previous iteration}}, \quad k = 1, 2, \dots$$

$$\text{Measurement update: } p_{x(k)|z(1:k)}(\bar{x}(k)|\bar{z}(1:k)) = \frac{\overbrace{p_{z(k)|x(k)}(\bar{z}(k)|\bar{x}(k))}^{\text{measurement model}} \overbrace{p_{x(k)|z(1:k-1)}(\bar{x}(k)|\bar{z}(1:k-1))}^{\text{prior}}}{\underbrace{\sum_{i \in \mathcal{X}} p_{z(k)|x(k)}(\bar{z}(k)|i) p_{x(k)|z(1:k-1)}(i|\bar{z}(1:k-1))}_{\text{normalization}}}$$

Example

Consider an object moving randomly on a unit circle in discrete steps. We can measure the distance to the object and want to estimate its location.



5 Extracting Estimates from Probability Distributions

核心论点就是，在前面的章节中，我们挺你讨论了 how to compute PDF of an unknown quantity of interest (denoted by x) conditioned on given observations (denoted by z). 尽管 $p_{x|z}(\bar{x}|\bar{z})$ captures the full information that one has about x given z in the Bayesian sense. 我们更感兴趣的是如何从这个分布中得到一个具体的值（标量或者矢量）从而来刻画整个概率密度分布。

目前感觉这个地方的逻辑更像是你有一堆data，这个data满足某个特定的统计学分布，但是具体的参数我们并不知道，我们要做的是从这些data推算出最大可能的统计学分布是啥。

5.1 Maximum Likelihood (ML)

This method is often used when $x \in \mathcal{X}$ is an unknown (constant) parameter without a (known) probabilistic description $p_x(\bar{x})$. Let $z \in \mathcal{Z}$ be the measurement with observation model $p_{z|x}(\bar{z}|\bar{x})$. For a given observation \bar{z} , the method seeks the value for the parameter x that makes the observation \bar{z} most likely; that is,

$$\hat{x}^{\text{ML}} := \arg \max_{\bar{x} \in \mathcal{X}} p_{z|x}(\bar{z}|\bar{x}).$$

- x is an unknown (constant) parameter without a (known) probabilistic description. 没有process model?
- 我们仅仅只根据observation model $p_{z|x}(\bar{z}|\bar{x})$ 来进行求解，目标就是使得这种observation最有可能发生。

Example:

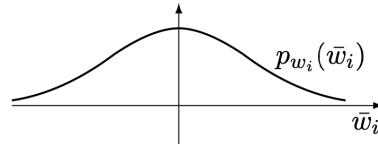
Consider two measurements of a scalar quantity $x \in \mathbb{R}$,

$$\begin{aligned} z_1 &= x + w_1 \\ z_2 &= x + w_2, \end{aligned}$$

where w_1 and w_2 are two normally distributed CRVs with zero mean and unit variance; that is,

$$p_{w_i}(\bar{w}_i) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\bar{w}_i^2}{2}\right)$$

(Shorthand: $w_i \sim \mathcal{N}(0, 1)$).



$$p_{z_1 z_2 | x}(\bar{z}_1, \bar{z}_2 | \bar{x}) = \frac{1}{2\pi} \exp\left(-\frac{1}{2} ((\bar{z}_1 - \bar{x})^2 + (\bar{z}_2 - \bar{x})^2)\right).$$

Differentiating with respect to \bar{x} and setting to 0 yields,

$$(\bar{z}_1 - \bar{x}) + (\bar{z}_2 - \bar{x}) = 0 \iff \bar{x} = \frac{\bar{z}_1 + \bar{z}_2}{2}.$$

That is, the ML estimate is the average (in this example).

- 相当于我们只要likelihood function即可求得我们的final solution，使得我们likelihood function最大。

- 我们通过observation来对系统的state做一个估测。

5.2 Maximum a posteriori (MAP)

We can use the MAP estimate when x is a random variable with a known PDF. We already know how to compute the posterior distribution by using Bayes' rule,

$$p_{x|z}(\bar{x}|\bar{z}) = \frac{p_{z|x}(\bar{z}|\bar{x}) p_x(\bar{x})}{p_z(\bar{z})}.$$

Then $\hat{x}^{\text{MAP}} := \arg \max_{\bar{x} \in \mathcal{X}} p_{z|x}(\bar{z}|\bar{x}) p_x(\bar{x})$.

What choice of parameters are the most likely ones, given the observations and the prior belief about x ?

- 先验知识 $p_x(\bar{x})$ 来源于 process model。
- 相比于 likelihood, 似乎这里是我们有 prior belief。

Example:

Consider the scalar observation

$$z = x + w \quad \text{with } w \sim \mathcal{N}(0, 1), x \sim \mathcal{N}(\mu, \sigma^2), x \text{ and } w \text{ independent.}$$

Then

$$p_x(\bar{x}) \propto \exp\left(-\frac{1}{2} \frac{(\bar{x} - \mu)^2}{\sigma^2}\right) \quad \text{and} \quad p_{z|x}(\bar{z}|\bar{x}) \propto \exp\left(-\frac{1}{2} (\bar{z} - \bar{x})^2\right).$$

Applying Bayes' rule to compute $p_{x|z}(\bar{x}|\bar{z})$ and differentiating it with respect to \bar{x} , setting to 0, and solving for \bar{x} , we obtain,

$$\bar{x} = \frac{1}{1 + \sigma^2} \mu + \frac{\sigma^2}{1 + \sigma^2} \bar{z}, \quad \text{a weighted sum.}$$

Notice the following special cases,

$$\begin{aligned} \sigma = 0 : & \quad \bar{x} = \mu \quad (\text{maximum of prior}) \\ \sigma \rightarrow \infty : & \quad \bar{x} = \bar{z} \quad (\text{ML}). \end{aligned}$$

5.3 Recursive Least Squares (RLS)

This section is a prelude to the standard way that the Kalman Filter is often derived. This approach bypasses PDFs and works directly with mean and variance.

Problem

Observation model

$$z(k) = H(k)x + w(k) \quad \text{with } z(k), w(k) \in \mathbb{R}^m, x \in \mathbb{R}^n.$$

- Prior knowledge: mean and variance of x , $\hat{x}_0 := \text{E}[x]$ and $P_x := \text{E}[(x - \hat{x}_0)(x - \hat{x}_0)^T] = \text{Var}[x]$, are given.

Notice that we have no process model: x does not change (our knowledge of it changes, however).

- Measurement noise: zero-mean with known variance, $\text{E}[w(k)] = 0$, $R(k) := \text{Var}[w(k)]$.
- $x, w(1), w(2), \dots$ are mutually independent.
- Typically, $n > m$; that is, fewer equations than unknowns at any particular time.

The objective is to compute an estimate $\hat{x}(k)$ of x from the observations $\{\bar{z}(1), \bar{z}(2), \dots, \bar{z}(k)\}$ in the least squares sense (minimizing a quadratic error).

For this problem, standard weighted LS is collecting a whole bunch of data until time k and convert this to a standard, weighted least squares problem.

但是，这种方法很不好，因为每一次有新的测量输入进来我们都需要重新求解Least squares。

因此，Can we build an estimate efficiently; that is, update the current estimate of x recursively when a new measurement $\bar{z}(k)$ comes in?

$$\hat{x}(k) = \hat{x}(k-1) + K(k)(\bar{z}(k) - H(k)\hat{x}(k-1))$$

Summary (RLS algorithm)

Initialization: $\hat{x}(0) = \hat{x}_0$, $P(0) = P_x = \text{Var}[x]$

Recursion:

Observe: $\bar{z}(k)$

Update: $K(k) = P(k-1)H^T(k)(H(k)P(k-1)H^T(k) + R(k))^{-1}$

$\hat{x}(k) = \hat{x}(k-1) + K(k)(\bar{z}(k) - H(k)\hat{x}(k-1))$

$P(k) = (I - K(k)H(k))P(k-1)(I - K(k)H(k))^T + K(k)R(k)K^T(k)$

The matrices $K(k)$ and $P(k)$ can be pre-computed from the problem data P_x , $\{H(\cdot)\}$, and $\{R(\cdot)\}$.

- 在我们没有priori knowledge仅仅只为mean和variance的情况下，我们对一个时刻的state进行反复的measurement，这种方法告诉了我们当measurement增加时，如何update我们对 $x(k)$ 分步的认识。
- 这个地方很关键的一点是我们没有process model，也就是 x doesn't change。

6 Kalman Filter I

目前看来，Kalman Filter与第五章的内容似乎无关，主要是和第四章Bayesian Tracking的内容有关？

In contrast to the Bayesian Tracking algorithm, the state is now a continuous random variable.

Problem structure: linear system and Gaussian distributions

Goal: keeps track of the PDF of the system state conditioned on all past measurements, $p(x(k)|z(1:k))$, for linear time-varying systems with Gaussian process and measurement noise.

6.1 Model

$$x(k) = A(k-1)x(k-1) + u(k-1) + v(k-1) \quad (6.1)$$

$$z(k) = H(k)x(k) + w(k) \quad (6.2)$$

$x(k)$: state

$u(k)$: known control input

$v(k)$: process noise

$z(k)$: measurement

$w(k)$: sensor noise

The CRVs $x(0)$, $v(\cdot)$, and $w(\cdot)$ are mutually independent, Gaussian distributed (note that vector-valued Gaussian random variables will be introduced in Section 6.2, below):

$$\begin{aligned} x(0) &\sim \mathcal{N}(x_0, P_0), \text{ i.e. } x(0) \text{ has a Gaussian distribution with mean } x_0 \text{ and variance } P_0, \\ v(k) &\sim \mathcal{N}(0, Q(k)), w(k) \sim \mathcal{N}(0, R(k)). \end{aligned}$$

6.2 Gaussian Random Variable (GRV)

一个vector的高斯分布：

The PDF of a Gaussian distributed (also called normally distributed) vector CRV $y = (y_1, \dots, y_D)$ is given by

$$p(y) = \frac{1}{(2\pi)^{D/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(y - \mu)^T \Sigma^{-1} (y - \mu)\right),$$

where $\mu \in \mathbb{R}^D$ is the mean vector and $\Sigma \in \mathbb{R}^{D \times D}$ is the variance, a symmetric ($\Sigma = \Sigma^T$), positive definite matrix (i.e. all eigenvalues are strictly positive), and $\det(\Sigma)$ denotes the determinant of Σ .

这里的vector指的是一个数据有多个维度？也就是想我们在IML课上Dimension Reduction看到的数据，需要注意的是，我们这些数据的不同维度之间往往有correlation，这也造成了variance Σ 不应该是个对角矩阵？

Special case

这里我们考虑special case，也就是 Σ is a diagonal matrix

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_D^2 \end{bmatrix} \text{ and } \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_D \end{bmatrix}.$$

Then,

$$p(y) = \prod_{i=1,\dots,D} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(y_i - \mu_i)^2}{2\sigma_i^2}\right).$$

That is, the PDF simplifies to the product of D scalar GRVs. Note that the variables are mutually independent¹ if and only if Σ is diagonal.

- Note that the variables are mutually independent if and only if Σ is diagonal.

Jointly Gaussian Random Variables

Two random variables x and y , both of which may be vectors, are said to be **jointly Gaussian** if the joint vector random variable (x, y) is a GRV. Note that, if $x \sim \mathcal{N}(\mu_x, \Sigma_x)$ and $y \sim \mathcal{N}(\mu_y, \Sigma_y)$ are GRVs, then this **does not imply** in general that they are jointly Gaussian. However, if x and y are additionally independent, then,

$$\begin{aligned} p(x, y) &= p(x)p(y) \propto \exp\left(-\frac{1}{2}\left((x - \mu_x)^T \Sigma_x^{-1} (x - \mu_x) + (y - \mu_y)^T \Sigma_y^{-1} (y - \mu_y)\right)\right) \\ &= \exp\left(-\frac{1}{2}[(x - \mu_x)^T (y - \mu_y)^T] \begin{bmatrix} \Sigma_x^{-1} & 0 \\ 0 & \Sigma_y^{-1} \end{bmatrix} \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix}\right), \end{aligned}$$

which implies that they are jointly Gaussian.

- 刚好和上面的做对比，一个variable是个vector表示数据有多个维度，当然不同维度之间可能有关系，所以我们可能使用dimension reduction
- 但不同的variable则是两个截然不同的variable，这些variable当然也可能dependent。

6.2.1 Important of the Gaussian Property

The assumption of Gaussian random variables will allow us to derive the Kalman Filter as the analytical solution to the Bayesian state estimation problem. In particular, the following properties will be useful.

- 这些properties用在bayesian tracking会使得问题变得更简单。

Property 1: An affine transformation of a GRV is a GRV

Let y be a GRV, and let x be defined by $x = My + b$, with a constant matrix M and a constant vector b of appropriate dimensions. Then, x is a GRV.

$$x \sim \mathcal{N}(b + m\mu, m^2\sigma^2)$$

Property 2: A linear combination of two jointly GRVs is a GRV

Let x and y be **two jointly Gaussian random variables**, and let z be defined by $z = M_x x + M_y y$, where M_x and M_y are constant matrices of appropriate dimensions. Then, z is a GRV.

This property follows **directly from Property 1** above:

$$z = M\xi \quad \text{with } M = [M_x \ M_y] \text{ and } \xi = (x, y) \text{ a GRV.}$$

Hence, $z \sim \mathcal{N}(\mu_z, \Sigma_z)$.

As before, we can calculate μ_z and Σ_z directly. Let $x \sim \mathcal{N}(\mu_x, \Sigma_x)$ and $y \sim \mathcal{N}(\mu_y, \Sigma_y)$, and we assume that x and y are independent (for x and y jointly Gaussian, but not necessarily independent, see (PSET 4: P1)). Then,

$$\begin{aligned} \mu_z &= E[z] = M_x E[x] + M_y E[y] = M_x \mu_x + M_y \mu_y \\ \Sigma_z &= E[(z - \mu_z)(z - \mu_z)^T] \\ &= E[(M_x x - M_x \mu_x + M_y y - M_y \mu_y)(M_x x - M_x \mu_x + M_y y - M_y \mu_y)^T] \\ &= M_x E[(x - \mu_x)(x - \mu_x)^T] M_x^T + M_y E[(y - \mu_y)(y - \mu_y)^T] M_y^T \\ &\quad + M_x \underbrace{E[(x - \mu_x)(y - \mu_y)^T]}_{=0 \text{ (by independence)}} M_y^T + M_y \underbrace{E[(y - \mu_y)(x - \mu_x)^T]}_{=0 \text{ (by independence)}} M_x^T \\ &= M_x \Sigma_x M_x^T + M_y \Sigma_y M_y^T. \end{aligned}$$

6.3 Problem Formulation and Auxiliary Variables

和Bayesian Tracking, calculate $p_{x(k)|z(1:k)}$. The only difference is that we are now working with CRVs instead of DRVs, which can readily be accommodated for by replacing sums with integrals.

Step 1 (S1): Prior update (yields a priori state estimate).

$$p_{x(k)|z(1:k-1)}(\bar{x}(k)|\bar{z}(1:k-1)) = \int \underbrace{p_{x(k)|x(k-1)}(\bar{x}(k)|\lambda)}_{\text{process model}} \underbrace{p_{x(k-1)|z(1:k-1)}(\lambda|\bar{z}(1:k-1))}_{\text{previous iteration (S2)}} d\lambda \quad (6.3)$$

Step 2 (S2): Measurement update (yields a posteriori state estimate).

$$p_{x(k)|z(1:k)}(\bar{x}(k)|\bar{z}(1:k)) = \frac{\underbrace{p_{z(k)|x(k)}(\bar{z}(k)|\bar{x}(k))}_{\text{measurement model}} \underbrace{p_{x(k)|z(1:k-1)}(\bar{x}(k)|\bar{z}(1:k-1))}_{\text{prior (S1)}}}{\underbrace{\int p_{z(k)|x(k)}(\bar{z}(k)|\lambda) p_{x(k)|z(1:k-1)}(\lambda|\bar{z}(1:k-1)) d\lambda}_{\text{normalization}} = p_{z(k)|z(1:k-1)}(\bar{z}(k)|\bar{z}(1:k-1))} \quad (6.4)$$

Next, we want to exploit the structure:

- Linearity
- Gaussian random variables

in order to convert the problem to pure matrix manipulations. (Note that all our problem data consists of matrices, so this is a reasonable goal.)

- 这样我们能够得到analytical solution。

6.3.1 Auxiliary variables

Define new random variables to simplify derivation: $x_p(k)$, $x_m(k)$, and $z_m(k)$, where the subscript “p” denotes “prediction” or “prior update”, and subscript “m” denotes “measurement” or “measurement update”:

$$\begin{aligned} \text{Init: } & x_m(0) := x(0) \\ \text{S1: } & x_p(k) := A(k-1)x_m(k-1) + u(k-1) + v(k-1) \\ \text{S2: } & z_m(k) := H(k)x_p(k) + w(k) \\ & x_m(k) \text{ defined via its PDF} \\ & p_{x_m(k)}(\xi) := p_{x_p(k)|z_m(k)}(\xi|\bar{z}(k)) \quad \forall \xi \end{aligned} \quad \left. \right\} \quad k = 1, 2, \dots$$

We now claim that for $k = 1, 2, \dots$,

$$p_{x_p(k)}(\xi) = p_{x(k)|z(1:k-1)}(\xi|\bar{z}(1:k-1)) \quad \forall \xi \quad (6.5)$$

$$p_{x_m(k)}(\xi) = p_{x(k)|z(1:k)}(\xi|\bar{z}(1:k)) \quad \forall \xi. \quad (6.6)$$

That is, $x_p(k)$ is the random variable $x(k)$ conditioned on $z(1:k-1)$, and $x_m(k)$ is the random variable $x(k)$ conditioned on $z(1:k)$.

- 相当于就是做了个变量替换，把本来的bayseian tracking变得更简单了。同时把process model和measurement model也很好的集成了进来，不是什么新的东西。

What comes next: efficiently calculating the mean and variance of $x_p(k)$ and $x_m(k)$. If we can additionally show that $x_p(k)$ and $x_m(k)$ are GRVs, we have a full characterization of their PDFs, completing our task.

- 也就是说，如果我们可以论证出state始终是保持Gaussian distribution变化的话，那么我们可以在系统forwarding的时候只存储mean和variance，从而来更好地做步进。

6.4 Kalman Filter Equations

6.4.1 Recap: Auxiliary variables

$x_p(k)$ and $x_m(k)$ capture the prior update and the measurement update, respectively, of the Bayesian state estimator. And in this chapter, we will first show that $x_p(k)$ and $x_m(k)$ are GRVs. As a consequence, their PDFs are fully characterized by mean and variance. The resulting recursive equations for mean and variance are the Kalman filter equations.

Notations:

$$\begin{aligned}\hat{x}_p(k) &:= \text{E}[x_p(k)], & P_p(k) &:= \text{Var}[x_p(k)], \\ \hat{x}_m(k) &:= \text{E}[x_m(k)], & P_m(k) &:= \text{Var}[x_m(k)].\end{aligned}$$

6.4.2 Auxiliary variables are GRVs

接下来，我们要做的第一件事是证明Auxiliary variable are GRVs:

Due to iterative form, we can prove by induction

- jointly gaussian: two variables are GRV and independent with each other.

即使基于GRV的一些properties进行简单的推导。其中我不太理解的是 $p_{z_m}(\bar{z})$ is a number

Since x_p is a GRV (proved in S1 above), we have

$$p_{x_p}(\xi) \propto \exp\left(-\frac{1}{2}(\xi - \hat{x}_p)^T P_p^{-1} (\xi - \hat{x}_p)\right)$$

and, by definition of z_m ,

$$p_{z_m|x_p}(\bar{z}|\xi) \propto \exp\left(-\frac{1}{2}(\bar{z} - H\xi)^T R^{-1} (\bar{z} - H\xi)\right).$$

Since \bar{z} denotes an actual measurement, $p_{z_m}(\bar{z})$ is a number (and does not depend on ξ); we can therefore write:

$$p_{x_m}(\xi) \propto \exp\left(-\frac{1}{2}\left((\xi - \hat{x}_p)^T P_p^{-1} (\xi - \hat{x}_p) + (\bar{z} - H\xi)^T R^{-1} (\bar{z} - H\xi)\right)\right). \quad (6.7)$$

Note that the argument of the exponential is a quadratic in ξ . Therefore, there exist μ and Σ such that

$$p_{x_m}(\xi) \propto \exp\left(-\frac{1}{2}(\xi - \mu)^T \Sigma^{-1} (\xi - \mu)\right). \quad (6.8)$$

That is, $x_m(k)$ is a GRV, which completes the induction.

6.4.3 Mean and variance of auxiliary variables

We compute mean and variance of $x_p(k)$ and $x_m(k)$.

Step1: derive the $\hat{x}_p(k)$ and $P_p(k)$ by using the properties of the expected value and the variance.

Step2: Relying on the properties of the expected value and the variance to calculate the (6.7) and compare the quadratic and linear terms with (6.8) to derive the $\hat{x}_m(k) = \mu$ and $P_m(k) = \Sigma$ of x_m . **This is our objective**

6.4.4 Summary (Kalman Filter equations)

The Kalman filter is given by the recursive update equations for mean and variance above.

Initialization: $\hat{x}_m(0) = x_0, P_m(0) = P_0$

Step 1 (S1): Prior update/Prediction step

$$\hat{x}_p(k) = A(k-1)\hat{x}_m(k-1) + u(k-1)$$

$$P_p(k) = A(k-1)P_m(k-1)A^T(k-1) + Q(k-1)$$

Step 2 (S2): A posteriori update/Measurement update step

Results from above, re-introducing time index k :

$$P_m(k) = (P_p^{-1}(k) + H^T(k)R^{-1}(k)H(k))^{-1}$$

$$\hat{x}_m(k) = \hat{x}_p(k) + P_m(k)H^T(k)R^{-1}(k)(\bar{z}(k) - H(k)\hat{x}_p(k))$$

- Kalman Filter is the analytic solution to the Bayesian state estimation problem for a linear system with Gaussian distributions.
- In update state, all the time step is k . To simplify notation we drop all " k " arguments.

6.4.5 Alternative Equations

An alternative form of the measurement update equations (can be derived using the matrix inversion lemma):

$$K(k) = P_p(k)H^T(k)(H(k)P_p(k)H^T(k) + R(k))^{-1} \quad (6.9)$$

$$\hat{x}_m(k) = \hat{x}_p(k) + K(k)(\bar{z}(k) - H(k)\hat{x}_p(k)) \quad (6.10)$$

$$P_m(k) = (I - K(k)H(k))P_p(k) \quad (6.11)$$

$$= (I - K(k)H(k))P_p(k)(I - K(k)H(k))^T + K(k)R(k)K^T(k) \quad (6.12)$$

- 这个地方，我们之所以要这面做的理由是将 $K(k)$ 分离出来，我们可以看到，在(6.10)里面，后验概率等于先验概率+（实际测量和理论测量之间的误差→来源于measurement noise的引入）再乘以Kalman Filter gain，像一个比例一样。而方差不受到measurement的影响。总之就是kalman filter gain的引入使得我们的问题更有一些物理和数学上的结构。
- (6.11)和(6.12)是等价的，作业题中还让证明两个一直。但是，这里之所以我们采用不同的表示是因为(6.12)在数值计算上更加的stable。**为什么更加stable不太清楚**
- The alternative equations are equivalent to equations on the summary.
- The matrix $K(k)$ is called the *Kalman Filter gain*.
- Note that implementing (6.12) is computationally more expensive than (6.11), but is less sensitive to numerical errors. This is known as the “Joseph form” of the covariance update.
- Derivation process:

—
Matrix inversion lemma: If A , D , and $D^{-1} + CA^{-1}B$ are nonsingular, then $A + BDC$ is nonsingular and

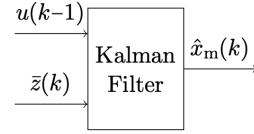
$$(A + BDC)^{-1} = A^{-1} - A^{-1}B(D^{-1} + CA^{-1}B)^{-1}CA^{-1}.$$

6.5 Remarks

Implementation

If the problem data is known ahead of time (i.e. $A(k)$, $H(k)$, $Q(k)$, and $R(k)$ for all k as well as P_0), all KF matrices (i.e. $P_p(k)$, $P_m(k)$, and $K(k)$) can be computed off-line. In particular, the KF gain does not depend on real-time measurement data!

$$\begin{aligned}\hat{x}_p(k) &= A(k-1)\hat{x}_m(k-1) + u(k-1) \\ \hat{x}_m(k) &= \hat{x}_p(k) + K(k)(\bar{z}(k) - H(k)\hat{x}_p(k)) \\ \hat{x}_m(k) &= (I - K(k)H(k))A(k-1)\hat{x}_m(k-1) \\ &\quad + (I - K(k)H(k))u(k-1) + K(k)\bar{z}(k) \\ &= \tilde{A}(k)\hat{x}_m(k-1) + \tilde{B}(k)u(k-1) + K(k)\bar{z}(k)\end{aligned}$$



Note that the filter is linear and time varying.

- 这个非常有用，也就是我们只需要知道input和measurement即可估计当下系统的后验state。

Another interpretation

还稍微有点不清楚这里的square loss指的是什么,L5在看看p4. p5

RLS就是没有process model，对当前状态反复做测量，从而update对当前状态的估计。因为每一次测量都是引入误差，我们反复地测量就可以使得当前估计的variance缩小，从而对当前态有更高的可信度。而在这个问题里面，我们系统是有演化的，所以存在process model。RLS可以看做事稳态的process问题，这里可以看作是动态的。我们都是估测当前的state情况。

所以，KF其实可以看作是minimize the mean square error。因此，将概念引申，如果不是linear的GR和linear的问题，用KF其实也合理。属于best linear estimator in the MMSE sense.

Note that the equations (6.9)–(6.12) are the recursive least squares (RLS) equations! This is a second interpretation of the KF, which also applies to non-Gaussian random variables: among the class of *linear, unbiased* estimators (specifically, those that have the structure $\hat{x}_m(k) = \hat{x}_p(k) + K(k)(\bar{z}(k) - H(k)\hat{x}_p(k))$), compare RLS lecture #5), the KF is the one that *minimizes the mean squared error*.

- For a linear system and Gaussian noise, the KF is the best you can do (it keeps track of the full conditional PDFs). The linear estimator structure is optimal.
- Otherwise, one must be careful. The KF is no longer optimal (nonlinear may do better), but often reasonable (it is the best *linear* estimator in the MMSE sense).

Remark: Positive definiteness of variance matrices

not well understand

In the derivation of the KF, we assumed that $x(0)$, $v(k)$, and $w(k)$ are GRVs and therefore have positive definite variance matrices (i.e. $P_0 > 0$, $Q(k) > 0$, $R(k) > 0$). Note that the KF equations also make sense when some of the variance matrices are positive semidefinite (i.e. ≥ 0), as long as the involved matrix inversions are well defined (in (6.9), for example, we require $H(k)P_p(k)H^T(k) + R(k) > 0$ in order to be invertible, similarly to the derivation of the RLS).

Examples:

- $P_0 = 0$ means that the initial state is perfectly known.
- $Q \geq 0$ with some zero eigenvalue means that there is no process noise in some direction of the state space.

6.6 Example

自己跑一下那个程序看看，感觉很有意思

7 The Kalman Filter as State Observer

Model:

That is, $A(k) = A$, $H(k) = H$, $Q(k) = Q$, and $R(k) = R$ are constant.

$$\begin{aligned}x(k) &= Ax(k-1) + u(k-1) + v(k-1) & x(0) \sim \mathcal{N}(x_0, P_0), v(k-1) \sim \mathcal{N}(0, Q) \\z(k) &= Hx(k) + w(k) & w(k) \sim \mathcal{N}(0, R)\end{aligned}$$

where $x(k) \in \mathbb{R}^n$, $z(k) \in \mathbb{R}^m$.

- Having application for system with equilibrium point.

$$\begin{aligned}x(k) &= ax(k-1) + v(k-1) & x(0) \sim \mathcal{N}(0, p_0), v(k-1) \sim \mathcal{N}(0, q) \\z(k) &= hx(k) + w(k) & w(k) \sim \mathcal{N}(0, r).\end{aligned}$$

We simulate the system for 30 iterations, by drawing a random initial condition and random noise sequences $v(k-1)$ and $w(k)$, $k = 1, 2, \dots, 30$.

unstable system

8 Extended Kalman Filter (unfinished)

9 Particle Filter I & II (unfinished)

10 Observer-Based Control and the Separation Principle (unfinished)

Recitation

1 Probability

2 Basics of Probability

3 Bayes' Theorem

- Core theorems are to use:
 - product rule $P_x(\bar{x}) = P_{x|y}(\bar{x}|\bar{y})P_y(\bar{y})$
 - sum rule $P_x(\bar{x}) = \sum_{\bar{y} \in y} P_{yx}(\bar{y}\bar{x})$
 - Bayes' theorem
- We need to clearly define our observation random variable and state variable.
- For three random variables, what should we clearly use such core theorems.

4 Bayesian Tracking

- Core task: tracking the state of an object, system, or process **over time** using Bayes' theorem. The key idea is updating our belief or estimate about the state based on new evidence or observations.
- 通过process mode和observation model都分别可以推得系统的state。而Bayesian tracking就是将这两种model的优势结合起来。在完整的bayesian tracking公式中，我们的所有都是基于概率密度分布。包括我们的process model和observation model。再下面的kalman filter，我们则考虑线性的系统情况且process noise和measurement noise都是高斯分布
- **rectitation最后两问还有点疑问？**

这里面比较tricky的点:

I am confused. You can get state of the system solely based on transition model. You can also get the state based on observation model without transition model. Each of these methods can get the state of the system. But we combine it and get a more robust method? I didn't get the core idea behind this method.

Chatgpt:

I understand your confusion. Let's take a step back and look at the bigger picture.

In the real world, systems we want to track or predict are often complex, noisy, and sometimes behave in ways that are hard to model perfectly. We therefore have to deal with uncertainties when we're trying to estimate the state of these systems.

The state transition model and the observation model serve two different, but complementary, purposes.

The state transition model represents our best understanding of how the system evolves over time, under ideal conditions. It can predict the next state of the system based on the current state. However, it can't always accurately capture real-world complexities and unforeseen influences. So, if we only rely on the state transition model, our state estimation may drift away from reality over time, especially in noisy or unpredictable environments.

The observation model, on the other hand, relates the observable measurements to the system state. Measurements, however, are usually noisy and might be indirectly related to the state. If we estimate the state based only on the latest observation, ignoring the system dynamics represented by the transition model, we may get a state estimation that fluctuates wildly due to measurement noise or fails to capture the actual evolution of the system.

So, each method on its own has limitations. But when we combine them in a Bayesian tracking framework, we get the best of both worlds. The state transition model gives us a prediction that takes into account the system's dynamics, and the observation model updates this prediction based on the latest data. In other words, the state transition model provides a prediction, and the observation model corrects this prediction.

Bayesian tracking combines these models in a mathematically optimal way under the principles of probability theory. This results in a state estimate that is more accurate and robust than what we would get by using either model alone. The estimate balances the prediction from the state transition model and the correction from the observation model, considering the uncertainties associated with both. This is the core idea behind Bayesian tracking.

5 Extracting Estimates from Probability Distributions

这里的maximum likelihood (ML)和我们上文的假设非常的直，也就是我们其实可以仅仅指通过measurement得到系统的state。然后我们的state应该是使得measurement可能性最大的情况。也就是我们仅计算 $P_{z|x}(\bar{z}|\bar{x})$ ，然后 x 为多少时这个概率最大。然后注意，最后得到的 \bar{x} 其实就是least squares (LS)!

We can thus give least squares a statical interpretation: it is the maximum likelihood estimate when the errors are independent, zero mean, same variance, and normal distributed.

$$z = Hx + w, \quad H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_m \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad H_i = [h_{i1} \cdots h_{in}], \quad h_{ij} \in \mathbb{R}, \quad \text{with } z, w \in \mathbb{R}^m, x \in \mathbb{R}^n, m \geq n, w_i \sim \mathcal{N}(0, 1).$$

- LS is designed for overdetermined LSE. Here, different H can be viewed as different data x . z can be viewed as corresponding y . w_i can be viewed as the error for every point. The number of points to measure is m .

Recursive LS:

Using data to estimate the state/parameter of the system. The state/parameter is often assumed to be fixed. However, in bayesian tracking, the state of the system is changing with time.

Recursive LS don't have the process model. We use observation model to help us define the state of the system. In Bayesian tracking, the state (changing with time) of the system is updated online.

核心来说，在这里讨论的三种方法中，都不存在process model。而且系统的state或者parameter是不随参数变化的。我们的目标就是基于数据去更新我们的state或者parameter。对于ML，其本质就是general LS，我们对state没有先验知识，寻找最好的state/parameter从而使得我们observation出现的可能性最大。而MAP无非就是对state有一些先验知识，所以我们考虑的是后验可能性最大。而Recursive LS则是一种online的LS。其核心就是求一个Kalman Gain，这个kalman gain就是表征每一次更新系统应该变化量。然后用这个kalman gain和新的observation来更新我们的estimates和covariance matrix。注意，ML和MAP更像是从统计的角度出发的，我们得到likelihood或者后验的概率密度分布，从而选择概率密度最大的点作为我们的解。而Recursive LS更像是一种deterministic的方法。

Rectitation还没做。

6 The Kalman Filter

Kalman filter的核心就是基于Gaussian distribution的一些特性，我们的state和observation model一直都也是 Gaussian Distribution，因此我们可以通过track mean and covariance matrix来追踪我们系统的state。

考虑一下covariance matrix是对角矩阵的时候系统的降维问题

在Kalman Filter的measurement update function中，其形式和RLS一模一样（其实也可以理解，本来RLS就是一个 observation model，没有transition model）。唯一不同的点是RLS迭代的是自己的结果。而Kalman Filter迭代的结果是还需要过一下process model的。

KF is used for state estimation in dynamic systems, and it uses models for the state transition and observation process. RLS, on the other hand, is typically used for parameter estimation in regression problems, where the model is a static relationship between input and output variables.

- KF: state estimation. RLS: parameter estimation.
- The KF can infer information about states that are not directly measured by exploiting possible couplings of the states through the system dynamics.

7 The Kalman Filter as State Observer

This chapter mainly detect under what situation, the error variances of all states converge.

Detectability and stabilizability (with respect to the process noise) are properties of the system, and not of the estimation algorithm. Hence, they cannot be altered by using a different state estimation algorithm, but only by modifying the system (for example, by placing an additional sensor.)

Currently, I know the Kalman filter is a special case (continuous state variable, linear observation and transition model and Gaussian distribution for any random variable - process nosie, initial state and observation noise) for Bayesian Tracking. In the observation update process for Kalman Filter, the

8 Particles Filter

Monte Carlo的核心就是通过sampling来显示一个随机变量的概率密度分布 (PDF) . It can be applied to discrete/continuous/change of variables cases. However, we have learnt before, if we want to achieve a sampling of a specific PDF, we can achieve it from a uniform distribution at (0,1).

The key idea of particles filter is that we use Monte Carlo sampling to track the distribution of the random variable. By tracking the variation of these variables, we can achieve the tracking of the random variables.

9 Observer-Based Control and the Separation Principle

Observer-based control: Like in the previous chapters, we used the full the state of the systems to achieve a control objective.

Output-based control: We only use output (measurement) for our control problem. Note that these measurements may not tell your the full state information of the system.

- Note that the Output-based control can also be model-based method. The only difference is that you use measurements (output) not full state information to achieve your control task.

Questions

1. Observability and Detectability

- (a) **Observability:** This concept is concerned with whether it is possible to determine the entire state of a system from its output measurements alone, without any knowledge of the internal system dynamics or state transitions. Essentially, if a system is observable, you can reconstruct the complete state vector just from the measurements (outputs) at any given time. This doesn't necessarily rely on the transition model but is related to the ability to measure all the states directly or indirectly through the outputs.
- (b) **Detectability:** Detectability is a slightly weaker concept than observability. It pertains to the ability to eventually estimate the hidden or unobservable states from the outputs over time, even if you can't determine them instantly from the outputs. This often relies on the state transition model, as the transition model allows you to infer the behavior of hidden states as they evolve over time. In a detectable system, any errors in estimating hidden states will diminish over time, and the states won't grow unbounded, allowing for meaningful estimation and control.

So in summary, observability deals with the ability to determine all states instantly from the outputs, while detectability is concerned with the ability to estimate hidden states over time, often relying on the state transition model. Both concepts are important in different contexts and serve as vital tools in the analysis and design of control systems.

Observed states won't meet the situation of stability

2. Detectability and Stability

(a)

3. Calculating the Kalman gain for the steady-state Kalman Filter

In the context of a steady-state Kalman Filter (also known as an infinite-horizon Kalman Filter), the Kalman gain settles to a constant value over time. This constant gain can be computed directly using the Algebraic Riccati Equation (ARE).

Here's a step-by-step guide to compute the steady-state Kalman gain for a linear system:

1. **Define the System:** You need to know the state transition matrix (A), the observation matrix (C), the process noise covariance (Q), and the measurement noise covariance (R).
2. **Solve the Algebraic Riccati Equation:** The steady-state error covariance matrix (P) satisfies the continuous-time ARE or discrete-time ARE, depending on your system.
 - For a continuous-time system, the ARE is:

$$APA^T - P + PC^T R^{-1} CP = 0$$
 - For a discrete-time system, the ARE is:

$$P = APA^T - APC^T(CPC^T + R)^{-1}CPA^T + Q$$
3. **Compute the Kalman Gain:** Using the solution (P) from the ARE, you can compute the steady-state Kalman gain (K) as:

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

In practice, solving the ARE can be done using specialized numerical algorithms available in many scientific computing environments, like MATLAB's `dare` or `care` functions for discrete-time and continuous-time systems, respectively.

Keep in mind that the steady-state Kalman gain is only valid if the system is both detectable and stabilizable. If these conditions are not met, the error covariance may not converge, and the concept of a steady-state Kalman gain might not apply.

Linear Quadratic Regulator (LQR):

$$J_{LQR} = \sum_{k=0}^{\infty} x^T(k) \bar{Q} x(k) + u^T(k) \bar{R} u(k)$$

$$u(k) = Fx(k) = Fz(k)$$

$$F = -(B^T P B + \bar{R})^{-1} B^T P A$$

Where $P = P^T \geq 0$

理解detectability, observability和stabilizability以及其与 P_∞ 的关系。

答：对于一个LTI线性系统，如果其是detectable和stabilizable，则存在 P_∞ 。而存在 P_∞ 的意义是告诉你，随着时间的演化，你的observer可以很好的追踪你的系统，对你系统state的observation可以被很好的控制在一个范围内，见下面chatgpt：

In the context of a steady-state Kalman Filter (SSKF), (P_∞) represents the error covariance matrix as time approaches infinity. It signifies the point at which the Kalman Filter has reached a steady state, and the error covariance matrix is no longer changing with time. (Kalman Gain是在observation model中体现出来的，只与 P_p 有关，而 P_p 又只于 A, Q 有关，且 P_m 只与 Kalman Gain 有关，注意data只与 $\bar{x}_m(k)$ 有关，也就是data用来更新state的值，但系统的error covariance则趋于一个稳定)

The meaning and significance of (P_∞) can be understood as follows:

1. **Convergence:** The existence of (P_∞) implies that the Kalman Filter's error covariance converges to a finite value as time goes on. This is a desirable property because it means that the estimation error reaches a stable, bounded value.
2. **Optimality:** At the steady state, the Kalman gain also reaches a constant value, denoted by (K_∞) . This gain represents the optimal trade-off between the prediction error and the measurement noise for the given system dynamics and noise characteristics.
3. **Simplification:** Once the system reaches the steady state, the Kalman gain remains constant, so it doesn't need to be recalculated at each time step. This can lead to computational efficiencies, especially in real-time applications.
4. **Detectability and Stabilizability:** The existence of (P_∞) can be related to properties like detectability and stabilizability of the system. If the system is both detectable and stabilizable, then (P_∞) exists, and the Kalman Filter error covariance will converge to this steady-state value.
5. **Design Insights:** Analyzing (P_∞) can provide insights into the design of the system. It can show how different system parameters and noise characteristics will affect the long-term estimation error.

In summary, (P_∞) in the steady-state Kalman Filter encapsulates important properties about the long-term behavior of the filter, including the convergence and optimality of the state estimation.

而一个系统是否stabilizability, 是否stable以及是否detectable和observable是另外的概念。

- observable意味着你的系统的full state可以通过output来完全推得，你不需要知道process model。而 detectable的意思则是说，如果有些state在output中无法观测到，我们可以结合process model从而对系统的 full state进行估测。这个上面也有chatgpt回答。

- 而一个系统stable则是说这个系统的state演化不会unbounded，也就是初始条件的扰动不会随着演化变得无限大。对于线性系统来说，系统的stable通常指的是A矩阵小于1，是收敛的，所以系统最后会converge到0或者类似的一个常数值。而stabilizability则指的是纵使你的系统中有的state不是stable的，但是我们可以通过input从而使得你系统能够stable，这不仅涉及到系统本身的问题，而是控制的问题了。见chatgpt回答：

No, the concepts of stability and stabilizability are not the same. Here's an explanation of both:

1. Stability:

- A system is considered stable if the state of the system does not grow unbounded over time. In other words, small disturbances will not lead to uncontrollable growth in the state variables.
- For linear time-invariant (LTI) systems, stability is often examined using the eigenvalues of the system matrix. If all eigenvalues have negative real parts (or are inside the unit circle for discrete-time systems), then the system is stable.

2. Stabilizability:

- Stabilizability is a property related to the control of a system. It refers to the ability to design a control input that can make the system stable, even if the system is not stable by itself.
- Essentially, a system is stabilizable if the uncontrollable modes (if any) are stable. In other words, if you can't control a particular state, it won't grow unbounded over time.
- Stabilizability is a weaker condition than controllability. A system that is controllable is also stabilizable, but the reverse is not necessarily true.

In summary, stability refers to the inherent behavior of the system without considering control inputs, while stabilizability is about the possibility of designing a control law to make the system stable.

此外就是注意，在stabilizability中涉及到uncontrollable state: A system is stabilizable if the uncontrollable modes (states that cannot be affected by the input) are stable on their own. Even though you can't control them directly, they won't cause the system to become unstable.

在detectable中涉及到unobservable state:

1. • Detectability is a concept related to observability, but it's slightly weaker. Observability refers to the ability to reconstruct the complete state of a system from its output measurements over finite time. Detectability, on the other hand, refers to the ability to determine the complete state over infinite time.
- A system is detectable if the unobservable modes (states that cannot be seen through the output) are stable. These states don't affect the system's output, but since they are stable, they won't cause the estimation error to grow unbounded over time.
- In mathematical terms, the system is detectable if the eigenvalues of the unobservable modes have negative real parts (or lie inside the unit circle for discrete-time systems).

In summary, stabilizability is concerned with the ability to control a system so that it doesn't become unstable, even if some states are uncontrollable. Detectability, on the other hand, is concerned with the ability to estimate the states of a system over infinite time, even if some states are unobservable. Both concepts are based on the inherent stability of the uncontrollable or unobservable modes of the system.

下面还有就是数学上，我们怎么确定这些unobservable和uncontrollable，他们是一样的嘛，以及他们怎么确定，这些还有问题。

进一步解释observability和detectability

Detectability and Observability are two related but different concepts. We did not cover observability in the lecture this year, but to answer your question:

- Detectability can be interpreted as all unstable modes being observable, whereas observability (as the name implies) means that all modes both stable and unstable, are observable. For observability we have to check the rank of $\begin{bmatrix} A - \lambda I \\ H \end{bmatrix}$ for all eigenvalues $\lambda \in \mathbb{C}$, whereas for detectability we only need to check the rank for unstable eigenvalues ($|\lambda| > 1$). So you are right, in part b) we need to check for all eigenvalues (also in this case $\lambda=0$) which is why we cannot use the proof of part a) But since we didn't cover observability this year, it will not be part of the exam.

Observability and controllability are stronger concepts than detectability and stabilizability; but we did not cover observability/controllability and thus **they will not be covered in this year's exam**.

But to answer your question in short: Observability/detectability and controllability/stabilizability are dual problems. Observability means we can observe all modes, while detectability means we can observe the unstable modes. Observability thus implies detectability. Similarly, controllability means we can drive the system to any desired state using our control input (at least under the absence of noise). Stabilizability means we can drive the system state to 0 using our control input (i.e., the unstable modes are controllable, since the stable modes will go towards 0 by themselves). Controllability thus implies stabilizability.

Modes relate to the eigenvectors of our system and describe behaviors of a system that behave independently. If e.g. for a system $x(k) = Ax(k-1) + Bu(k-1)$,

where A, B are **diagonal**, then the modes are simply given by each element of the state, i.e., x_1, x_2 , etc. That means that the state element x_1 is not influenced by the state elements x_2, x_3 , etc. Generally, using a suitably chosen linear coordinate transformation $\tilde{x}(k) = Tx(k)$, we can make the dynamics of each element in $\tilde{x}(k)$ independent of each other. You can thus think of modes as linear transformations of the original state which behave independently.

These topics go very far into (linear) systems theory, and there is an abundance of books/paper etc. that cover these topics in great detail, but this is outside the scope of this lecture. For observability etc. see for example Optimal State Estimation by Dan Simon. There is also the lecture "Linear Systems Theory", "Control SystemsI/II" which cover these topics. To understand modes of a system you can e.g. check out: http://www.cds.altech.edu/~murray/books/AM05/pdf/am06-linsys_16Sep06.pdf

1. 边界是 $[0,1]$ or $[0,1)$ 都无所谓 ↵