ASE 381P: Statistical Estimation Theory

Todd Humphreys

Fall 2017

Contents

Chapter 1

Overview

1.1 Course Glossary

Stochastic means random, or non-deterministic. The state of a stochastic process or system can only be described probabilistically. All physical systems are fundamentally stochastic even if we sometimes we treat them as deterministic for convenience.

Estimation is the process of inferring the value of a quantity from indirect and inaccurate observations. The state or parameter to be estimated may be defined over a continuum of values.

Detection is the process of deciding among possible discrete states of the world based on noisy (stochastic) measurements. It inovlves hypothesis testing against a set of finite possibilities and selecting one that "best" represents our estimates. Detection problems can be though of as a subset of estimation problems.

Control of a dynamical system is the art and science of regulating the system to achieve a desired purpose. In stochastic control, our knowledge of the system state is uncertain; at most we have an estimate of the system state.

There are many heuristic approaches to detection, estimation, and control, but in this course we offer formal approaches; i.e., we offer the "proper" way of addressing the problem in the sense that our solution optimizes some agreed-upon criterion.

Dirty secret: Purists are still waiting for their extra hard drives to arrive and for their algorithms to finish executing. Out of expediency, we all make some approximations and and appeal from time to time to heuristics.

1.2 Example Applications

1.2.1 Fantastically simple: Least squares estimation problem

We often work with equations of the form

$$\boldsymbol{z} = H\boldsymbol{x} + \boldsymbol{w} \tag{1.1}$$

$\boldsymbol{x}$ is a vector of deterministic but unknown parameters
$\boldsymbol{w}$ is a vector of Gaussian noise

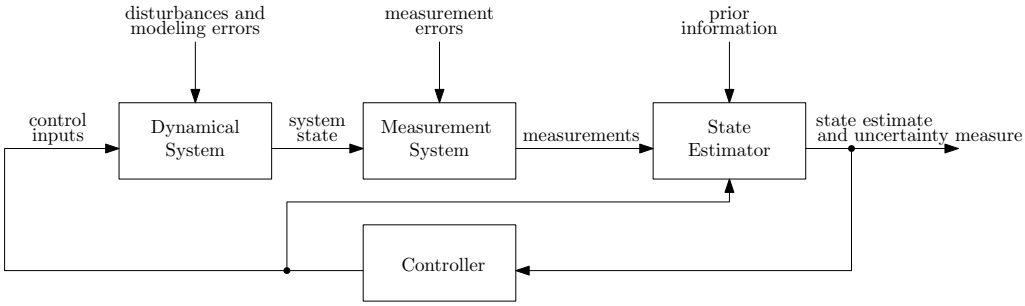


Figure 1.1: Elements of estimation and stochastic control.

H is a matrix that linearly relates $\boldsymbol{x}$ to
$\boldsymbol{z}$, the measurement vector.

This is the classical least-squares problem formulation. [Supplement: See the paper "From Gauss to Kalman" on the course website [1].]

From an information theory perspective, we say that $\boldsymbol{z}$ and $\boldsymbol{x}$ share mutual information: $\boldsymbol{z}$ tells you something about $\boldsymbol{x}$ while $\boldsymbol{w}$ acts to destroy this information coupling.

1.2.2 Fantastically complex: Estimation of GPS clocks and orbits

Consider the Global Positioning System: 32 satellites at medium earth orbit (MEO) (a = 26,561 km) across 6 orbital planes. The International GNSS Service (IGS) Tracking Network has 300+ ground sites (shown in Fig. 1.2) that track the GPS satellites.

Q: How do the sites know where they are?
A: They range off of several SVs whose positions are known and perform trilateration. ■
Q: How do the SVs know where they are?
A: They range off of the sites...[there must be some fun anectdote for this] ■

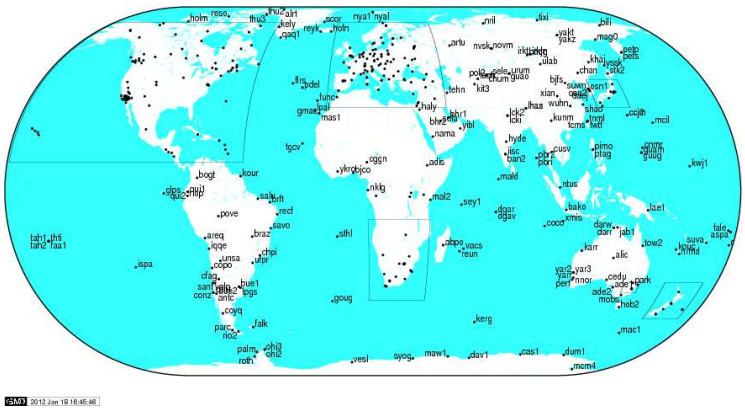


Figure 1.2: IGS ground station network (from `http://igscb.jpl.nasa.gov`).

When all the quantities of interest are compiled together, one has a staggeringly complex estimator:

- non-linear state dynamics and measurement equations
- non-uniform, time-varying gravitational potential

Figure 1.3: Representation of GPS satellite constellation at MEO.

- disturbance forces on SVs such as attitude-dependent solar heating
- crustal motion—alters ground site location
- ionosphere and neutral atmosphere effects
- earth orientation parameter variations (EOP)
- clock variations

Goal: estimate all site locations and velocities, all SV orbital parameters (position and velocity ephemerides), atmospheric delays, slowly-varying SV disturbances, EOP, clock errors. Wow!

1.3 General Solution Procedure for Estimation Problems

A. Develop a physics-based model that relates the unknown parameters or states to the measurements. Identify in the model the sources of uncertainty. Quantify the levels of uncertainty.

B. Do an analysis to determine whether the unknown quantities are *uniquely determinable* from the measurements (observability analysis).

C. If answer to B is yes, then apply estimation technique to estimate the unknown quantities. Also estimate the *uncertainty* of your estimate.

Importance of doubting our estimates, of *quantifying* the uncertainty in them:

> There are known knowns, These are things we know that we know. There are known unknowns. That is to say, there are things that we known we don't know. But there are also unknown unknowns. There are things we don't know we don't know. —*Donald Rumsfeld*, 2002

> It ain't what you don't know that gets you into trouble. It's what you know for sure that just ain't so. —*Mark Twain*

- known knowns: an estimate with a small error ellipse
- known unknowns: an estimate with a large error ellipse
- unknown unknowns: no estimate; no error ellipse
- known can't-knowns: observability analysis says not uniquely determinable
- unknown can't knowns: didn't do the observability analysis
- known but wrongs: an estimate with a small error ellipse, but estimate is wrong and you don't know it

Example. Missile Problem

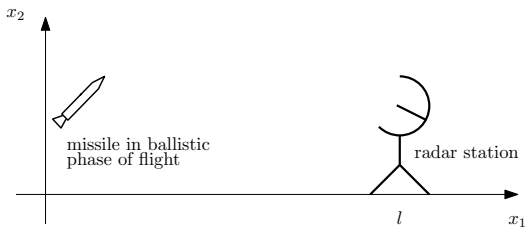


Figure 1.4: An illustration of the missile tracking problem.

Physics:

$$\ddot{x}_1 = 0$$
$$\ddot{x}_2 = -g$$

Known relationships:

$$y_{1k} = x_1(t_k) + n_{1k}$$
$$y_{2k} = x_2(t_k) + n_{2k}$$

Known noise characteristics:

$$\mathbb{E}[n_{1k}] = \mathbb{E}[n_{2k}] = 0$$
$$\mathbb{E}[n_{1k} n_{1j}] = \sigma_1^2 \delta_{kj}$$
$$\mathbb{E}[n_{2k} n_{2j}] = \sigma_2^2 \delta_{kj}$$

Unknown quantities (parameters): $x_1(0), x_2(0), v_1(0)$ and $v_2(0)$, where $v_1 = \dot{x}_1$ and $v_2 = \dot{x}_2$.

First solve the equations of motion to determine $x_1(t)$ and $x_2(t)$ (i.e., build the dynamics model):

$$x_1(t) = x_1(0) + v_1(0)t$$
$$x_2(t) = x_2(0) + v_2(0)t - \tfrac{1}{2}gt^2$$

Then relate the measurements to the unknown parameters (i.e., build the measurement model):

$$y_{1k} = x_1(0) + v_1(0)t_k + n_{1k}$$
$$y_{2k} = x_2(0) + v_2(0)t_k - \tfrac{1}{2}gt_k^2 + n_{2k}$$

Q: Are $x_1(0), x_2(0)$, etc. *observable* from $\{y_{11}, y_{21}, y_{12}, \ldots, y_{1n}, y_{2n}\}$?
A: Yes! (for $n \geq 2$) ■

Now develop an estimation algorithm and apply it:

$$\underbrace{\begin{bmatrix} y_{11} \\ y_{21} \\ \vdots \end{bmatrix}}_{\boldsymbol{z}} = \underbrace{\begin{bmatrix} 1 & t_1 & 0 & 0 \\ 0 & 0 & 1 & t_1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x_1(0) \\ v_1(0) \\ x_2(0) \\ v_2(0) \end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix} 0 \\ -\frac{1}{2}gt_1^2 \\ \vdots \end{bmatrix}}_{\boldsymbol{u}} + \underbrace{\begin{bmatrix} n_{11} \\ n_{21} \\ \vdots \end{bmatrix}}_{\boldsymbol{w}}$$

Assuming some estimation strategy, we get the following quantities: $\hat{x}_1(0), \hat{x}_2(0), \hat{v}_1(0), \hat{v}_2(0)$. Finally, predict the impact location noting that $x_2(t_{\text{impact}}) = 0$.

$$0 = \hat{x}_2(0) + \hat{v}_2(0)t_{\text{impact}} - \tfrac{1}{2}gt_{\text{impact}}^2$$
$$t_{\text{impact}} = \frac{\hat{v}_2(0) + \sqrt{\hat{v}_2^2(0) + 2g\hat{x}_2(0)}}{g}$$

Plug t_{impact} into (2.1) to obtain site estimate $\hat{x}_1(t_{\text{impact}})$. We can also estimate $\sigma_{t_{\text{impact}}}$ and $\sigma_{x_1(t_{\text{impact}})}$.

Recursion problem: How to incorporate new measurements as they arrive without starting from scratch?

Chapter 2

Linear Algebra Review

2.1 Vector and Matrix Operations

Vector A vector is an ordered list of quantities. In this course, $\boldsymbol{v}$ will always denote a column vector:

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \tag{2.1}$$

where v_j is the j^{th} scalar element and $\boldsymbol{v} \in \mathbb{R}^n$.

Matrix A matrix is a two-dimensional array of quantities.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{m1} & \cdots & \cdots & a_{mn} \end{bmatrix} \tag{2.2}$$

$$A = \begin{bmatrix} a_{ij} \end{bmatrix} \tag{2.3}$$

We say that A is an $m \times n$ matrix, or $A \in \mathbb{R}^{m \times n}$; a_{ij} is the $(i,j)^{th}$ element (row i, column j). (Beware: sometimes we will denote a_{ij} as A_{ij}.)

Addition and scalar multiplication Matrix addition is element-by-element, as is scalar multiplication:

$$C = \alpha A + \beta B \implies c_{ij} = \alpha a_{ij} + \beta b_{ij} \tag{2.4}$$

Identity The identity matrix is represented as

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{2.5}$$

$$\tag{2.6}$$

The size of the identity matrix can be denoted $I_{n \times n}$ if required.

Transpose Matrix and vector transpose are denoted by as A^T and $\boldsymbol{v}^T$. Note the following rule governing the transpose of a product:

$$\boxed{A = BC, \text{ then } A^T = C^T B^T} \tag{2.7}$$

Multiplication Matrix-vector and matrix-matrix multiplication are carried out as follows:

$$\underbrace{\boldsymbol{a}}_{n\times 1} = \underbrace{B}_{n\times m}\underbrace{\boldsymbol{c}}_{m\times 1} \implies a_i = \sum_{j=1}^{m} b_{ij}c_j, \quad i = 1, ..., n \tag{2.8}$$

$$\underbrace{A}_{m\times n} = \underbrace{B}_{m\times q}\underbrace{C}_{q\times n} \implies a_{ij} = \sum_{k=1}^{l} b_{ik}c_{kj}, \quad i = 1, ..., n, \quad j = 1, ..., m \tag{2.9}$$

Inner product The inner product between two $n \times 1$ vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is the scalar $\alpha = \boldsymbol{a}^T\boldsymbol{b}$. Alternate notation: $\alpha = \langle \boldsymbol{a}, \boldsymbol{b} \rangle$

Outer product If $\boldsymbol{b} \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{c} \in \mathbb{R}^{m\times 1}$, then the outer product of $\boldsymbol{b}$ and $\boldsymbol{c}$ is $A = \boldsymbol{b}\boldsymbol{c}^T \in \mathbb{R}^{n\times m}$, with $a_{ij} = b_i c_j$.

Rank of a matrix is the number of independent rows or columns. Independent rows (columns) cannot be expressed as a linear combination of the other rows (columns) in a matrix.

Q: What is the rank of an outer product?
A: The rank is 1: all columns and all rows are linearly dependent.

Trace of A is the sum of the diagonal elements: $\text{tr}(A) = \sum_{i=1}^{n} a_{ii}$. Note that if $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{m\times n}$, then

$$\boxed{\text{tr}(\underbrace{AB}_{n\times n}) = \text{tr}(\underbrace{BA}_{m\times m})} \tag{2.10}$$

Symmetric matrix A matrix A is symmetric if $A = A^T$. Note that symmetry implies A must be *square*; i.e., $A \in \mathbb{R}^{n\times n}$.

Quadratic form If $P = P^T$, then we can define

$$\boxed{\alpha = \boldsymbol{x}^T P \boldsymbol{x}} \tag{2.11}$$

a quadratic form in $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$. The scalar α is related to P and $\boldsymbol{x}$ by

$$\alpha = \sum_{j=1}^{n}\sum_{i=1}^{n} x_i P_{ij} x_j \tag{2.12}$$

Positive definiteness Given $P = P^T$, if $\boldsymbol{x}^T P \boldsymbol{x} > 0 \;\forall\; \boldsymbol{x} \neq \boldsymbol{0}$, then P is positive definite (notation: $P > 0$). If only $\boldsymbol{x}^T P \boldsymbol{x} \geq 0$, then P is positive semidefinite (notation: $P \geq 0$).

Q: Which is positive definite?: $\begin{bmatrix} 1 & -0.5 \\ -0.5 & 2 \end{bmatrix}$ or $\begin{bmatrix} 1 & 5 \\ 5 & 2 \end{bmatrix}$
A: The first one is, the second is not. You can see that the first matrix is almost diagonal, which means that its eigenvalues are near 1 and 2; the second matrix is not nearly diagonal, so it's harder to tell.

Q: Can one discern positive definiteness by inspection?
A: Not in general, but $P > 0 \Leftrightarrow$ all eigenvalues are positive (we will learn about eigenvalues soon). So if you can discern eigenvalues by inspection, you can discern positive definiteness too.

Weighted inner product The weighted inner product

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle_P = \boldsymbol{a}^T P \boldsymbol{b}$$

is a proper inner product if $P > 0$.

2.2 Norms

The l_1, l_2, and l_∞ norms of a vector $\boldsymbol{x}$ are defined as:

l_1: $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$, "Manhattan norm"

l_2: $\|\boldsymbol{x}\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2} = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}$, "Euclidean norm"

l_∞: $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$, "Chessboard norm"

The l_∞ norm is called the chessboard norm or chessboard distance because it is equal to the minimum number of moves a king must make to go from one square to another, if each square is of side length 1. For an entertaining (and informative) read on taxicab geometry (that uses the l_1- instead of the l_2-norm for distance), see `http://www.ams.org/samplings/feature-column/fcarc-taxi`.

In general, the p-norm can be defined $\forall\ p \geq 1$:

$$\|\boldsymbol{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p} \tag{2.13}$$

In this course, we will work exclusively with the l_2 norm because it makes optimization easier (since it has continuous derivatives), and because, as it turns out, l_2 has a deep connection to the maximum likelihood criterion for Gaussian random variables.

Hereafter, as a matter of notation, $\|\boldsymbol{x}\| \triangleq \|\boldsymbol{x}\|_2$.

P-weighted norm If $P > 0$, then $\|\boldsymbol{x}\|_P = \sqrt{\boldsymbol{x}^T P \boldsymbol{x}}$ is a valid norm called the P-weighted norm.

Induced matrix norm The concept of a vector norm can be extended to matrices. The induced matrix norm of A is defined as

$$\| \underbrace{A}_{n \times m} \| = \max_{\|\boldsymbol{x}\|=1} \|A\boldsymbol{x}\| \tag{2.14}$$

This norm is called *induced* because it is defined through the norm of the vector $A\boldsymbol{x}$.

Properties of norms All norms satisfy three properties:

1. $\|\boldsymbol{x}\| \geq 0$ and $\|\boldsymbol{x}\| = 0$ iff $\boldsymbol{x} = \boldsymbol{0}$
2. $\|\alpha \boldsymbol{x}\| = |\alpha| \|\boldsymbol{x}\|$
3. $\|\boldsymbol{a} + \boldsymbol{b}\| \leq \|\boldsymbol{a}\| + \|\boldsymbol{b}\|$ (the triangle inequality)

Schwartz inequality:

$$|\boldsymbol{x}^T \boldsymbol{y}| \leq \|\boldsymbol{x}\| \|\boldsymbol{y}\| \tag{2.15}$$

This is equivalent to saying $|\cos\theta| \leq 1$ because $|\boldsymbol{x}^T \boldsymbol{y}| = \|\boldsymbol{x}\| \|\boldsymbol{y}\| \cos\theta$.

2.3 Determinant and Inverse

Determinant The determinant of a square matrix A is denoted $|A|$ and is defined as

$$|A| = \sum_{i=1}^{n} a_{ij}|C_{ij}|(-1)^{i+j}, \quad j = 1, \ldots, n \tag{2.16}$$

where C_{ij} is A with the i^{th} row and j^{th} column removed. Note that this definition is recursive with the following the terminal case:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \tag{2.17}$$

Also, $\big|\big[a\big]\big| = a$ for a 1×1 matrix (Note that this is not the same as the absolute value; a may be negative).

Q: Where does the determinant come from? What does it tell us?

A: The determinant shows up when taking a matrix inverse: If $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, then $A^{-1} = \frac{1}{|A|}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$. The determinant tells us crucial information about a matrix (e.g., whether the matrix is invertible.)

Singularity and non-singularity

$A \in \mathbb{R}^{n \times n}$ is singular if $|A| = 0$.

$|A| = 0$ implies that A is rank deficient, or $\text{rank}(A) < n$.

If $|A| \neq 0$, then A is non-singular (all columns are linearly independent).

If $|A| = 0$, then $\exists\ \boldsymbol{x} \neq 0$ such that $A\boldsymbol{x} = \mathbf{0}$.

But if $|A| \neq 0$, then $A\boldsymbol{x} = \mathbf{0} \Rightarrow \boldsymbol{x} = \mathbf{0}$.

Matrix inversion If $A \in \mathbb{R}^{n \times n}$ and $|A| \neq 0$, $\exists\ A^{-1}$ such that $A^{-1}A = AA^{-1} = I$.

$$A^{-1} = \frac{1}{|A|}\begin{bmatrix} |C_{11}| & -|C_{21}| & \cdots & (-1)^{n+1}|C_{n1}| \\ -|C_{12}| & |C_{22}| & & \vdots \\ \vdots & & \ddots & \vdots \\ (-1)^{1+n}|C_{1n}| & \cdots & \cdots & |C_{nn}| \end{bmatrix} \tag{2.18}$$

If $A = BC$ and A, B, and C are non-singular, then

$$\boxed{A^{-1} = C^{-1}B^{-1}} \tag{2.19}$$

Solution of linear equations If $A \in \mathbb{R}^{n \times n}$ and $|A| \neq 0$, and $\boldsymbol{b} \in \mathbb{R}^{n \times 1}$, then we can solve $A\boldsymbol{x} = \boldsymbol{b}$ by $\boldsymbol{x} = A^{-1}\boldsymbol{b}$. Note that $\boldsymbol{x} \in \mathbb{R}^{n \times 1}$ is a unique solution.

Inversion of partitioned matrices Let $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$ with P_{11}, P_{22} square. If P_{11} is invertible and its so-called Schur complement $\Delta = P_{22} - P_{21}P_{11}^{-1}P_{12}$ is invertible, then:

$$P^{-1} = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \tag{2.20}$$

Where

$$\begin{aligned} V_{11} =& P_{11}^{-1} + P_{11}^{-1}P_{12}\Delta^{-1}P_{21}P_{11}^{-1} \\ V_{12} =& -P_{11}^{-1}P_{12}\Delta^{-1} \\ V_{21} =& -\Delta^{-1}P_{21}P_{11}^{-1} \\ V_{22} =& \Delta^{-1} \end{aligned}$$

Note that $\dim(P_{ij}) = \dim(V_{ij})$.
Beware: the alternate formulas in Bar-Shalom 2nd edition have errors in them.

Matrix inversion lemma

$$\boxed{(A+BCD)^{-1} = A^{-1} - A^{-1}B\left(DA^{-1}B + C^{-1}\right)^{-1}DA^{-1}} \tag{2.21}$$

Useful alternate forms:

If $D = B^T$, then

$$\boxed{\left(A+BCB^T\right)^{-1} = A^{-1} - A^{-1}B\left(B^TA^{-1}B + C^{-1}\right)^{-1}B^TA^{-1}} \tag{2.22}$$

If $A = P^{-1}$, $B = H^T$, $D = H$, $C = R^{-1}$, then

$$\boxed{\left(P^{-1} + H^TR^{-1}H\right)^{-1} = P - PH^T\left(HPH^T + R\right)^{-1}HP} \tag{2.23}$$

Orthonormal matrices If Q is $n \times n$ and $|Q| \neq 0$ and $Q^{-1} = Q^T$, then Q is orthonormal (often simply referred to as orthogonal). Thus $Q^TQ = QQ^T = I$. Also, all columns of Q are perpendicular to each other and have unit norm: $\boldsymbol{q}_i^T\boldsymbol{q}_j = \boldsymbol{\delta}_{ij}$.

Important property of an orthonormal matrix:

$$\boxed{\|Q\boldsymbol{x}\| = \|\boldsymbol{x}\|} \tag{2.24}$$

In other words, Q is isometric. Proof: $\|Q\boldsymbol{x}\|^2 = \boldsymbol{x}^TQ^TQ\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{x} = \|\boldsymbol{x}\|^2$

2.4 Matrix Factorization

Householder transformation (See Bierman pp. 59-63, Gill et al. pp. 37-38.) Let $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$. Then there is a matrix H such that

$$H\boldsymbol{x} = \begin{bmatrix} \alpha & 0 & \cdots & 0 \end{bmatrix}^T \tag{2.25}$$

for some α. The matrix H, called a Householder matrix, is both symmetric and orthogonal: $H^TH = I$, $H^T = H$, and $\|H\boldsymbol{x}\| = \|\boldsymbol{x}\| = |\alpha|$.

The Householder transformation can be used to compress the magnitude ("energy") of a vector into a single component, zeroing all other components.

H has the form

$$H = I - \frac{2\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}} \tag{2.26}$$

One can solve for the vector $\boldsymbol{v}$ that causes H to satisfy $H\boldsymbol{x} = \begin{bmatrix}\alpha & 0 & \cdots & 0\end{bmatrix}^T$ as follows:

$$\boldsymbol{v} = \boldsymbol{x} + \text{sign}(x_1)\|\boldsymbol{x}\| \begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix} \tag{2.27}$$

Also, $\alpha = -\text{sign}(x_1)\|\boldsymbol{x}\|$.

QR Factorization Given any $A \in \mathbb{R}^{m\times n}$, we can express A as a product of an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ and an upper triangular matrix $R \in \mathbb{R}^{m\times n}$:

$$QR = A \tag{2.28}$$

All elements of R below the main diagonal are zero. If A is square and invertible then so is R. What's more, because R is upper triangular, R^{-1} is easy to compute. If $m > n$ then R is not square, but it can be partitioned into a square upper-triangular matrix $\widetilde{R} \in \mathbb{R}^{n\times n}$ and a zero matrix:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots \\ 0 & r_{22} & r_{23} & \cdots \\ 0 & 0 & r_{33} & \cdots \\ 0 & 0 & 0 & \ddots \\ \vdots & \vdots & \vdots & \end{bmatrix} = \begin{bmatrix}\widetilde{R}\\0\end{bmatrix}$$

The orthogonal matrix Q can be formed as a product of Householder matrices:

$$Q = H_1 H_2 \ldots H_p, \quad p = \min(m, n) \tag{2.29}$$

You'll calculate some example Householder matrices in one of the problem sets.

In MATLAB:

```
[Q, R] = qr(A);
```

Cholesky Factorization Given an $n \times n$ matrix $P = P^T$ with $P > 0$, there is a unique upper triangular R such that $R^T R = P$. One can think of R as a matrix square root of P.

In MATLAB:

```
R = chol(P);
```

2.5 Eigenvalues and Eigenvectors

Given $A \in \mathbb{R}^{n \times n}$, $\exists$ scalars λ_i (possibly complex) and associated vectors $\boldsymbol{v}_i$ (whose elements are complex if $\lambda_i \in \mathbb{C}$) such that

$$\boxed{A\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i} \tag{2.30}$$

with $\|\boldsymbol{v}_i\| \neq 0$. This condition can also be written as $(A - \lambda_i I)\boldsymbol{v}_i = \boldsymbol{0}$. Thus, $\det(A - \lambda_i I) = 0$ gives a degree n polynomial in λ_i (called the characteristic equation) whose roots are λ_i. The scalars λ_i are called eigenvalues, and the vectors $\boldsymbol{v}_i$ are their corresponding eigenvectors.

Properties

- $(A = A^T) \Rightarrow$ the n eigenvalues of A are all real and the n eigenvectors are independent and orthogonal. The converse is not necessarily true.
- $(A > 0) \Leftrightarrow (\lambda_i > 0)$
- $(A \geq 0) \Leftrightarrow (\lambda_i \geq 0)$
- $|A| = \prod_{i=1}^{n} \lambda_i$
- $\mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i$

Interesting anectdote on eigenvalues

> [W]hen Werner Heisenberg discovered "matrix" mechanics in 1925, he didn't know what a matrix was (Max Born had to tell him), and neither Heisenberg nor Born knew what to make of the appearance of matrices in the context of the atom. (David Hilbert is reported to have told them to go look for a differential equation with the same eigenvalues, if that would make them happier. The did not follow Hilbert's well-meant advice and thereby may have missed discovering the Schrodinger wave equation.
>
> — *M.R. Schroeder* in "Number theory in science and communications"

Chapter 3

Probability Theory Review

> The true logic of this world is in the calculus of probabilities.
>
> —*Maxwell*

> Our most precise description of nature must be in terms of probabilities.
>
> —*Feynman*

3.1 Origins of Randomness

Q: One often finds in engineering textbooks a phrase such as "the disturbance $d(t)$ is a random process." But is this a proper characterization? Is $d(t)$ truly random? One way of defining random is "unpredictable," but this definition is somewhat subjective. For whom is it unpredictable?
A: Safest approach: Say that $d(t)$ can be *modeled* as a random process.

Q: Where does randomness—or apparent randomness—originate?
A: We can entertain two answers: some processes are *essentially* random; others are *apparently* random.

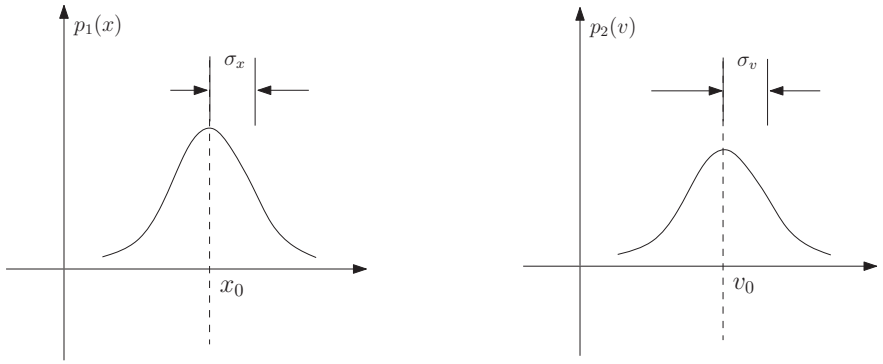
3.1.1 Essential Randomness at the Quantum Scale

Heisenberg uncentainty principle The accuracy with which we can determine the position and the velocity (or momentum) of a particle is limited by the relation (Figure 3.1)

$$\sigma_x \sigma_v \geq \frac{\hbar}{m} \tag{3.1}$$

where $\hbar$ is Planck's constant and m is the mass of the particle.

Some physicists have interpreted this as an inherent clumsiness in our measurement techniques: position measurement unavoidably gives a particle a random "kick," thereby affecting its velocity. But this is wrong. No refinement of our measurement techniques will allow us to violate the uncertainty principle. Fundamentally, the principle arises from the relationship between the wave function $\psi(t)$, which represents the particle's position as an amplitude distribution, and its Fourier transform $\tilde{\psi}$, which represents the particle's momentum. So for us the Heisenberg uncertainty principle can be reduced to the statement that what is precisely defined in one domain (e.g., a delta function $\delta(t)$ in the time domain), has a spectrally flat Fourier transform, and vice versa.



(a) Position of particle (b) Velocity of particle

Figure 3.1: Probability distributions for a particle's position and velocity.

(Interesting note: The evolution of $\psi(t)$ is entirely deterministic, it's governed by the Schrodinger wave equation.) Many physicists (though not all) have accepted the unsettling conclusion that the wave function $\psi(t)$ describes a particle's objective reality. The particle really has a simultaneous existence at multiple places.

Q: But does quantum indeterminism scale up to the classical level—to the level of dice and wind gusts and brain synapses? If not, what do we mean by "random" at the classical scale?
A: Probably not (still some debate here). The noise in the voltage across a resistor has to do with random movement of electrons; this might be *essentially* unpredictable. But by the time you get to cells and dice the wave functions have been magnified to the classical level and quantum superposition has been destroyed, effects have averaged out, and behavior is no longer indeterminate. ■

3.1.2 Effective Randomness at the Classical Scale

Consider the following analog to flipping a coin. Assume a classical-scale particle (e.g., a marble) is caught in a potential given by $V(x) = \frac{x^4}{4} - \frac{x^2}{2}$ (Fig. 3.2).

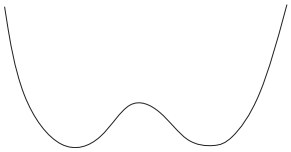


Figure 3.2: Potential function $V(x)$.

For motion only along x and a particle of mass $m = 1$, we have

$$\ddot{x} = -\frac{dV}{dx} = -x^3 + x \tag{3.2}$$

Add a bit of damping to get $\ddot{x} = -\beta\dot{x} - x^3 + x$. Then the particle dynamics are given by

$$\dot{x} = v \tag{3.3}$$

$$\ddot{x} = \dot{v} = -\beta\dot{x} - x^3 + x. \tag{3.4}$$

The phase portrait (Fig. 3.3) has three fixed points: two sinks and a saddle. The basin of attraction for the sink at $(x^*, v^*) = (-1, 0)$ winds snake-like around all the fixed points right next to the snake-like basin of attraction for the sink $(x^*, v^*) = (1, 0)$.

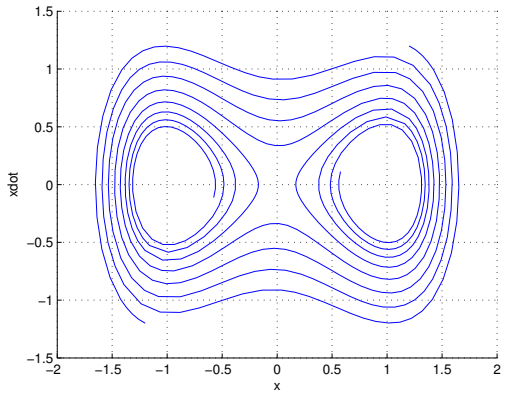


Figure 3.3: Phase portrait of particle dynamics.

For large energies $E(x, v) = \frac{1}{2}v^2 + V(x)$,the total density of the basin of attraction for some neighborhood ΔE approaches 50%. So we require increasing energy precision $\delta E = \frac{\Delta E}{E}$ to specify the sink that we want (e.g., heads).

For practical purposes, the outcome of the experiment, whether heads or tails, is random for high energies $E(x, v)$. *Deterministic does not necessarily mean practically predictable.*

Chaos theorists call it "transient chaos" when a system is sensitive to initial conditions but is ultimately periodic (has a simple final state). Strogatz: "You don't need strange attractors to generate effectively random behavior" ([2], p. 333).

3.2 Competing Definitions of Probability

Measure of belief: Make arguments for outcomes based, e.g., on the phase portrait in proceeding example. (No experiments required—perhaps imagined experiments—maybe not even that.)

Relative frequency:

$$P(A) = \lim_{N \to \infty} \frac{N_A}{N} \tag{3.5}$$

The relative frequency definition gives pure mathemeticians fits:

(a) In what sense doese the limit exist? We can't ever perform $N = \infty$ experiments!

(b) What about non-repeatable experiments?

3.3 Axiomatic Approach to Probability

experiment: a process with (effectively) random outcomes.
event: a set of possible outcomes of experiments.

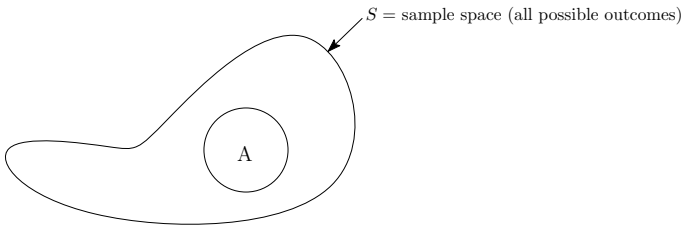


Figure 3.4: The event A occurs when an experiment has been done and the outcome is an element of A.

Probability of A: a number that satisfies the follows axioms:

1. $P(A) \geq 0$
2. $P(S) = 1$. (S is the sample space, the "sure event")
3. Let $A \cap B$ denote the set of outcomes in both A and B. If $A \cap B = \emptyset$, then $P(A \cup B) = P(A) + P(B)$

Corallaries:

1. $P(A) = 1 - P(\overline{A})$, where $\overline{A} \triangleq$ "not A"
2. $A \cup \overline{A} = S$
3. $A \cap \overline{A} = \emptyset$
4. $P(A \cup \overline{A}) = P(A) + P(\overline{A}) = 1$

Practical approach: Assign probabilities by whatever means (e.g., relative frequency, measure of belief) but make sure they satisfy the axioms.

Beware of pitfalls when assigning probabilities! [Show the magic trick with 1, 2, 3, 4.]

3.4 Scalar Random Variables

Probability Density Function (pdf) Let an event be defined as $\{x : \xi - d\xi < x \leq \xi\}$. Then the probability density function $p_x(\xi)$ can be defined as

$$\boxed{p_x(\xi) \triangleq \lim_{d\xi \to 0} \frac{P(\xi - d\xi < x \leq \xi)}{d\xi}} \tag{3.6}$$

Q: $p_x(\xi)$ is commonly written $p(x)$, but this shortcut notation makes purists wince. Why?
A: ■

Q: What is the value of $P(x = \xi)$?
A: ■

Probability of x being within an interval:

$$P(\eta < x \leq \xi) = \int_{\eta}^{\xi} p(x)dx \tag{3.7}$$

Cumulative density function (cdf)

$$P_x(\xi) \triangleq P(x \leq \xi) = \int_{-\infty}^{\xi} p(x)dx \tag{3.8}$$

Note: $P_x(\infty) = 1$ is the "sure event"; it places a normalization requirement on the pdf.

Gaussian pdf

$$\boxed{p_x(\xi) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left[-\frac{(\xi-\mu)^2}{2\sigma^2}\right]}} \text{ (deathbed identity)} \tag{3.9}$$

Notation: $x \sim \mathcal{N}(\mu, \sigma^2)$, $p_x(\xi) = \mathcal{N}(x; \mu, \sigma^2)$

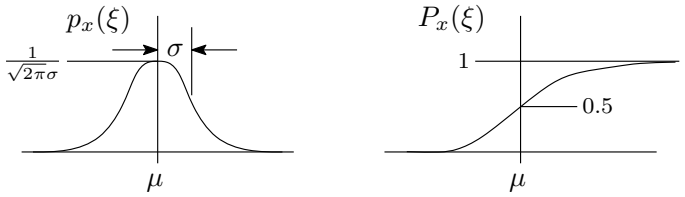


Figure 3.5: Gaussian pdf and cdf.

Discrete-valued random variables If x can only take on discrete values ξ_i, for $i = 1, \ldots, n$, then we define probability masses μ_i, $i = 1, 2, \ldots, n$ as follows:

$$\mu_i \triangleq P(x = \xi_i) \Rightarrow \sum_i \mu_i = 1 \tag{3.10}$$

and we can define the distribution of x as

$$p(x) = \sum_{i=1}^{n} \mu_i \delta(x - \xi_i) \tag{3.11}$$

where $\delta(\cdot)$ is the Dirac delta function.

Expected value The expected value, or the mean, of a scalar radom variable x is defined as

$$\mathbb{E}[x] \triangleq \int_{-\infty}^{\infty} xp(x)dx = \bar{x} \tag{3.12}$$

In general, for any function $g(x)$, whether linear or nonlinear,

$$\mathbb{E}[g(x)] = \int_{-\infty}^{\infty} g(x)p(x)dx \tag{3.13}$$

Variance The variance of a scalar random variable x is defined as

$$\sigma_x^2 \triangleq \mathbb{E}\left[(x - \bar{x})^2\right] = \int_{-\infty}^{\infty} (x - \bar{x})^2 p(x)dx \tag{3.14}$$

$$= \mathbb{E}\left[x^2\right] - \bar{x}^2 \tag{3.15}$$

The quantity $\sigma_x \geq 0$ is the standard deviation of x.

Shorthand: $x \sim [a, b]$ means $\mathbb{E}[x] = a, \sigma_x^2 = b$.

Note that if $x \sim \mathcal{N}(\mu, \sigma^2)$, then

$$\bar{x} = \int_{-\infty}^{\infty} x(\frac{1}{\sqrt{2\pi}\sigma})e^{-\frac{(x-\mu)^2}{2\sigma_x^2}}\,dx = \mu \tag{3.16}$$

Similarly, $\sigma_x^2 = \sigma^2$.

3.5 Jointly-Distributed Random Variables

Joint density of two random variables Events on a joint distribution are regions in 2-space; outcomes are points:

$$p_{x,y}(\xi, \eta) \triangleq \lim_{d\xi \to 0, d\eta \to 0} \frac{P\{(\xi - d\xi < x \leq \xi) \cap (\eta - d\eta < y \leq \eta)\}}{d\xi d\eta} \tag{3.17}$$

One creates a marginal density from a joint density by "integrating out" one of the dimensions:

$$p_x(\xi) = \int_{-\infty}^{\infty} p_{x,y}(\xi, \eta)d\eta. \tag{3.18}$$

This is often written as

$$p(x) = \int_{-\infty}^{\infty} p(x, y)dy. \tag{3.19}$$

Mean and variance of one of the random variables is defined in terms of marginal densities:

$$\begin{aligned} \mathbb{E}[x] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xp(x, y)dxdy \\ &= \int_{-\infty}^{\infty} x \underbrace{\int_{-\infty}^{\infty} p(x, y)dy}_{p(x)} dx \\ &= \int_{-\infty}^{\infty} xp(x)dx = \bar{x} \end{aligned} \tag{3.20}$$

A similar argument holds for $\mathbb{E}[y]$; variance definitions follow a simillar pattern.

Joint cumulative distribution of two random variables

$$P_{x,y}(\xi, \eta) \triangleq P(x \leq \xi, y \leq \eta) = \int_{-\infty}^{\xi}\int_{-\infty}^{\eta} p(x, y)\ dxdy \tag{3.21}$$

Covariance

$$\begin{aligned} \mathrm{Cov}(x, y) \triangleq& \mathbb{E}[(x - \bar{x})(y - \bar{y})] \\ =& \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x - \bar{x})(y - \bar{y})p(x, y)dxdy \\ =& \sigma_x\sigma_y\rho_{xy} \end{aligned} \tag{3.22}$$

where we define the correlation coefficient ρ_{xy} as

$$\rho_{xy} \triangleq \frac{\mathrm{Cov}(x,y)}{\sigma_x \sigma_y} \tag{3.23}$$

Properties of ρ_{xy}:

- $|\rho_{xy}| \leq 1$
- $(\rho_{xy} = 0) \Rightarrow x, y$ are uncorrelated
- $(\rho_{xy} = 1) \Rightarrow x, y$ are linearly dependent

Conditional probability Are events A and B related in the sense that knowledge of the occurrence of one alters the probability of the other? Then they share *mutual information*.

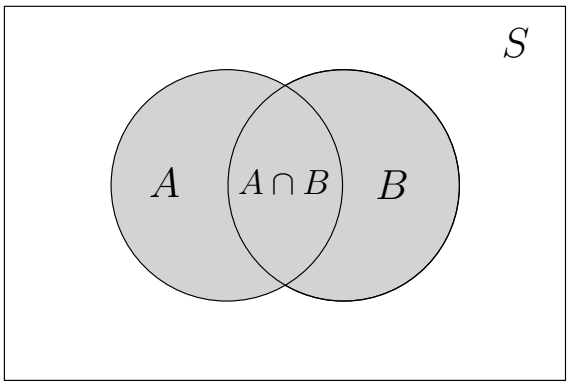


Figure 3.6: Say we know B occurred. Then the sample space S "shrinks" to B; the event A also occurred only if the outcome is in $A \cap B$.

To find the probability that event A occurred given that B occurred, we only need to divide the area of the overlapping region $A \cap B$ by the reduced sample space B. Symbolically,

$$\boxed{P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \cap A)}{P(B)}} \tag{3.24}$$

Similarly, for random variables we have

$$\boxed{p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y,x)}{p(y)}} \tag{3.25}$$

Independent events and independent random variables If knowledge of the occurrence of B does not alter the probability of A, then A and B are independent events. We say that events A and B share no mutual information. Symbolically, if

$$P(A) = P(A|B) \tag{3.26}$$

then A and B are independent. Note that

$$\left[P(A) = P(A|B) = \frac{P(A \cap B)}{P(B)}\right] \Leftrightarrow [P(A \cap B) = P(A)P(B)] \tag{3.27}$$

By definition, A and B are said to be independent events if

$$P(A \cap B) = P(A)P(B) \tag{3.28}$$

Similarly, random variables are independent if the joint probability density function $p(x, y)$ can be factored into the product of its marginal densities $p(x)$ and $p(y)$, i.e., if

$$p(x, y) = p(x)p(y) \tag{3.29}$$

Note that if x and y are independent then

$$\begin{aligned} \mathrm{Cov}(x, y) &= \int\int (x - \bar{x})(y - \bar{y})p(x, y)\, dxdy \\ &= \int (x - \bar{x})p(x)\, dx \int (y - \bar{y})p(y)\, dy = 0 \end{aligned} \tag{3.30}$$

$$\Rightarrow \rho_{xy} = \frac{\mathrm{Cov}(x, y)}{\sigma_x \sigma_y} = 0 \tag{3.31}$$

In other words, there is no coupling between x and y.

Q: Does the Venn diagram in Figure 3.7 imply independence between events A and B? If not, can we make a Venn diagram that does?

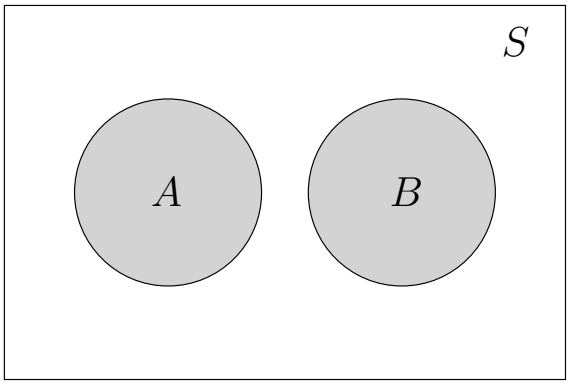


Figure 3.7: Does this Venn diagram imply independence between events A and B?

A: Far from implying such, the Venn diagram in Figure 3.7 is not even compatible with independence between A and B. From the diagram we may infer that $P(A) > 0$ and $P(B) > 0$ since A and B are nonempty sets of *possible* outcomes of the experiment. Also from the diagram we have that $P(A \cap B) = 0$ since $A \cap B = \varnothing$. But if A and B are independent then $P(A \cap B) = P(A)P(B) \neq 0$, a contradiction. In general, Venn diagrams, on their own, are an inadequate tool for describing independence. Such diagrams can only tell us whether A and B are disjoint ($A \cap B = \varnothing$), or, conversely, whether they share some possible outcomes ($A \cap B \neq \varnothing$). However, if we overlay a probability distribution on the sample space of a Venn diagram, then the combined distribution and diagram can imply independence (see Fig. 3.8).

Total probability theorem If $B_1, B_2 \ldots, B_n$ are mutually exclusive and exhaustive events; i.e., $\{B_i \cap B_j = \varnothing\}, \forall i \neq j$, and $\{B_1 \cup B_2 \cup \cdots \cup B_n\} = S$, so that the B_i form a *partitioning* of the sample space, then

$$\boxed{P(A) = \sum_{i=1}^{n} P(A, B_i) = \sum_{i=1}^{n} P(A|B_i)P(B_i)} \tag{3.32}$$

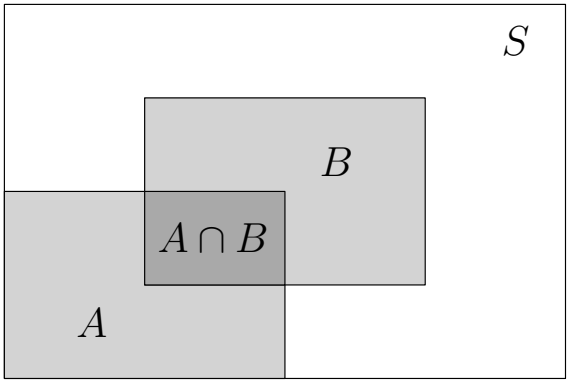


Figure 3.8: This Venn diagram, taken with an assumed uniform distribution of outcome probability across S, implies independence between events A and B. This is because, due to the uniform distribution, event probability is proportional to area, and, by inspection, one can see that the independence condition $\frac{P(A\cap B)}{P(B)} = P(A)$ is satisfied.

For random variables,

$$\boxed{p(x) = \int_{-\infty}^{+\infty} p(x,y)dy = \int_{-\infty}^{+\infty} p(x|y)p(y)dy} \tag{3.33}$$

Bayes's theorem Bayes's theorem allows us to *reverse the conditioning* in conditional probabilities. Consider n mutually exclusive and exhaustive events $B_1, B_2 \ldots, B_n$, and let A be any event with $P(A) > 0$. Then

$$\boxed{P(B_i|A) = \frac{P(B_i \cap A)}{P(A)} = \frac{P(A|B_i)P(B_i)}{\sum_{j=1}^{n} P(A|B_j)P(B_j)}} \tag{3.34}$$

For continuous random variables x and y, Bayes's theorem is expressed in terms of densities:

$$\boxed{p(x|y) = \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x)p(x)dx}} \tag{3.35}$$

This formula for the conditional density $p(x|y)$ is the "workhorse of estimation." Note that $p(x|y)$ is a valid probability density function and hence sums to 1:

$$\int_{-\infty}^{+\infty} p(x|y)\, dx = \int_{-\infty}^{+\infty} \frac{p(y|x)p(x)}{p(y)}\, dx = \int_{-\infty}^{+\infty} \frac{p(x,y)}{p(y)}\, dx = \frac{p(y)}{p(y)} = 1 \tag{3.36}$$

For mixed cases, Bayes's theorem is written

$$\boxed{P(B_i|x) = \frac{p(x|B_i)P(B_i)}{p(x)} = \frac{p(x|B_i)P(B_i)}{\sum_{j=1}^{n} p(x|B_j)P(B_j)}} \tag{3.37}$$

or

$$\boxed{p(x|B_i) = \frac{P(B_i|x)p(x)}{P(B_i)}} \tag{3.38}$$

Bayes's theorem can also accommodate multiple conditioning random variables or events; for example,

$$\boxed{P(B_i|x,y) = \frac{p(x|B_i,y)P(B_i|y)}{p(x|y)} = \frac{p(x|B_i,y)P(B_i|y)}{\sum_{j=1}^{n} p(x|B_j,y)P(B_j|y)}} \tag{3.39}$$

Bayesian jargon

- $p(x)$ is an *a priori* distribution, sometimes called a prior for short
- $p(x|y)$ is an *a posteriori* distribution, sometimes called a posterior for short

Q: What if we have no prior distribution, no preconceptions, no stereotypes, no prejudice, no idea at all about the probable value of x? Or what if we object, on principle, to the use of a prior because it smacks of tainting an objective, empirical problem with a subjective judgement?
A: You can run from the Bayesian framework, but you'll only end up a *breathless Bayesian*. Just wrap your ignorance of x, or your high-minded empiricism, in a *diffuse* prior:

$$p(x) = \begin{cases} \epsilon & |x| \leq \frac{1}{2\epsilon} \\ 0 & \text{otherwise} \end{cases} \tag{3.40}$$

where $\epsilon > 0$. It can be shown that for a wide $1/\epsilon$, ϵ divides out of the formula $\frac{p(y|x)p(x)}{p(y)}$. ■

Conditional expectation Conditional expectation of a random variable x given another random variable y is defined as

$$\mathbb{E}[x|y] = \int_{-\infty}^{+\infty} x\, p(x|y)\, dx \tag{3.41}$$

In general, $\mathbb{E}[x|y] \neq \mathbb{E}[x]$, but

$$\begin{aligned} \underbrace{\mathbb{E}\left[\underbrace{\mathbb{E}[x|y]}_{\text{over } x|y}\right]}_{\text{over } y} &= \int_{-\infty}^{+\infty} \mathbb{E}[x|y]\; p(y)\; dy \\ &= \int_{-\infty}^{+\infty} \left[\int_{-\infty}^{+\infty} xp(x|y)dx\right] p(y)dy \\ &= \int_{-\infty}^{+\infty} x \left[\int_{-\infty}^{+\infty} p(x,y)dy\right] dx \\ &= \mathbb{E}[x] = \bar{x} \end{aligned}$$

Q: How should we interpret $\mathbb{E}[\mathbb{E}[x|y]]$?
A: Think of the inner expectation as $g(y) = \mathbb{E}[x|y]$, some ordinary function of y, and then apply (3.13) to take the expectation of $g(y)$. Note that because y is a random variable, we think of $g(y) = \mathbb{E}[x|y]$ as a random variable. ■

Vector-valued random variables Let $\boldsymbol{x}$ denote an n-dimensional vector random variable:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

The joint pdf describing the statistics of the random vector $\boldsymbol{x}$ is defined as

$$p_{\boldsymbol{x}}(\boldsymbol{\xi}) = p_{\boldsymbol{x}}(\xi_1, \xi_2, \ldots, \xi_n)$$

$$\triangleq \lim_{\substack{d\xi_1 \to 0, \\ d\xi_2 \to 0, \\ \cdots \\ d\xi_n \to 0}} \frac{P\{(\xi_1 - d\xi_1 < x_1 \leq \xi_1) \cap (\xi_2 - d\xi_2 < x_2 \leq \xi_2) \cap \cdots \cap (\xi_n - d\xi_n < x_n \leq \xi_n)\}}{d\xi_1 d\xi_2 \ldots d\xi_n} \tag{3.42}$$

The expected value of the random vector $\boldsymbol{x}$ is denoted $\mathbb{E}[\boldsymbol{x}] = \bar{\boldsymbol{x}} \in \mathbb{R}^n$ and is defined as

$$\mathbb{E}[\boldsymbol{x}] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \boldsymbol{x} p(\boldsymbol{x})\, dx_1 dx_2 \ldots dx_n \tag{3.43}$$

The covariance matrix of $\boldsymbol{x}$ is denoted $P_{xx} \in \mathbb{R}^{n \times n}$ and is defined as

$$P_{xx} = \mathrm{Cov}(\boldsymbol{x}) = \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}}\right] = \int (\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}} p(\boldsymbol{x}) d\boldsymbol{x} \tag{3.44}$$

where the (i,j)th element of P_{xx} is given by

$$P_{xxij} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} (x_i - \bar{x}_i)(x_j - \bar{x}_j) p(x_1, x_2, \ldots, x_n)\, dx_1 dx_2 \ldots dx_n$$

The covariance matrix P_{xx} is symmetric and positive semidefinite; i.e., $P_{xx} = P_{xx}^{\mathrm{T}} \geq 0$. If there is no linear dependence between any of the random elements of $\boldsymbol{x}$, then $P_{xx} > 0$.

Gaussian random vectors Given an $n \times n$ matrix $P = P^{\mathrm{T}} \geq 0$ and $\boldsymbol{\mu} \in \mathbb{R}^n$, we define the Gaussian distribution of $\boldsymbol{x}$ as

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |P|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} P^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right] \tag{3.45}$$

Properties:

- $\mathbb{E}[\boldsymbol{x}] = \boldsymbol{\mu}$
- $\mathbb{E}\left[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}}\right] = P$

Complex-valued random vectors Let $\boldsymbol{z} \in \mathbb{Z}^n$ be a complex-valued random vector with real and imaginary vector components $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$ so that

$$\boldsymbol{z} = \boldsymbol{x} + j\boldsymbol{y}$$

where $j = \sqrt{-1}$. The probability distribution of $\boldsymbol{z}$ is equivalent to that of the stacked vector $\boldsymbol{v} = [\boldsymbol{x}^{\mathrm{T}}, \boldsymbol{y}^{\mathrm{T}}]^{\mathrm{T}}$. For example, suppose the elements of $\boldsymbol{x}$ and $\boldsymbol{y}$ are Gaussian distributed and

mutually independent so that

$$\begin{aligned}
&\mathbb{E}[x_i] = \mu_{xi}, \quad \mathbb{E}[y_i] = \mu_{yi} \\
&\mathbb{E}[(x_i - \mu_{xi})(x_k - \mu_{xk})] = \sigma^2 \delta_{ik} \\
&\mathbb{E}[(y_i - \mu_{yi})(y_k - \mu_{yk})] = \sigma^2 \delta_{ik} \\
&\mathbb{E}[(x_i - \mu_{xi})(y_k - \mu_{yk})] = 0 \quad \forall i, k
\end{aligned}$$

Then the distribution of $\boldsymbol{z}$ can be written

$$p(\boldsymbol{x}) = \left(\frac{1}{2\pi\sigma^2}\right)^n \exp\left[-\frac{1}{2\sigma^2}\sum_{i=0}^{n-1}(x_i - \mu_{xi})^2 + (y_i - \mu_{yi})^2\right] \tag{3.46}$$

Joint and conditional Gaussian random variables *Note: Take some time to appreciate this section, as forms the basis of our approach to optimal estimation.*

Given two Gaussian random vectors $\boldsymbol{x}$ and $\boldsymbol{z}$, we can define a new Gaussian random vector $\boldsymbol{y}$ by stacking vectors $\boldsymbol{x}$ and $\boldsymbol{z}$:

$$\boldsymbol{y} = \begin{bmatrix}\boldsymbol{x} \\ \boldsymbol{z}\end{bmatrix}, \quad \dim(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = (n_x, n_y, n_z)$$

$$\mathbb{E}[\boldsymbol{y}] = \bar{\boldsymbol{y}} = \begin{bmatrix}\bar{\boldsymbol{x}} \\ \bar{\boldsymbol{z}}\end{bmatrix}, \quad P_{yy} = \mathrm{Cov}(\boldsymbol{y}) = \begin{bmatrix}P_{xx} & P_{xz} \\ P_{zx} & P_{zz}\end{bmatrix}$$

$$p(\boldsymbol{x}|\boldsymbol{z}) = \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z})} = \frac{p(\boldsymbol{y})}{p(\boldsymbol{z})} = \frac{(2\pi)^{\frac{n_z}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{y} - \bar{\boldsymbol{y}})^{\mathrm{T}} P_{yy}^{-1}(\boldsymbol{y} - \bar{\boldsymbol{y}})\right] |P_{zz}|^{\frac{1}{2}}}{(2\pi)^{\frac{n_x + n_z}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{z} - \bar{\boldsymbol{z}})^{\mathrm{T}} P_{zz}^{-1}(\boldsymbol{z} - \bar{\boldsymbol{z}})\right] |P_{yy}|^{\frac{1}{2}}} \tag{3.47}$$

After much algebra (including the block inverse formula), we can show that

$$\begin{aligned}
p(\boldsymbol{x}|\boldsymbol{z}) \quad = \quad & \frac{1}{(2\pi)^{\frac{n_x}{2}} |P_{xx} - P_{xz}P_{zz}^{-1}P_{xz}^{\mathrm{T}}|^{\frac{1}{2}}} \times \\
& \exp\left\{-\frac{1}{2}\left[\boldsymbol{x} - \bar{\boldsymbol{x}} - P_{xz}P_{zz}^{-1}(\boldsymbol{z} - \bar{\boldsymbol{z}})\right]^{\mathrm{T}} \left[P_{xx} - P_{xz}P_{zz}^{-1}P_{xz}^{\mathrm{T}}\right]^{-1} \left[\boldsymbol{x} - \bar{\boldsymbol{x}} - P_{xz}P_{zz}^{-1}(\boldsymbol{z} - \bar{\boldsymbol{z}})\right]\right\}
\end{aligned}$$

- **Mean**:

$$\mathbb{E}[\boldsymbol{x}|\boldsymbol{z}] = \bar{\boldsymbol{x}} + P_{xz}P_{zz}^{-1}(\boldsymbol{z} - \bar{\boldsymbol{z}}) \tag{3.48}$$

  This applies a correction term to $\bar{\boldsymbol{x}}$ based on the value of $\boldsymbol{z}$.

- **Covariance**:

$$\mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}|\boldsymbol{z}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}|\boldsymbol{z}])^{\mathrm{T}}|\boldsymbol{z}\right] = P_{xx} - P_{xz}P_{zz}^{-1}P_{xz}^{\mathrm{T}} \tag{3.49}$$

  Note that so long as $\boldsymbol{x}$ and $\boldsymbol{z}$ are correlated (i.e., share mutual information), and so long as $\boldsymbol{z}$ is not infinitely uncertain, then $P_{xz}P_{zz}^{-1}P_{xz}^{\mathrm{T}} > 0$. Thus, the conditioning on $\boldsymbol{z}$ reduces the covariance P_{xx}, which implies that $\mathbb{E}[\boldsymbol{x}|\boldsymbol{z}]$ is a better estimate of $\boldsymbol{x}$ than $\mathbb{E}[\boldsymbol{x}]$.

Expected value of a quadratic form Given $A = A^{\mathrm{T}}$ and a random vector $\boldsymbol{x}$ with $\mathbb{E}[\boldsymbol{x}] = \mathbf{0}$, then

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x}\right] &= \mathbb{E}\left[\mathrm{tr}(\boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x})\right] \\
&= \mathbb{E}\left[\mathrm{tr}(A \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}})\right] \quad (\text{since } \boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x} = \mathrm{tr}(\boldsymbol{x}^{\mathrm{T}} A \boldsymbol{x})) \\
&= \mathrm{tr}(A\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\right]) \quad (\text{since tr}(.) \text{ and } \mathbb{E}[.] \text{ are linear operators}) \\
&= \mathrm{tr}(A\mathrm{Cov}(\boldsymbol{x})) \quad (\text{since } \mathbb{E}[\boldsymbol{x}] = \mathbf{0} \text{ and } \mathrm{Cov}(\boldsymbol{x}) = \mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\right]) \\
&= \mathrm{tr}(AP_{xx}) \qquad (3.50)
\end{aligned}$$

3.6 The Chi-square Distribution

We will find that Chi-square-distributed random variables arise in estimator consistency analysis.

Suppose $\{x_1, x_2, ..., x_n\}$ is a set of independent Gaussian random variables each with mean 0 and variance 1. These are *independent, identically-distributed* (iid) random variables. Then

$$q = \sum_{i=1}^{n} x_i^2 \qquad (3.51)$$

is a Chi-square random variable of degree n, denoted $q \sim \chi_n^2$.

Mean

$$\mathbb{E}[q] = \sum_{i=1}^{n} \mathbb{E}\left[x_i^2\right] = n$$

Variance

$$\begin{aligned}
\mathrm{Var}(q) = \mathbb{E}\left[(q-n)^2\right] =& \mathbb{E}\left[\left(\sum_{i=1}^{n} x_i^2 - n\right)^2\right] \\
=& \mathbb{E}\left[\left(\sum_{i=1}^{n} (x_i^2 - 1)\right)^2\right] \\
=& \mathbb{E}\left[\sum_{i=1}^{n} (x_i^2 - 1)^2\right] \quad (\text{cross} - \text{terms cancel}) \\
=& \sum_{i=1}^{n} \mathbb{E}\left[(x_i^2 - 1)^2\right] \\
=& \sum_{i=1}^{n} \left[\mathbb{E}\left[x_i^4\right] - 2\mathbb{E}\left[x_i^2\right] + 1\right] \\
=& \sum_{i=1}^{n} (3 - 2 + 1) = 2n
\end{aligned}$$

Thus, if $q \sim \chi_n^2$ then q has a mean of n and a variance of $2n$, or $q \sim [n, 2n]$.

Distribution

$$p(q) = \begin{cases} \frac{1}{2^{\left(\frac{n}{2}\right)}\Gamma\left(\frac{n}{2}\right)} q^{\left(\frac{n-2}{2}\right)} \exp\left(\frac{-q}{2}\right) & q \geq 0 \\ 0 & q < 0 \end{cases} \qquad (3.52)$$

where the Gamma function satisfies $\Gamma(1/2) = \sqrt{\pi}$, $\Gamma(1) = 1$, and $\Gamma(m+1) = m\Gamma(m) = m!$ for m an integer.

Sum of Chi-square random variables If $q_1 \sim \chi^2_{n_1}$ and $q_2 \sim \chi^2_{n_2}$, then the sum $q_3 = q_1 + q_2$ is distributed as

$$q_3 \sim \chi^2_{n_1+n_2}$$

3.7 Random Sequences

A random sequence is a time-ordered sequence of random variables; e.g., $\{\ldots, \boldsymbol{x}(-2), \boldsymbol{x}(-1), \boldsymbol{x}(0), \ldots, \boldsymbol{x}(3)\}$. We define

$$X^k \triangleq \{\ldots, \boldsymbol{x}(k-1), \boldsymbol{x}(k)\} \tag{3.53}$$

as the set of all elements (data) in the sequence through index k.

Mean of a random sequence

$$\bar{\boldsymbol{x}}(k) \triangleq \mathbb{E}\left[\boldsymbol{x}(k)\right] \triangleq \int \boldsymbol{\xi} p_{\boldsymbol{x}(k)}(\boldsymbol{\xi}) d\boldsymbol{\xi} \tag{3.54}$$

Note that in general the mean is a function of the time index k.

Autocorrelation function

$$R(k,j) \triangleq \mathbb{E}\left[\boldsymbol{x}(k)\boldsymbol{x}^T(j)\right] \triangleq \int\int \boldsymbol{\xi}\boldsymbol{\eta} p_{\boldsymbol{x}(k),\boldsymbol{x}(j)}(\boldsymbol{\xi},\boldsymbol{\eta}) d\boldsymbol{\xi} d\boldsymbol{\eta} \tag{3.55}$$

Note that $R(k,j) = R^T(j,k)$.

Autocovariance

$$\begin{aligned} &\mathbb{E}\left[\left[\boldsymbol{x}(k) - \bar{\boldsymbol{x}}(k)\right]\left[\boldsymbol{x}(j) - \bar{\boldsymbol{x}}(j)\right]^T\right] \\ &= \mathbb{E}\left[\boldsymbol{x}(k)\boldsymbol{x}^T(j)\right] - \mathbb{E}\left[\boldsymbol{x}(k)\bar{\boldsymbol{x}}^T(j)\right] - \mathbb{E}\left[\bar{\boldsymbol{x}}(k)\boldsymbol{x}^T(j)\right] + \mathbb{E}\left[\bar{\boldsymbol{x}}(k)\bar{\boldsymbol{x}}^T(j)\right] \\ &= R(k,j) - \bar{\boldsymbol{x}}(k)\bar{\boldsymbol{x}}^T(j) \end{aligned} \tag{3.56}$$

Stationarity If $\bar{\boldsymbol{x}}(k) = \bar{\boldsymbol{x}}$ and if $R(k,j) = R(k-j,0)$, then $\boldsymbol{x}(k)$ is wide-sense stationary (WSS). For a WSS process we often use the shorthand notation $R(k-j)$ to denote $R(k-j,0)$, since the autocorrelation function only depends on the difference $k-j$. Note that for a WSS process $R(l) = R^T(-l)$. Also note that for a white noise process to be stationary we require $Q(k) = Q$, a constant.

Ergodicity A stationary random sequence is ergodic if expected values (i.e., ensemble averages) can be computed by time averaging; i.e., if

$$\bar{\boldsymbol{x}} = \int \boldsymbol{\xi} p_{\boldsymbol{x}(k)}(\boldsymbol{\xi}) d\boldsymbol{\xi} = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} \boldsymbol{x}(k) \tag{3.57}$$

Markov sequence A random sequence is Markov if

$$p[\boldsymbol{x}(k)|X^j] = p[\boldsymbol{x}(k)|\boldsymbol{x}(j)], \quad k \geq j \tag{3.58}$$

For Markov sequences, all important information about the past for purposes of (imperfectly) predicting the future is summarized in the most recent data sample.

White noise sequence A white noise sequence $\boldsymbol{v}(k)$ satisfies

$$\mathbb{E}\left[\boldsymbol{v}(k)\right] = 0 \quad \forall k \tag{3.59}$$

$$\mathrm{Cov}\left[\boldsymbol{v}(k), \boldsymbol{v}(j)\right] \triangleq \mathbb{E}\left[\boldsymbol{v}(k)\boldsymbol{v}^T(j)\right] = Q(k)\delta_{kj} \tag{3.60}$$

where $Q(k) > 0$ is the white noise intensity. Thus, for a white noise sequence the future is uncorrelated with the past.

Random walk Suppose $\boldsymbol{v}(k)$ is a white noise sequence with $Q(k) = Q$ so that $\mathbb{E}\left[\boldsymbol{v}(k)\boldsymbol{v}^T(j)\right] = Q\delta_{kj}$. Now consider the sequence $\boldsymbol{x}(k)$ defined by

$$\begin{aligned} \boldsymbol{x}(0) &= 0 \\ \boldsymbol{x}(k+1) &= \boldsymbol{x}(k) + \boldsymbol{v}(k), \quad k = 0, 1, \dots \end{aligned}$$

Q: Is $\boldsymbol{x}(k)$ a Markov sequence?
A: Yes. To see this, write $\boldsymbol{x}(k)$ as

$$\boldsymbol{x}(k) = \sum_{j=0}^{k-1} \boldsymbol{v}(j) \tag{3.61}$$

Now suppose we wanted to predict $\boldsymbol{x}(k+1)$ given $\{\boldsymbol{x}(k), \boldsymbol{v}(k-1), \boldsymbol{v}(k-2), \dots\}$. Can you see that the data $\{\boldsymbol{v}(k-1), \boldsymbol{v}(k-2), \dots\}$ are superfluous? Only $\boldsymbol{x}(k)$ is needed—it summarizes all past $\boldsymbol{v}(k)$ up to $\boldsymbol{v}(k-1)$. ■

The mean and autocorrelation of the random walk sequence are given by

$$\mathbb{E}\left[\boldsymbol{x}(k)\right] = \mathbb{E}\left[\sum_{j=0}^{k-1} \boldsymbol{v}(j)\right] = 0 \tag{3.62}$$

$$R(k,j) \triangleq \mathbb{E}\left[\boldsymbol{x}(k)\boldsymbol{x}^T(j)\right] = \sum_{l=0}^{\min(k,j)-1} \mathbb{E}\left[\boldsymbol{v}(l)\boldsymbol{v}^T(l)\right] = \min(k,j) \cdot Q \tag{3.63}$$

Note that since $R(k,j)$ depends on k, the random walk sequence is not stationary. To further appreciate this fact, consider a scalar random walk sequence $x(k)$. The variance at time index k is given by

$$\sigma(k) \triangleq \mathbb{E}\left[x^2(k)\right] = \sqrt{kQ} \tag{3.64}$$

Thus, a random walk sequence has zero mean but infinite variance. This is why it is sometimes called the "gambler's ruin": in an iterated game of chance against a fair house you're bound to break even eventually, but you'll need an infinite bank account to guarantee as much.

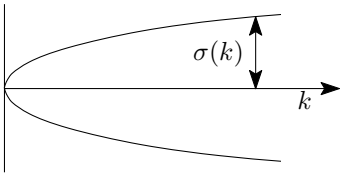


Figure 3.9: The standard deviation envelope of a scalar random walk sequence widens with index k.

Gauss-Markov sequence Let $\boldsymbol{v}(k)$ be a white noise sequence whose elements are Gaussian distributed and let $\boldsymbol{x}(0)$ be a Gaussian random variable. Then the sequence $\boldsymbol{x}(k)$ defined by

$$\boldsymbol{x}(k+1) = F(k)\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{3.65}$$

is a Gauss-Markov sequence. It is Gaussian because passing a Gaussian process through a linear system with a Gaussian initial state yields a Gaussian process. It is Markov because

$$p[\boldsymbol{x}(k+1)|\boldsymbol{x}(k), \boldsymbol{v}(k-1), \boldsymbol{v}(k-2), \dots] = p[\boldsymbol{x}(k+1)|\boldsymbol{x}(k)] \tag{3.66}$$

3.8 Law of large numbers

Consider a stationary scalar random sequence $\{x_1, x_2, \dots\}$ with mean $\mathbb{E}[x_i] = \bar{x}$ and covariance $\mathbb{E}[(x_i - \bar{x})(x_j - \bar{x})] = \sigma^2\rho(i-j)$, where $\rho(0) = 1$. If the sequence is asymptotically uncorrelated so that

$$\lim_{|i-j|\to\infty} \rho(i-j) = 0 \tag{3.67}$$

and if $\rho(i-j)$ approaches 0 *fast enough* with widening $|i-j|$; i.e., if

$$|\rho(i-j)| < e^{-\alpha|i-j|} \tag{3.68}$$

for some $\alpha > 0$, then the sample average

$$y_n = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.69}$$

converges to the expectation value in a mean square sense as $n \to \infty$; i.e.,

$$\mathbb{E}\left[(y_n - \bar{x})^2\right] \to 0 \quad \text{as} \quad n \to \infty \tag{3.70}$$

3.9 Central limit theorem

Consider a sequence of independent random variables $\{x_1, x_2, \dots\}$ whose elements are not necessarily identically distributed. So long as no one element dominates (i.e., so long as there is no σ_j such that $\sigma_j \ll \sigma_i \quad \forall i \neq j$), then the distribution of the normalized sum

$$z_n = \frac{1}{\sqrt{n}}\sum_{i=1}^{n} x_i \tag{3.71}$$

will tend toward a Gaussian distribution as $n \to \infty$. This is why the Gaussian distribution is so ubiquitous: it is a good model for random effects that are actually composites of smaller random contributions.

Intuition about the central limit theorem Let $z = x + y$ where x and y are independent. The cdf of z is given by (see Fig. 3.10)

$$\begin{aligned} P_z(\xi) = P(z < \xi) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\xi-x'} p_{x,y}(x', y')dy'dx' \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\xi-x'} p_x(x')p_y(y')dy'dx' \quad (x, y \quad \text{independent}) \end{aligned}$$

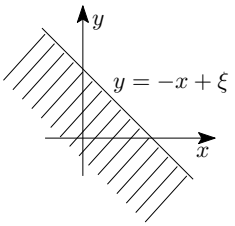


Figure 3.10: The line $y = -x + \xi$ marks the border of the region $z < \xi$.

and the pdf of z is

$$p_z(\xi) = \frac{d}{d\xi} P_z(\xi) = \int_{-\infty}^{\infty} p_x(x') p_y(\xi - x') dx' \\ = p(x) * p(y) \tag{3.72}$$

In other words, the pdf of the sum z is the convolution of the pdfs of the independent summands x and y. This is the key insight behind the central limit theorem. Why? Because repeated convolution tends to produce a Gaussian distribution, as illustrated in Fig. 3.11.

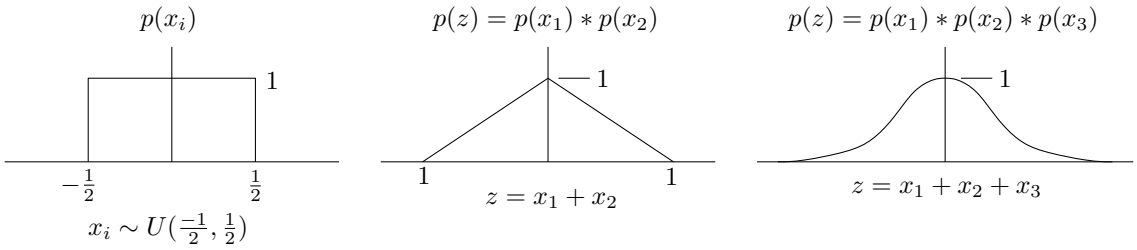


Figure 3.11: The probability distribution $p(z)$ begins to look Gaussian with just three independent uniformly-distributed components.

Chapter 4

Detection Theory Basics

4.1 Hypothesis Testing (HT)

HT provides a foundation for estimator consistency checking, multiple model estimation, signal detection (e.g., GPS signal acquisition), anomaly detection, etc.

Consider a parameter (or vector of parameters) θ. Assume that θ can take on only discrete values from the set $\{\theta_0, \theta_1, ..., \theta_{M-1}\}$. Let H_i denote the hypothesis that the true value of θ is θ_i:

$$\left.\begin{array}{r} H_0 : \theta = \theta_0 \\ H_1 : \theta = \theta_1 \\ \vdots \\ H_{M-1} : \theta = \theta_{M-1} \end{array}\right\} \text{ M-ary HT}$$

$$\left.\begin{array}{ll} H_0 : \theta = \theta_0 & \text{(null hypothesis)} \\ H_1 : \theta = \theta_1 & \text{(alternative hypothesis)} \end{array}\right\} \text{ binary HT}$$

Our focus for now will be on binary HT, for which we define the following probabilities:

$$\begin{array}{lll} P_F = & P[\text{ accept } H_1 | H_0 \text{ true }] & \text{(false alarm)} \\ P_D = & P[\text{ accept } H_1 | H_1 \text{ true }] & \text{(detection)} \\ P_M = & P[\text{ accept } H_0 | H_1 \text{ true }] = 1 - P_D & \text{(missed detection)} \end{array}$$

A decision between H_0 and H_1 is based on a set of random variables (observations) $\boldsymbol{z}$, where the *likelihood function* $p(\boldsymbol{z}|\theta_0)$ and $p(\boldsymbol{z}|\theta_1)$ are known. These likelihood functions behave like conditional distributions although, strictly speaking, they are not conditional distributions because, for now, θ is only considered an unknown parameter, not a random variable.

4.2 Neyman-Pearson Lemma (NP)

NP tells us how to choose optimally between H_0 and H_1.

Q: What do you suppose is a reasonable definition of optimality in this context?
A: Given a fixed maximum P_F, maximize P_D. ■

Let $\nu > 0$ be a threshold. NP states that the optimal test is of the form

$$\begin{cases} \text{choose } H_0 \text{ if} & p(\boldsymbol{z}|\theta_1) < \nu\, p(\boldsymbol{z}|\theta_0) \\ \text{choose } H_1 \text{ if} & p(\boldsymbol{z}|\theta_1) > \nu\, p(\boldsymbol{z}|\theta_0) \\ \text{choose randomly if} & p(\boldsymbol{z}|\theta_1) = \nu\, p(\boldsymbol{z}|\theta_0) \end{cases}$$

NP also states that such a test exists and that ν is unique for every $0 \leq P_F \leq 1$.

We can write the test more compactly by comparing the *likelihood ratio* $\Lambda(\boldsymbol{z})$ to ν:

$$\Lambda(\boldsymbol{z}) \triangleq \frac{p(\boldsymbol{z}|\theta_1)}{p(\boldsymbol{z}|\theta_0)} = \underbrace{\frac{p(\boldsymbol{z}|H_1)}{p(\boldsymbol{z}|H_0)}}_{\substack{\text{alternative} \\ \text{notation}}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu$$

Apply NP as follows:

1. Assign H_0 and H_1. Typically, one assigns H_0 to the hypothesis whose P_F one wants to strictly limit.

2. Select P_F. By convention, statisticians often set $P_F = 0.05$ or 0.01. $P_F = 0.05$ means that we are willing to accept a 5% chance that we reject H_0 ("incoming missile!") when true ("no missile").

3. Simplify the threshold test if possible. For example, it's often cleaner to consider

$$\log \Lambda(\boldsymbol{z}) \underset{H_0}{\overset{H_1}{\gtrless}} \log \nu$$

and since log is a monotonic function, the inequalities remain valid. We can denote successive simplified tests with primes:

$$\Lambda(\boldsymbol{z}) \underset{H_0}{\overset{H_1}{\gtrless}} \nu \quad \xrightarrow[\text{simplify}]{} \quad \Lambda'(\boldsymbol{z}) \underset{H_0}{\overset{H_1}{\gtrless}} \nu' \quad \xrightarrow[\text{simplify}]{} \quad \Lambda''(\boldsymbol{z}) \underset{H_0}{\overset{H_1}{\gtrless}} \nu'' \quad \xrightarrow[\cdots]{} \quad \Lambda^*(\boldsymbol{z}) \underset{H_0}{\overset{H_1}{\gtrless}} \nu^*$$

We denote by $\Lambda^*(\boldsymbol{z})$ the simplest reduction of the likelihood ratio that we could find, and by ν^* its corresponding threshold. We often refer to $\Lambda^*(\boldsymbol{z})$ as the *test statistic* or *detection statistic*. It is the function of the observations that ultimately gets compared to the threshold in the HT. It is also a *sufficient statistic*: for purposes of the HT, knowing $\Lambda^*(\boldsymbol{z})$ is just as good as knowing $\Lambda(\boldsymbol{z})$. Note that, because $\Lambda^*(\boldsymbol{z})$ is a function of the random variable $\boldsymbol{z}$, it is itself a random variable.

Aside
Employing a stricter notation for the likelihood functions, we could write

$$\Lambda_{\boldsymbol{z}}(\boldsymbol{\xi}) \triangleq \frac{p_{\boldsymbol{z}}(\boldsymbol{\xi}|\theta_1)}{p_{\boldsymbol{z}}(\boldsymbol{\xi}|\theta_0)}$$

In this notation, $\Lambda_{\boldsymbol{z}}(\boldsymbol{\xi})$ (or its reduction $\Lambda'_{\boldsymbol{z}}(\boldsymbol{\xi})$) is a function of $\boldsymbol{\xi}$, which isn't a random variable but rather a *test value* for the random variable $\boldsymbol{z}$. In the context of an HT, $\boldsymbol{\xi}$ is the *actual value* of the observation (measurement) that was made: it is a realization of the random variable $\boldsymbol{z}$. Thus, $\Lambda_{\boldsymbol{z}}(\boldsymbol{\xi})$ cannot be considered a random variable, and it wouldn't make sense to talk of finding its distribution. However, when we wish to evaluate the performance of an HT, we allow $\boldsymbol{\xi}$ to be replaced by $\boldsymbol{z}$ so that we can treat

$\Lambda_{\boldsymbol{z}}$ as a random variable and thereby measure the chances of $\Lambda_{\boldsymbol{z}}$ being above or below the given threshold. In keeping with strict notation, we could write

$$\Lambda_{\boldsymbol{z}}(\boldsymbol{z}) \triangleq \frac{p_{\boldsymbol{z}}(\boldsymbol{z}|\theta_1)}{p_{\boldsymbol{z}}(\boldsymbol{z}|\theta_0)}$$

when we mean to treat $\Lambda_{\boldsymbol{z}}$ as a random variable for this purpose. But at this point we might as well use the more casual notation $p(\boldsymbol{z}|\theta)$ for the likelihood function, and $\Lambda(\boldsymbol{z})$ for the likelihood ratio, since $\Lambda_{\boldsymbol{z}}(\boldsymbol{z})$ elides the distinction between $\boldsymbol{\xi}$ and $\boldsymbol{z}$ anyhow. (See also [3], pp. 467-468.)

4. Determine (or approximate) the distribution of $\Lambda^*(\boldsymbol{z})$ under H_0 and, if possible, under H_1.
5. Based on the distribution of $\Lambda^*(\boldsymbol{z})$ under H_0, calculate the value of ν^* that yields the chosen P_F.

Example. Assume the received signal in a radar detection problem can be modeled as

$$z = \theta + w, \qquad w \sim \mathcal{N}(0, \sigma^2)$$

Suppose our binary hypotheses are

$$H_0 : \theta = 0$$
$$H_1 : \theta = \theta_1 > 0$$

so that

$$p(z|\theta_0) = \mathcal{N}(z; 0, \sigma^2)$$
$$p(z|\theta_1) = \mathcal{N}(z; \theta_1, \sigma^2)$$

We form the log-likelihood ratio as

$$\Lambda'(z) = \log \Lambda(z) = \log \left[\frac{p(z|\theta_1)}{p(z|\theta_0)}\right] = -\left[\frac{(z-\theta_1)^2 - z^2}{2\sigma^2}\right] = \frac{2z\theta_1 - \theta_1^2}{2\sigma^2}$$

At this point the HT amounts to

$$\Lambda'(z) \underset{H_0}{\overset{H_1}{\gtrless}} \nu', \qquad \nu' \triangleq \log(\nu).$$

We can further simplify as

$$\Lambda^*(z) = z \underset{H_0}{\overset{H_1}{\gtrless}} \nu^*, \qquad \nu^* \triangleq \frac{2\sigma^2\nu' + \theta_1^2}{2\theta_1}$$

We choose ν^* to satisfy

$$P_F = P[z > \nu^* \,|\, \theta_0] = \int_{\nu^*}^{\infty} p(z|\theta_0)\, dz$$

For our choice of ν^* we obtain

$$P_D = P[z > \nu^* \,|\, \theta_1] = \int_{\nu^*}^{\infty} p(z|\theta_1)\, dz$$

The MATLAB syntax for determining ν^* and P_D is

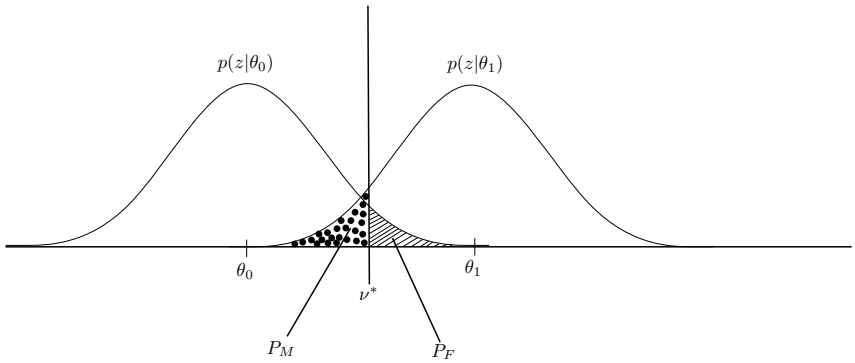


Figure 4.1: Binary hypothesis tests probability distributions. The area of the shaded regions corresponds to P_F and P_M.

```
nu_star = norminv(1-PF, theta_0, sigma);
PD = 1 - normcdf(nu_star, theta_1, sigma);
```

Remarks.

1. The structure of the test (i.e., the form of the detection statistic $\Lambda^*(\boldsymbol{z})$) depends on $p(\boldsymbol{z}|\theta_0)$ and $p(\boldsymbol{z}|\theta_1)$.
2. The choice of threshold ν^* depends only on P_F and $p(\Lambda^*(\boldsymbol{z})|\theta_0)$.
3. P_F applies to *each* hypothesis test; that is, to each time a measurement $\boldsymbol{z}$ is made and $\Lambda^*(\boldsymbol{z})$ is compared against a threshold. The combined P_F after repeated measurements may be large.
4. NP does not take into account any *a priori* judgement on the likelihood of H_0 and H_1.

Q: What rationale can we give for our choice of P_F? The NP formulation doesn't give any guidance here.
A: Remarks 3 and 4 point us to a Bayesian framework where we consider the expected cost, otherwise known as *risk*. ■

$$P_j = \text{prior probability of } H_j$$
$$C_{jk} = \text{cost of deciding } H_j \text{ when } H_k \text{ is true}$$

$$\mathcal{R} = C_{00}\,P_0\,P[\text{decide } H_0|H_0 \text{ is true}] + C_{10}\,P_0\,\underbrace{P[\text{decide } H_1|H_0 \text{ is true}]}_{P_F}$$
$$+ C_{11}\,P_1\,\underbrace{P[\text{decide } H_1|H_1 \text{ is true}]}_{P_D} + C_{01}\,P_1\,\underbrace{P[\text{decide } H_0|H_1 \text{ is true}]}_{P_M}$$

Goal: minimize $\mathcal{R}$.

As it turns out, choosing the decision regions to minimize $\mathcal{R}$ yields a likelihood ratio test, just as with NP:

$$\frac{p(\boldsymbol{z}|H_1)}{p(\boldsymbol{z}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \nu$$

But in the Bayesian case, the value of the threshold ν is set not to satisfy a predesignated P_F, but rather to minimize $\mathcal{R}$. The minimizing value of ν is

$$\nu = \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})}$$

Q: If the Bayesian approach automatically chooses the "right" value of ν, then why take the NP approach with its seemingly arbitrary value of P_F?
A: It's often not possible, or not practical, to determine C_{jk} and P_j. NP approach simplifies the problem by leaving P_F as a tuning parameter. ■

4.3 Detector Performance

A detector designed according to the NP formulation is guaranteed to have maximum P_D for a given P_F. Even so, P_D might not be acceptable. It is often useful to plot P_D as a function of P_F. The curve is called the *receiver operating characteristic* (ROC).

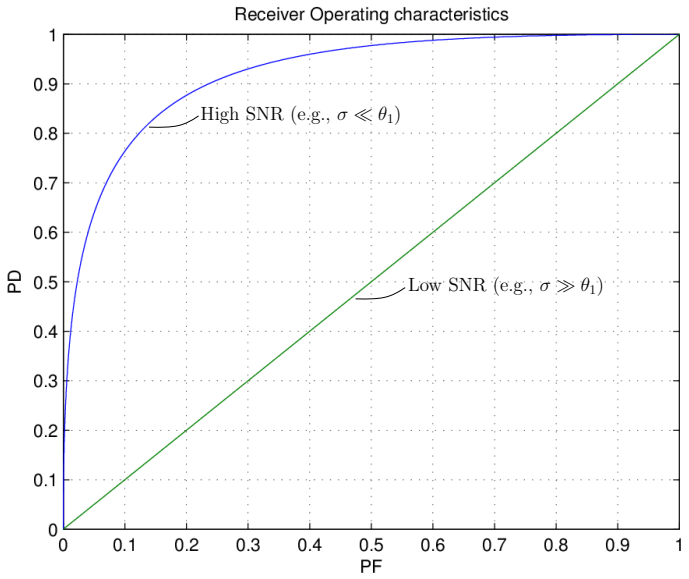


Figure 4.2: Example ROC curves.

4.4 Composite Hypothesis Testing

In a simple hypothesis test (HT), the possible values of the parameter θ are discrete: $\theta \in \{\theta_0, \theta_1, \ldots, \theta_{M-1}\}$. In a composite HT, θ can take on a *range* of values (e.g., $\theta \in [0, 2\pi]$). In many cases of practical interest, a composite hypothesis test again boils down to a likelihood ratio test.

Example.

$$\boldsymbol{z} \triangleq [z_1, z_2]^T \qquad \begin{matrix} z_1 = \theta_1 \cos\theta_2 + w_1 \\ z_1 = \theta_1 \sin\theta_2 + w_2 \end{matrix} \qquad w_1, w_2 \sim \mathcal{N}(0, \sigma^2)$$

$$\boldsymbol{\theta} \triangleq [\theta_1, \theta_2]^T \qquad \begin{matrix} \theta_1 \in \{0, A\} \\ \theta_2 \in [0, 2\pi] \end{matrix} \qquad \begin{matrix} A > 0 \text{ is a positive constant} \\ \theta_2 \sim \mathcal{U}(0, 2\pi) \text{ is a random variable} \\ w_1, w_2, \theta_2 \text{ are mutually independent} \end{matrix}$$

Consider a binary HT that decomposes the parameter space $\mathcal{X} = \{0, A\} \times [0, 2\pi]$ on which $\boldsymbol{\theta}$ lives into two disjoint regions:

$$H_0 : \boldsymbol{\theta} \in \mathcal{X}_0 \triangleq \{\boldsymbol{\theta} \in \mathcal{X} | \theta_1 = 0\}$$
$$H_1 : \boldsymbol{\theta} \in \mathcal{X}_1 \triangleq \{\boldsymbol{\theta} \in \mathcal{X} | \theta_1 = A\}$$

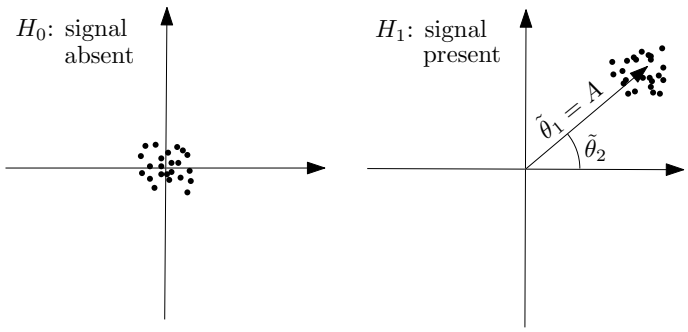


Figure 4.3: Composite HT problem.

The HT boils down to

$$\Lambda(\boldsymbol{z}) = \frac{p(\boldsymbol{z}|\boldsymbol{\theta} \in \mathcal{X}_1)}{p(\boldsymbol{z}|\boldsymbol{\theta} \in \mathcal{X}_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \nu$$

It is easy to show that

$$p(\boldsymbol{z}|\boldsymbol{\theta}) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{z} - \boldsymbol{\mu})^T(\boldsymbol{z} - \boldsymbol{\mu})\right]$$

where

$$\boldsymbol{\mu} = \boldsymbol{\mu}(\theta_1, \theta_2) = \begin{bmatrix} \theta_1 \cos\theta_2 \\ \theta_1 \sin\theta_2 \end{bmatrix}$$

Finding $p(\boldsymbol{z}|\boldsymbol{\theta} \in \mathcal{X}_0)$ is also straighforward, since $\theta_1 = 0$ in this case:

$$p(\boldsymbol{z}|\boldsymbol{\theta} \in \mathcal{X}_0) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{1}{2\sigma^2}\boldsymbol{z}^T\boldsymbol{z}\right]$$

Q: Finding $p(\boldsymbol{z}|\boldsymbol{\theta} \in \mathcal{X}_1)$ is not so easy. Can you see how?
A: Recall the total probability theorem for pdfs:

$$p(x) = \int p(x|y)p(y)dy$$

Similarly, to get $p(\boldsymbol{z}|\theta_1 = A)$ we'll need to average over θ_2:

$$p(\boldsymbol{z}|\boldsymbol{\theta} \in \mathcal{X}_1) = \int_0^{2\pi} p(\boldsymbol{z}|\theta_1 = A, \theta_2 = \xi) \underbrace{p_2(\xi)}_{\mathcal{U}[\xi;0,2\pi]} \mathrm{d}\xi$$
$$= \frac{1}{4\pi^2\sigma^2} \int_0^{2\pi} \exp\left\{-\frac{1}{2\sigma^2}[z - \boldsymbol{\mu}(A,\xi)]^T[z - \boldsymbol{\mu}(A,\xi)]\right\} \mathrm{d}\xi$$

■

Now we form the likelihood ratio

$$\Lambda(z) = \frac{p(\boldsymbol{z}|\boldsymbol{\theta} \in \mathcal{X}_1)}{p(\boldsymbol{z}|\boldsymbol{\theta} \in \mathcal{X}_0)} = \frac{\exp(-\frac{A^2}{2\sigma^2})}{2\pi} \int_0^{2\pi} \exp\left[\frac{A}{\sigma^2}(z_1 \cos\xi + z_2 \sin\xi)\right] d\xi$$

and simplify further by going to the polar coordinates:

$$r \triangleq \sqrt{z_1^2 + z_2^2} \quad \phi \triangleq \arctan\left(\frac{z_2}{z_1}\right)$$
$$z_1 = r\cos\phi \qquad z_2 = r\sin\phi$$

Plug in and simplify:

$$\Lambda(\boldsymbol{z}) = \exp(-\frac{A^2}{2\sigma^2}) \cdot \underbrace{\frac{1}{2\pi} \int_0^{2\pi} \exp\left[\frac{Ar}{\sigma^2}\cos(\xi - \phi)\right] d\xi}_{=I_0\left(\frac{Ar}{\sigma^2}\right)}$$

Here, $I_0\left(\frac{Ar}{\sigma^2}\right)$ is a 0th-order modified Bessel function of the 1st kind. Thus, we can simplify the HT as

$$\Lambda'(\boldsymbol{z}) = I_0\left(\frac{Ar}{\sigma^2}\right) \underset{H_0}{\overset{H_1}{\gtrless}} \nu'$$

But since $I_0(x)$ is monotonically increasing in x, the test is equivalent to

$$\Lambda^*(\boldsymbol{z}) \triangleq r \underset{H_0}{\overset{H_1}{\gtrless}} \nu^*$$

Thus, the optimal test compares $r = \sqrt{z_1^2 + z_2^2}$ against a threshold. One can show that

$$r \sim \begin{cases} \text{Rayleigh distribution} & \text{under } H_0 \\ \text{Rice distribution} & \text{under } H_1 \end{cases}$$

Q: Why do we so often find ourselves performing a correlation as part of a detection test?
A: A correlation operation generates an optimal detection test statistic for the broadly-applicable general Gaussian HT. ■

4.4.1 General Gaussian Problem

Consider a HT for a vector-valued Gaussian random variable $\boldsymbol{z}$:

$$H_0 : \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu_0}, P_0)$$
$$H_1 : \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu_1}, P_1)$$

Note that the elements of $\boldsymbol{z}$ may be correlated (P_j may not be diagonal). We can show the HT reduces to [HW]

$$\Lambda'(\boldsymbol{z}) = (\boldsymbol{z}-\boldsymbol{\mu}_0)^T P_0^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_0) - (\boldsymbol{z}-\boldsymbol{\mu}_1)^T P_1^{-1}(\boldsymbol{z}-\boldsymbol{\mu}_1) \underset{H_0}{\overset{H_1}{\gtrless}} \nu'$$

In the general case, finding an analytical expression for the distribution of $\Lambda'(z)$ is *not* easy. But some special cases are tractable.

Example. Suppose $P_0 = P_1 = P$, then the test reduces to:

$$\Lambda''(\boldsymbol{z}) = \Delta\boldsymbol{\mu}^T P^{-1}\boldsymbol{z} \underset{H_0}{\overset{H_1}{\gtrless}} \nu''$$

where $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. Let $\boldsymbol{z}_a = R_a^{-T}\boldsymbol{z}$ and $\Delta\boldsymbol{\mu}_a = R_a^{-T}\Delta\boldsymbol{\mu}$, where $R_a^T R_a = P$ is the Cholesky factorization of P. Then the HT becomes

$$\Lambda^*(\boldsymbol{z}_a) = \Delta\boldsymbol{\mu}_a^T\boldsymbol{z}_a \underset{H_0}{\overset{H_1}{\gtrless}} \nu^*$$

which implies a correlation-and-accumulation structure:

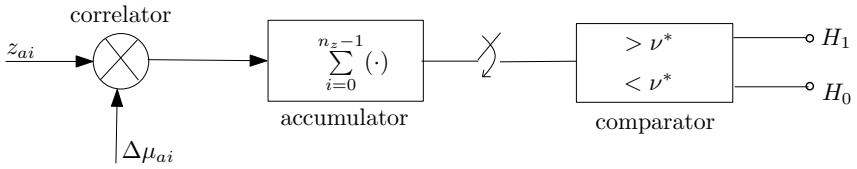


Figure 4.4: Correlation and accumulation structure implied by the equal-covariance case of the general Gaussian HT.

Chapter 5

Estimation Basics

5.1 Basic Problem Statement

Given a data set $Z^k \triangleq \{\boldsymbol{z}(1), \boldsymbol{z}(2), \ldots, \boldsymbol{z}(k)\}$, which could be, e.g., actual sensor measurements, we wish to estimate some unknown parameter $\boldsymbol{x}$. Our estimate will be a function of the data set and possibly time, i.e.,

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(k, Z^k) = \boldsymbol{f}(k, Z^k)$$

Note: We will usually abbreviate $\hat{\boldsymbol{x}}(k, Z^k)$ as $\hat{\boldsymbol{x}}(k)$.

Let's consider a few reasonable ways to define the function $\boldsymbol{f}(k, Z^k)$, i.e., ways to design an estimator:

5.2 Maximum Likelihood (ML) Estimation

Suppose we have no *a priori* information about the parameter $\boldsymbol{x}$ (or we otherwise don't wish to prejudice the estimator with our subjective biases). Then we simply regard $\boldsymbol{x}$ as an unknown constant $n \times 1$ vector. Define the *likelihood function*

$$\Lambda_{Z^k}(\boldsymbol{x}) \triangleq p_{Z^k|\boldsymbol{x}}(\tilde{Z}^k|\boldsymbol{\xi}) = \underbrace{p(Z^k|\boldsymbol{x})}_{\text{casual notation}}$$

We often write the likelihood function as $\Lambda(\boldsymbol{x})$ for short. The maximum likelihood (ML) estimate is the value of $\boldsymbol{x}$ that maximizes $\Lambda(\boldsymbol{x})$:

$$\hat{\boldsymbol{x}}_{\text{ML}} = \arg\max_{\boldsymbol{x}} \Lambda(\boldsymbol{x})$$

We can interpret the ML estimate as follows: After having performed the experiment, we "look back" and view it from the point of view of someone about to perform it. We pick the $\boldsymbol{x}$ that makes the data look most likely.

How to maximize $\Lambda(\boldsymbol{x})$? The usual strategy: satisfy the first order necessary condition (FONC) of optimality by setting its derivative with respect to $\boldsymbol{x}$ to zero:

$$\frac{\partial \Lambda(\boldsymbol{x})}{\partial \boldsymbol{x}} = \left(\frac{\partial \Lambda}{\partial x_1}, \frac{\partial \Lambda}{\partial x_2}, \ldots, \frac{\partial \Lambda}{\partial x_n} \right) = \boldsymbol{0}$$

This condition implicitly defines $\hat{\boldsymbol{x}}_{\text{ML}}$ as the solution of n equations in n unknowns.

5.3 Maximum A Posteriori (MAP) Estimation

Assume that we have some prior information about the parameter $\boldsymbol{x}$: we know it is a sample from $p(\boldsymbol{x})$, called the prior pdf. Assume we also know the conditional pdf $p(Z^k|\boldsymbol{x})$. Then the posterior distribution can be written by Bayes's rule as

$$p(\boldsymbol{x}|Z^k) = \frac{p(Z^k|\boldsymbol{x})p(\boldsymbol{x})}{\int p(Z^k|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}} \tag{5.1}$$

The MAP estimate $\hat{\boldsymbol{x}}_{\text{MAP}}$ is the value of $\boldsymbol{x}$ that maximizes the posterior distribution:

$$\hat{\boldsymbol{x}}_{\text{MAP}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|Z^k)$$

Remarks.

1. The denominator in (5.1) is constant with respect to the maximization parameter $\boldsymbol{x}$, so it doesn't affect the maximization.
2. ML and MAP are equivalent if $p(\boldsymbol{x})$ is diffuse:

$$p(\boldsymbol{x}) = \lim_{\epsilon \to 0} \begin{cases} \epsilon, & |x| \leq \frac{1}{2\epsilon} \\ 0, & |x| > \frac{1}{2\epsilon} \end{cases}$$

Example. ML Estimator for a Linear First-Order Dynamical System

Consider a linear dynamical system characterized by the differential equation

$$\dot{y} = -ay.$$

The solution to this system is

$$y(t) = y(0)e^{-at}$$

where $y(0)$ is the initial condition. Suppose our objective is to estimate $x \triangleq y(0)$ given the discrete measurements

$$z(j) = y(j\Delta t) + w(j), \quad j = 1, 2, \ldots, k$$

where Δt is the sampling interval, $w(j) \sim p_w(\xi) = \mathcal{N}(\xi; 0, \sigma^2)$, and $\mathbb{E}\left[w(i)w(j)\right] = \sigma^2\delta_{ij}$.

Setting $\gamma_j \triangleq e^{-aj\Delta t}$, we can rewrite the measurement equation as

$$z(j) = \gamma_j x + w(j), \quad j = 1, 2, \ldots, k$$

By independence of the measurements, the likelihood function can be written as

$$\Lambda(\boldsymbol{x}) \triangleq p(Z^k|\boldsymbol{x}) = p(z(1)|x)p(z(2)|x)\cdots p(z(k)|x)$$

But the pdf $p(z(j)|x)$ is nothing more than $p_w(\xi)$ shifted by $z(j) - \gamma_j x$:

$$p(z(j)|x) = \mathcal{N}(z(j); \gamma_j x, \sigma^2) = p_w(z(j) - \gamma_j x)$$

Therefore, we can write

$$p(Z^k|\boldsymbol{x}) = p_w(z(1) - \gamma_1 x)p_w(z(2) - \gamma_2 x)\cdots p_w(z(k) - \gamma_k x)$$

and the likelihood function becomes

$$\Lambda(\boldsymbol{x}) = \frac{1}{(2\pi)^{k/2}\sigma^k} \exp\left\{\frac{-1}{2\sigma^2}\sum_{j=1}^{k}\left[z(j) - \gamma_j x\right]^2\right\}$$

To maximize $\Lambda(\boldsymbol{x})$, we apply the FONC:

$$\Lambda(\boldsymbol{x}) \cdot \left[\frac{-1}{\sigma^2} \sum_{j=1}^{k} [z(j) - \gamma_j x] (-\gamma_j) \right] \triangleq 0$$

Recognizing that $\Lambda(\boldsymbol{x})$ cannot equal zero, we see that the sum must equal 0. Solving for x yields the ML estimate:

$$\hat{x}_{\text{ML}} = \frac{\sum_{j=1}^{k} z(j)\gamma_j}{\sum_{j=1}^{k} \gamma_j^2}$$

Example. MAP Estimator for a Linear First-Order Dynamical System Suppose we know x was sampled from a Gaussian distribution with known mean and variance: $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$. The posterior pdf in this case is given by

$$p(x|Z^k) = \frac{C}{(2\pi)^{\frac{k+1}{2}} \sigma^k \sigma_x} \exp \left[\frac{-1}{2\sigma^2} \sum_{j=1}^{k} [z(j) - \gamma_j x]^2 - \frac{1}{2\sigma_x^2} (x - \mu_x)^2 \right]$$

Here,

$$C = \left[\int_{-\infty}^{\infty} p(Z^k|x) p(x) dx \right]^{-1}$$

is a constant that does not depend on x and so is immaterial for purposes of maximization. To find $\hat{x}_{\text{MAP}}$, we apply the FONC to maximize $p(\boldsymbol{x}|Z^k)$:

$$\begin{aligned} 0 &= p(\boldsymbol{x}|Z^k) \left[\frac{1}{\sigma^2} \sum_{j=1}^{k} [z(j) - \gamma_j x] \gamma_j - \frac{1}{\sigma_x^2} (x - \mu_x) \right] \\ \Rightarrow \hat{x}_{\text{MAP}} &= \frac{\frac{1}{\sigma^2} \sum_{j=1}^{k} z(j)\gamma_j + \frac{1}{\sigma_x^2} \mu_x}{\frac{1}{\sigma^2} \sum_{j=1}^{k} \gamma_j^2 + \frac{1}{\sigma_x^2}} \end{aligned}$$

It is worth noting that the ML and MAP estimators coincide as $\sigma_x \to \infty$.

Note the posterior pdf can be rewritten as

$$p(\boldsymbol{x}|Z^k) = \frac{1}{\sqrt{2\pi}\sigma_{\text{new}}} \exp \left[\frac{-1}{2\sigma_{\text{new}}^2} (x - \hat{x}_{\text{MAP}})^2 \right],$$

where

$$\sigma_{\text{new}}^2 \triangleq \frac{1}{\frac{1}{\sigma^2} \sum_{j=1}^{k} \gamma_j^2 + \frac{1}{\sigma_x^2}} = \frac{\sigma_x^2}{\frac{\sigma_x^2}{\sigma^2} \sum_{j=1}^{k} \gamma_j^2 + 1} < \sigma_x^2$$

This form reveals that, compared to the prior pdf, the posterior pdf has reduced uncertainty: the information in the data has tightened the distribution.

Remarks. Working directly with pdfs is often messy.

5.4 Least-Squares (LS) Estimation

LS is an entirely different philosophy from ML and MAP: LS is a geometric approach to estimation; it pretends to know nothing about probability or distributions. The LS strategy is to minimize a cost function equal to the weighted sum of the squares of the error $\boldsymbol{\epsilon}$ between the data and the model:

$$\hat{\boldsymbol{x}}_{\text{LS}} = \arg\min_{\boldsymbol{x}} C(x, Z^k) \triangleq \boldsymbol{\epsilon}^T W \boldsymbol{\epsilon} = \|\boldsymbol{\epsilon}\|_W^2$$

Said a different way, the LS estimate minimizes the Euclidean norm of the error vector $\boldsymbol{\epsilon}$ weighted by the matrix W.

Example. LS Estimator for a Linear First-Order Dynamical System Let us apply the LS estimator to the linear first-order dynamical system of the previous examples:

$$\hat{x}_{\text{LS}} = \arg\min_x C(x, Z^k) \triangleq \frac{1}{2\sigma^2} \sum_{j=1}^{k} [z(j) - \gamma_j x]^2$$

Differentiate $C(x, Z^k)$ and set to zero:

$$\frac{\partial C}{\partial x} = \frac{1}{\sigma^2} \sum_{j=1}^{k} [z(j) - \gamma_j x] (-\gamma_j) = 0$$

Now solve for x:

$$\hat{x}_{\text{LS}} = \frac{\sum_{j=1}^{k} z(j)\gamma_j}{\sum_{j=1}^{k} \gamma_j^2}.$$

Remarks. The LS estimate turns out to be identical to the ML estimate in this case. The connection is that, for Gaussian random variables, ML estimation corresponds to a Euclidean distance metric. This is because maximizing $p(Z^k|x)$ is the same as maximizing

$$\log p(Z^k|x) = \text{const} - \underbrace{\left\{ \frac{1}{2\sigma^2} \sum_{j=1}^{k} [z(j) - \gamma_j(x)]^2 \right\}}_{C(x, Z^k)}$$

so minimizing $C(x, z^k)$ maximizes $p(Z^k|x)$ when the weighting is chosen appropriately.

5.5 Minimum Mean-Squared Error (MMSE) Estimation

Assume that $p(x)$ and $p(Z^k|x)$ are known so that we can determine $p(x|Z^k)$. (For this introduction, we'll focus only on the case where x is a scalar.) The MMSE strategy is to minimize the average squared difference between some number $\hat{x}$ and the random variable x when conditioned on Z^k:

$$\hat{x}_{\text{MMSE}} = \arg\min_{\hat{x}} C(\hat{x}, Z^k) \triangleq \mathbb{E}\left[(\hat{x} - x)^2 | Z^k\right] = \int (\hat{x} - x)^2 p(x|Z^k) dx$$

We apply the usual procedure to minimize C:

$$\begin{aligned} 0 = \frac{\partial C}{\partial \hat{x}} &= 2\left[\int \hat{x} p(x|Z^k) dx - \int x p(x|Z^k) dx\right] \\ &= 2\left[\hat{x} - \mathbb{E}\left[x|Z^k\right]\right] \\ &\boxed{\Rightarrow \hat{x}_{\text{MMSE}} = \mathbb{E}\left[x|Z^k\right]} \end{aligned}$$

Q: Under what conditions is $\hat{x}_{\text{MMSE}} = \hat{x}_{\text{MAP}}$?

A: Whenever the mean and the peak (mode) of $p(x|Z^k)$ coincide. [Draw picture.]

Remarks. MMSE estimation is a special case of a Bayesian estimation technique designed to minimize conditional risk:

$$\mathcal{R}_B(\hat{x}|Z^k) \triangleq \int_{-\infty}^{\infty} C_B[\hat{x}(k, Z^k) - x] p(x|Z^k) dx$$

The Bayesian cost function $C_B[x - \hat{x}(k, Z^k)]$ can be defined in many reasonable ways; for MMSE,

$$C_B[x - \hat{x}(k, Z^k)] = [\hat{x}(k, Z^k) - x]^2$$

A quadratic cost function is a common choice for C_B because of the relative ease of finding an $\hat{x}$ that minimizes $\mathcal{R}_B(\hat{x}|Z^k)$, but it's not the only option. In fact, for robust estimation in the face of wild outliers, a quadratic cost is a poor choice. We may instead wish to choose a cost that attenuates the effect of outliers or ignores them completely.

Summary

- LS $\Leftrightarrow$ ML for Gaussian measurement noise (when the LS cost is properly weighted).
- ML $\Leftrightarrow$ MAP if a diffuse prior is assumed.
- MAP $\Leftrightarrow$ MMSE if the mean and the mode of $p(\boldsymbol{x}|Z^k)$ coincide.
- ML $\Leftrightarrow$ MAP $\Leftrightarrow$ LS $\Leftrightarrow$ MMSE for Gaussian random variables with $p(\boldsymbol{x})$ diffuse.

5.6 Unbiased Estimators

Let $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$ be the *estimation error*. An estimator is said to be *unbiased* if $\mathbb{E}[\tilde{\boldsymbol{x}}] = 0$.

Remarks. The estimate $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(k, Z^k)$ is random because of the random measurement errors, but $\hat{\boldsymbol{x}}(k, Z^k)$ might have additional randomness if $\boldsymbol{x}$ is a new sample from the distribution $p(\boldsymbol{x})$ for each experiment. When we take $\mathbb{E}[\tilde{\boldsymbol{x}}]$, we are taking the expectation with respect to (1) random measurement errors and, (2) if $\boldsymbol{x}$ is thought of as random, with respect to $\boldsymbol{x}$.

Example. Non-Bayesian case:

In this example, $\boldsymbol{x}$ is considered an unknown constant.

$$\begin{aligned}
\tilde{x} = x - \hat{x}_{\mathrm{ML}}(Z^k) &= x - \frac{\sum_j z(j)\gamma_j}{\sum_j \gamma_j^2} = x - \frac{\sum_j \{[\gamma_j x + w(j)]\gamma_j\}}{\sum_j \gamma_j^2} \\
&= x - \frac{x\sum_j \gamma_j^2}{\sum_j \gamma_j^2} - \frac{\sum_j w(j)\gamma_j}{\sum_j \gamma_j^2} \\
&\Rightarrow \mathbb{E}[\tilde{x}] = -\frac{\sum_j \mathbb{E}[w(j)]\,\gamma_j}{\sum_j \gamma_j^2} = 0 \\
&\Rightarrow \text{Estimator is unbiased in this case}
\end{aligned}$$

Example. Bayesian case:

Now we assume x is a random sample from $p(x) = \mathcal{N}(x; \mu_x, \sigma_x^2)$.

$$\tilde{x} = x - \hat{x}_{\text{MAP}} = x - \left(\frac{\frac{1}{\sigma^2}\sum_j z(j)\gamma_j + \frac{1}{\sigma_x^2}\mu_x}{\underbrace{\frac{1}{\sigma^2}\sum_j \gamma_j^2 + \frac{1}{\sigma_x^2}}_{\beta}} \right) = x - \left(\frac{\frac{1}{\sigma^2}\sum_j [\gamma_j x + w(j)]\gamma_j + \frac{1}{\sigma_x^2}\mu_x}{\beta} \right)$$

$$= \frac{\frac{1}{\sigma_x^2}(x - \mu_x)}{\beta} - \frac{\frac{1}{\sigma^2}\sum_j \gamma_j w(j)}{\beta} \qquad \text{(This is correct–just work it out)}$$

$$\Rightarrow \mathbb{E}[\tilde{x}] = \frac{1}{\beta}\left[\frac{1}{\sigma_x^2}\mathbb{E}[x - \mu_x] - \frac{1}{\sigma^2}\sum_j \gamma_j \mathbb{E}[w(j)] \right] = 0 \qquad \text{(Note the different use of } \mathbb{E}[\cdot]\text{)}$$

$\Rightarrow$ Estimator is also unbiased in this case

5.7 Variance and Mean Square Error of an Estimator

Mean square error (MSE): $\mathbb{E}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T\right]$
Covariance: $\mathbb{E}\left[(\tilde{\boldsymbol{x}} - \bar{\tilde{\boldsymbol{x}}})(\tilde{\boldsymbol{x}} - \bar{\tilde{\boldsymbol{x}}})^T\right] = \mathbb{E}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T\right] - \bar{\tilde{\boldsymbol{x}}}\bar{\tilde{\boldsymbol{x}}}^T$

Remarks.

- For an unbiased estimator, MSE = Covariance
- For biased estimators, MSE $\geq$ Covariance (in the positive definite sense)

[TTT: Note: much missing material here]

5.8 Consistency of Estimators

[TTT: Note: missing material here]

5.9 Fisher Information Matrix, Cramer Rao Lower Bound, Efficiency

Estimation theorists are haunted by the question "Is there a better estimator?"

Q: Suppose $\boldsymbol{x}$ is viewed as a deterministic constant. What criteria would you offer for the best estimator? At a minimum, it should be unbiased? What else? One might be tempted to say "the one that produces an estimate closest to the true value of the parameter to be estimated," but this ignores the fact that the estimate is a random variable—we wouldn't want to declare our estimator the best on the basis of a lucky outcome. Instead, we might declare the best estimator to be the one that minimizes

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}(k, Z^k)\tilde{\boldsymbol{x}}^T(k, Z^k)\right]$$

in the positive definite sense. Is this a good definition for best?
A: For a Gaussian-distributed $\tilde{\boldsymbol{x}}(k, Z^k)$, yes: for a Gaussian, the variance is a unique measure of dispersion. For non-Gaussian-distributed $\tilde{\boldsymbol{x}}(k, Z^k)$ this is still a reasonable criterion, though others

could also be offered. ■

[TTT: Add example.]

Fisher Information Matrix (FIM) (for $\boldsymbol{x}$ a deterministic constant) Recall that in the ML framework ($\boldsymbol{x}$ a deterministic constant), we develop our estimator based on the likelihood function $\Lambda_{Z^k}(\boldsymbol{x}) \triangleq p(Z^k|\boldsymbol{x})$, which we write as $\Lambda(\boldsymbol{x})$ for short. Each element of the FIM is larger if, on average, the likelihood function is more sensitive to changes in the corresponding elements of $\boldsymbol{x}$. Fisher's great insight was to recognize that greater average sensitivity indicates more information about $\boldsymbol{x}$.

For a scalar x, the FIM, written J is defined as

$$J \triangleq -\mathbb{E}\left[\left.\frac{\partial^2 \log \Lambda(x)}{\partial x^2}\right|_{x_0}\right] = \mathbb{E}\left[\left.\left(\frac{\partial \log \Lambda(x)}{\partial x}\right)^2\right|_{x_0}\right] \tag{5.2}$$

where x_0 is the true value of the unknown constant x. This equation generalizes for a vector $\boldsymbol{x}$, but it's more complicated to write, as it involves a Hessian. However, it's easy to define the (i,j)th element of J, written J_{ij}. Let

$$H_{x_i} \triangleq \left.\frac{\partial \log \Lambda(\boldsymbol{x})}{\partial x_i}\right|_{\boldsymbol{x}_0}$$

Then

$$J_{ij} = E[H_{x_i} H_{x_j}] \tag{5.3}$$

Note that if $\boldsymbol{x}_0$ is not available, as is often the case in practice, then we can approximate J by evaluating at the estimate $\hat{\boldsymbol{x}}$. We can think of FIM as quantifying the maximum amount of information we could possibly squeeze out of the data.

Cramer-Rao Lower Bound (CRLB) For any unbiased estimator

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}(k, Z^k)\tilde{\boldsymbol{x}}^T(k, Z^k)\right] \geq J^{-1}$$

in the positive definite sense. This bound is true for *any* unbiased estimator when there is no *a priori* information about $\boldsymbol{x}$.

Efficient Estimator An efficient estimator achieves the CRLB:

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}(k, Z^k)\tilde{\boldsymbol{x}}^T(k, Z^k)\right] = J^{-1}$$

Bayesian Case If $\boldsymbol{x}$ is viewed as a random variable, with $p(\boldsymbol{x})$ its known prior distribution, the FIM is instead given as

$$J = \mathbb{E}\left[\left(\left.\frac{\partial}{\partial \boldsymbol{x}} \log p(Z^k, \boldsymbol{x})\right|_{x_0}\right)\left(\left.\frac{\partial}{\partial \boldsymbol{x}} \log p(Z^k, \boldsymbol{x})\right|_{x_0}\right)^T\right] \tag{5.4}$$

Note that since $p(Z^k, \boldsymbol{x}) = p(\boldsymbol{x}|Z^k)p(Z^k)$ and $\frac{\partial}{\partial \boldsymbol{x}} p(Z^k) = 0$, we could use $p(\boldsymbol{x}|Z^k)$ just as well as $p(Z^k, \boldsymbol{x})$ in the expression for J.

Example. Non-Bayesian case:

Consider the problem of estimating an unknown frequency parameter ω from complex measurements

$$z_k = \exp(j\omega kT) + n_k, \quad k = 0, 1, ..., N-1$$

where $j = \sqrt{-1}$ and where $n_k = a_k + jb_k$ is a sequence of independent, identically-distributed complex zero-mean Gaussian noise samples with variance σ^2:

$$a_k, b_k \sim \mathcal{N}(0, \sigma^2), \quad \mathbb{E}\left[a_k a_j\right] = \sigma^2 \delta_{kj}, \quad \mathbb{E}\left[b_k b_j\right] = \sigma^2 \delta_{kj}, \quad \mathbb{E}\left[a_k b_j\right] = 0 \ \ \forall k, j$$

Here, δ_{kj} is the Kronecker delta (equal to unity for $k = j$ and otherwise zero).

Suppose the N measurements are stacked as a complex vector $\boldsymbol{z} = \boldsymbol{x} + j\boldsymbol{y}$, where $\boldsymbol{z} = [z_0, z_1, ..., z_{N-1}]^T$, and where $\boldsymbol{x} = [x_0, x_1, ..., x_{N-1}]^T$ and $\boldsymbol{y} = [y_0, y_1, ..., y_{N-1}]^T$ are the real and imaginary components of $\boldsymbol{z}$.

We wish to find the CRLB for an ML estimate of ω. Since $\boldsymbol{z}$ is Gaussian, the likelihood function is simply

$$\Lambda(\omega) \triangleq p(\boldsymbol{z}|\omega) = \left(\frac{1}{2\pi\sigma^2}\right)^N \exp\left(-\frac{1}{2\sigma^2}[\boldsymbol{z} - \boldsymbol{\mu}_z]^H [\boldsymbol{z} - \boldsymbol{\mu}_z]\right)$$

where $\boldsymbol{\mu}_z = \mathbb{E}\left[\boldsymbol{z}\right]$ and where $\boldsymbol{x}^H$ represents the conjugate transpose of the vector $\boldsymbol{x}$. Expressing the inner product as a summation, and recognizing that $\exp(j\omega kT) = \cos(\omega kT) + j\sin(\omega kT)$, $\Lambda(\omega)$ may be written

$$\Lambda(\omega) = \left(\frac{1}{2\pi\sigma^2}\right)^N \exp\left(-\frac{1}{2\sigma^2}\sum_{k=0}^{N-1} \left[x_k - \cos(\omega kT)\right]^2 + \left[y_k - \sin(\omega kT)\right]^2\right)$$

Taking the natural log, we have

$$\log\Lambda(\omega) = C - \frac{1}{2\sigma^2}\sum_{k=0}^{N-1} \left[x_k - \cos(\omega kT)\right]^2 + \left[y_k - \sin(\omega kT)\right]^2$$

where C is a constant that does not depend on ω. Now taking the derivative with respect to ω, we have

$$H_\omega \triangleq \frac{d\log\Lambda(\omega)}{d\omega} = -\frac{T}{\sigma^2}\sum_{k=0}^{N-1} ka_k \sin(\omega kT) - kb_k \cos(\omega kT)$$

Recognizing that the cross-terms involving a_k and b_k vanish when calculating $\mathbb{E}\left[H_\omega^2\right]$, we have

$$J \triangleq \mathbb{E}\left[H_\omega^2\right] = \frac{T^2}{\sigma^4}\sum_{k=0}^{N-1} k^2\sigma^2\left[\sin^2(\omega kT) + \cos^2(\omega kT)\right]$$

which reduces to $J = QT^2/\sigma^2$, where

$$Q = \sum_{k=0}^{N-1} k^2 = \frac{N(N-1)(2N-1)}{6}$$

Thus, the CRLB is

$$\text{CRLB} \triangleq J^{-1} = \frac{\sigma^2}{QT^2}$$

Chapter 6

Linear estimation for static systems

6.1 MAP estimator for Gaussian problems

Given a system model $\boldsymbol{z} = H\boldsymbol{x} + \boldsymbol{w}$. Suppose we wish to estimate $\boldsymbol{x} \in \mathbb{R}^{n_x}$. Assume that $\boldsymbol{w} \sim \mathcal{N}(0, R)$ is $n_z \times 1$ noise vector, and H is a known $n_z \times n_x$ matrix. We also have *a priori* knowledge that $\boldsymbol{x} \sim \mathcal{N}(\bar{\boldsymbol{x}}, P_{\boldsymbol{xx}})$, and we assume that $\mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})\boldsymbol{w}^T\right] = 0$, i.e., $(\boldsymbol{x} - \bar{\boldsymbol{x}})$ and $\boldsymbol{w}$ are uncorrelated.

Our approach to this problem is to develop a joint pdf for $\boldsymbol{x}$ and $\boldsymbol{z}$, then use our understanding of conditional Gaussian distributions to determine $p(\boldsymbol{x} \mid \boldsymbol{z})$. Thereby, we can find $\hat{\boldsymbol{x}}_{\text{MAP}}$ such that

$$\hat{\boldsymbol{x}}_{\text{MAP}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x} \mid \boldsymbol{z}) \tag{6.1}$$

In order to find $p(\boldsymbol{x} \mid \boldsymbol{z})$, we first need $p(\boldsymbol{x}, \boldsymbol{z})$, so let's define some things that will help us get there. First, let's find $\boldsymbol{z}$.

$$\begin{aligned}
\bar{\boldsymbol{z}} &= \mathbb{E}\left[\boldsymbol{z}\right] && (6.2)\\
&= \mathbb{E}\left[H\boldsymbol{z} + \boldsymbol{w}\right] && (6.3)\\
&= H\mathbb{E}\left[\boldsymbol{x}\right] + \mathbb{E}\left[\boldsymbol{w}\right] && (6.4)\\
&= H\bar{\boldsymbol{x}} && (6.5)
\end{aligned}$$

Next, we need covariance matrices $P_{\boldsymbol{xz}}$, $P_{\boldsymbol{zx}}$, and $P_{\boldsymbol{zz}}$.

$$\begin{aligned}
P_{\boldsymbol{xz}} &= \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{z} - \bar{\boldsymbol{z}})^T\right] && (6.6)\\
&= \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(H\boldsymbol{x} + \boldsymbol{w} - H\bar{\boldsymbol{x}})^T\right] && (6.7)\\
&= \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T H^T + (\boldsymbol{x} - \bar{\boldsymbol{x}})\boldsymbol{w}^T\right] && (6.8)\\
&= \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T H^T\right] + \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})\boldsymbol{w}^T\right] && (6.9)\\
&= P_{\boldsymbol{xx}} H^T && (6.10)\\
& && (6.11)
\end{aligned}$$

Because of the symmetry of covariance matrices, we can also say that

$$P_{\boldsymbol{zx}} = P_{\boldsymbol{xz}}^T = HP_{\boldsymbol{xx}} \tag{6.12}$$

For $P_{\boldsymbol{zz}}$ we have

$$P_{\boldsymbol{zz}} = \mathbb{E}\left[(\boldsymbol{z}-\bar{\boldsymbol{z}})(\boldsymbol{z}-\bar{\boldsymbol{z}})^T\right] \tag{6.13}$$
$$= \mathbb{E}\left[(H(\boldsymbol{x}-\bar{\boldsymbol{x}})+\boldsymbol{w})(H(\boldsymbol{x}-\bar{\boldsymbol{x}})+\boldsymbol{w})^T\right] \tag{6.14}$$
$$= HP_{\boldsymbol{xx}}H^T + R \tag{6.15}$$

Now, we can define $p(\boldsymbol{x},\boldsymbol{z})$, from which we can find $p(\boldsymbol{x} \mid \boldsymbol{z})$.

$$p(\boldsymbol{x} \mid \boldsymbol{z}) = \frac{p(\boldsymbol{x},\boldsymbol{z})}{p(\boldsymbol{z})} \tag{6.16}$$
$$= c(\boldsymbol{z}) \exp\left\{-\frac{1}{2}\begin{bmatrix} (\boldsymbol{x}-\bar{\boldsymbol{x}})^T & (\boldsymbol{z}-\bar{\boldsymbol{z}})^T \end{bmatrix} \begin{bmatrix} P_{\boldsymbol{xx}} & P_{\boldsymbol{xz}} \\ P_{\boldsymbol{zx}} & P_{\boldsymbol{zz}} \end{bmatrix}^{-1} \begin{bmatrix} (\boldsymbol{x}-\bar{\boldsymbol{x}}) \\ (\boldsymbol{z}-\bar{\boldsymbol{z}}) \end{bmatrix}\right\} \tag{6.17}$$

Recall from the linear algebra review that

$$\begin{bmatrix} P_{\boldsymbol{xx}} & P_{\boldsymbol{xz}} \\ P_{\boldsymbol{zx}} & P_{\boldsymbol{zz}} \end{bmatrix}^{-1} = \begin{bmatrix} V_{xx} & V_{xz} \\ V_{zx} & V_{zz} \end{bmatrix} \tag{6.18}$$

where

$$V_{xx} = (P_{\boldsymbol{xx}} - P_{\boldsymbol{xz}}P_{\boldsymbol{zz}}^{-1}P_{\boldsymbol{xz}}^T)^{-1} \tag{6.19}$$
$$V_{xz} = -V_{xx}P_{\boldsymbol{xz}}P_{\boldsymbol{zz}}^{-1} \tag{6.20}$$
$$V_{zz} = (P_{\boldsymbol{zz}} - P_{\boldsymbol{xz}}P_{\boldsymbol{xx}}^{-1}P_{\boldsymbol{xz}})^{-1} \tag{6.21}$$

Now we can find $\hat{\boldsymbol{x}}_{\text{MAP}}$ by maximizing $p(\boldsymbol{x} \mid \boldsymbol{z})$, or equivalently, we can minimize over $\boldsymbol{x}$

$$-\log p(\boldsymbol{x} \mid \boldsymbol{z}) = \text{const} + C(\boldsymbol{x} \mid \boldsymbol{z}) \tag{6.22}$$
$$C(\boldsymbol{x} \mid \boldsymbol{z}) = \frac{1}{2}\begin{bmatrix} (\boldsymbol{x}-\bar{\boldsymbol{x}})^T & (\boldsymbol{z}-\bar{\boldsymbol{z}})^T \end{bmatrix} \begin{bmatrix} V_{xx} & V_{xz} \\ V_{zx} & V_{zz} \end{bmatrix} \begin{bmatrix} (\boldsymbol{x}-\bar{\boldsymbol{x}}) \\ (\boldsymbol{z}-\bar{\boldsymbol{z}}) \end{bmatrix} \tag{6.23}$$

Note that $C(\boldsymbol{x} \mid \boldsymbol{z})$ is a scalar value. Set $\frac{\partial C}{\partial \boldsymbol{x}} = 0$ to get

$$0 = \frac{\partial C}{\partial \boldsymbol{x}} \tag{6.24}$$
$$= \begin{bmatrix} \frac{\partial C}{\partial x_1} \\ \vdots \\ \frac{\partial C}{\partial x_{n_x}} \end{bmatrix} \tag{6.25}$$
$$= V_{xx}(\boldsymbol{x}-\bar{\boldsymbol{x}}) + V_{xz}(\boldsymbol{z}-\bar{\boldsymbol{z}}) \tag{6.26}$$

Now we solve for $\hat{\boldsymbol{x}}_{MAP}$ to get

$$\hat{\boldsymbol{x}}_{\text{MAP}} = \bar{\boldsymbol{x}} - V_{xx}^{-1}V_{xz}(\boldsymbol{z}-\bar{\boldsymbol{z}}) \tag{6.27}$$
$$= \bar{\boldsymbol{x}} + P_{\boldsymbol{xz}}P_{\boldsymbol{zz}}^{-1}(\boldsymbol{z}-\bar{\boldsymbol{z}}) \tag{6.28}$$

Using our original formulas for the covariance matrices, we get

$$\hat{\boldsymbol{x}}_{\text{MAP}} = \bar{\boldsymbol{x}} + P_{\boldsymbol{xx}}H^T(HP_{\boldsymbol{xx}}H^T + R)^{-1}(\boldsymbol{z}-H\bar{\boldsymbol{x}}) \tag{6.29}$$

Analysis of MAP Estimate Let's find $\tilde{\boldsymbol{x}}$, $\mathbb{E}[\tilde{\boldsymbol{x}}]$, and $P_{\boldsymbol{xx}|\boldsymbol{z}}$.

$$\tilde{\boldsymbol{x}} \triangleq \boldsymbol{x} - \hat{\boldsymbol{x}} \tag{6.30}$$
$$= (\boldsymbol{x} - \bar{\boldsymbol{x}}) - P_{\boldsymbol{xx}} H^T (H P_{\boldsymbol{xx}} H^T + R)^{-1} (\boldsymbol{z} - H\bar{\boldsymbol{x}}) \tag{6.31}$$
$$\mathbb{E}[\tilde{\boldsymbol{x}}] = \underbrace{\mathbb{E}[\boldsymbol{x} - \bar{\boldsymbol{x}}]}_{=0} - P_{\boldsymbol{xx}} H^T (H P_{\boldsymbol{xx}} H^T + R)^{-1} \underbrace{\mathbb{E}[\boldsymbol{z} - H\bar{\boldsymbol{x}}]}_{=0} \tag{6.32}$$
$$= 0 \tag{6.33}$$

Because $\mathbb{E}[\tilde{\boldsymbol{x}}] = 0$, $\hat{\boldsymbol{x}}_{\text{MAP}}$ is an unbiased estimator.

$$P_{\boldsymbol{xx}|\boldsymbol{z}} \triangleq \mathbb{E}\left[\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T\right] \tag{6.34}$$
$$= \mathbb{E}\left[(\boldsymbol{x} - \hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}})^T\right] \tag{6.35}$$
$$= \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T\right] - P_{\boldsymbol{xz}} P_{\boldsymbol{zz}}^{-1} \mathbb{E}\left[(\boldsymbol{z} - \bar{\boldsymbol{z}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T\right] \tag{6.36}$$
$$- \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{z} - \bar{\boldsymbol{z}})^T\right] P_{\boldsymbol{zz}}^{-1} P_{\boldsymbol{xz}}^T + P_{\boldsymbol{xz}} P_{\boldsymbol{zz}}^{-1} \mathbb{E}\left[(\boldsymbol{z} - \bar{\boldsymbol{z}})(\boldsymbol{z} - \bar{\boldsymbol{z}})^T\right] P_{\boldsymbol{zz}}^{-1} P_{\boldsymbol{xz}}^T \tag{6.37}$$
$$= P_{\boldsymbol{xx}} - P_{\boldsymbol{xz}} P_{\boldsymbol{zz}}^{-1} P_{\boldsymbol{xz}}^T \tag{6.38}$$

Therefore,

$$\boxed{P_{\boldsymbol{xx}|\boldsymbol{z}} = P_{\boldsymbol{xx}} - P_{\boldsymbol{xx}} H^T (H P_{\boldsymbol{xx}} H^T + R)^{-1} H P_{\boldsymbol{xx}}} \tag{6.39}$$

Note that $P_{\boldsymbol{xx}|\boldsymbol{z}} < P_{\boldsymbol{xx}}$, implying we are getting a better result. However, if the elements of R become too large (the noise is too powerful), $P_{\boldsymbol{xx}|\boldsymbol{z}} = P_{\boldsymbol{xx}}$. Note also that the above analysis assumes $\boldsymbol{x}$ and $\boldsymbol{w}$ are Gaussian distributed.

Q: What if $\boldsymbol{x}$ or $\boldsymbol{w}$ is not Gaussian? What can we say about the form of the estimator then?
A: See Bar-Shalom, section 3.3. Even if they're not Gaussian, if $\mathbb{E}[\boldsymbol{w}] = 0$ and we know $\bar{\boldsymbol{x}}$, $P_{\boldsymbol{xx}}$, and R, then $\hat{\boldsymbol{x}}_{\text{MAP}}$ found from assuming Gaussian distribution is the optimal linear estimator, i.e., an estimator which is a linear combination of $\bar{\boldsymbol{x}}$ and $\boldsymbol{z}$ taking the form $C\bar{\boldsymbol{x}} + D\boldsymbol{z}$ for some best choice of C and D, specifically

$$\hat{\boldsymbol{x}}_{\text{MAP}} = \bar{\boldsymbol{x}} + P_{\boldsymbol{xz}} P_{\boldsymbol{zz}}^{-1} (\boldsymbol{z} - \bar{\boldsymbol{z}}) \tag{6.40}$$
$$= (I - P_{\boldsymbol{xz}} P_{\boldsymbol{zz}}^{-1} H)\bar{\boldsymbol{x}} + P_{\boldsymbol{xz}} P_{\boldsymbol{zz}}^{-1} \boldsymbol{z} \tag{6.41}$$

However, there may still be a better non-linear estimator. ■

6.2 Batch Least Squares Estimation

Given measurements $\boldsymbol{z}(i) = H(i)\boldsymbol{x} + \boldsymbol{w}(i)$ (with no *a priori* $\boldsymbol{x}$ distribution). Suppose that

$$\boldsymbol{w}(i) \sim \mathcal{N}(0, R(i)) \tag{6.42}$$
$$\mathbb{E}[\boldsymbol{w}(i)\boldsymbol{w}(j)] = 0, \forall i \neq j \tag{6.43}$$
$$R(i) = R^T(i) > 0 \tag{6.44}$$

Define a (scalar) cost function $J(k)$ as

$$J(k) = \sum_{i=1}^{k} [\boldsymbol{z}(i) - H(i)\boldsymbol{x}]^T R^{-1}(i) [\boldsymbol{z}(i) - H(i)\boldsymbol{x}] \tag{6.45}$$

We want to use this cost function to deemphasize any noisy measurements.

Note that now we have a time index, and no a priori knowledge.

$$p(\boldsymbol{z}^k \mid \boldsymbol{x}) = C \exp^{-\frac{1}{2}J(k)} \tag{6.46}$$

$$J(k) = -2\log[p(\boldsymbol{z}^k \mid \boldsymbol{x})] + \text{const} \tag{6.47}$$

So, minimizing $J(k)$ with respect to $\boldsymbol{x}$ is equivalent to maximizing $\Lambda(\boldsymbol{x}) = p(\boldsymbol{Z}^k \mid \boldsymbol{x})$ (ML estimator).

We need a change of notation to incorporate data and parameters for each new time step.

$$\boldsymbol{Z}^{(k)} = \begin{bmatrix} \boldsymbol{z}(1) \\ \boldsymbol{z}(2) \\ \vdots \\ \boldsymbol{z}(k) \end{bmatrix} \in \mathbb{R}^{n_z \cdot k} \tag{6.48}$$

$$H^k = \begin{bmatrix} H(1) \\ H(2) \\ \vdots \\ H(k) \end{bmatrix} \in \mathbb{R}^{n_z \cdot k \times n_x} \tag{6.49}$$

$$\boldsymbol{W}^k = \begin{bmatrix} \boldsymbol{w}(1) \\ \boldsymbol{w}(2) \\ \vdots \\ \boldsymbol{w}(k) \end{bmatrix} \in \mathbb{R}^{n_z \cdot k} \tag{6.50}$$

$$R^k = \begin{bmatrix} R(1) & 0 & \cdots & 0 \\ 0 & R(2) & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & R(k) \end{bmatrix} \in \mathbb{R}^{n_z \cdot k \times n_z \cdot k} \tag{6.51}$$

$$\tag{6.52}$$

Then, we can rewrite $J(k)$ as

$$J(k) = [\boldsymbol{Z}^k - H^k \boldsymbol{x}]^T (R^k)^{-1} [\boldsymbol{Z}^k - H^k \boldsymbol{x}] \tag{6.53}$$

The summation notation is now gone, but $J(k)$ is the same as before, and still a scalar value. Now, to minimize $J(k)$,

$$0 = \frac{\partial J}{\partial \boldsymbol{x}} \tag{6.54}$$

$$= \begin{bmatrix} \frac{\partial J}{\partial x_1} \\ \frac{\partial J}{\partial x_2} \\ \vdots \\ \frac{\partial J}{\partial x_{n_x}} \end{bmatrix} \tag{6.55}$$

$$= 2(-H^k)(R^k)^{-1}(\boldsymbol{Z}^k - H^k \boldsymbol{x}) \tag{6.56}$$

This gives us n_x number of equations and n_x unknowns. Next, we solve for $\hat{\boldsymbol{x}}^k$ as

$$\hat{\boldsymbol{x}}^k = [(H^k)^T (R^k)^{-1} H^k]^{-1} (H^k)^T (R^k)^{-1} Z^k \tag{6.57}$$

Therefore,

$$\boxed{\hat{\boldsymbol{x}}^k = (H^T R^{-1} H)^{-1} H^T R^{-1} \boldsymbol{z}} \tag{6.58}$$

By dropping the k superscripts, we get one of the deathbed identities, the normal equations.

Properties of the Least Squares Estimator

The estimation error $\tilde{x} \triangleq \boldsymbol{x} - \hat{\boldsymbol{x}}$ is given by

$$\begin{aligned}
\tilde{\boldsymbol{x}} &= \boldsymbol{x} - \hat{\boldsymbol{x}} && (6.59)\\
&= \boldsymbol{x} - (H^T R^{-1} H)^{-1} H^T R^{-1} \boldsymbol{z} && (6.60)\\
&= \boldsymbol{x} - (H^T R^{-1} H)^{-1} H^T R^{-1} (H\boldsymbol{x} + \boldsymbol{w}) && (6.61)\\
&= [I - (H^T R^{-1} H)^{-1} H^T R^{-1} H]\boldsymbol{x} - (H^T R^{-1} H)^{-1} H^T R^{-1} \boldsymbol{w} && (6.62)
\end{aligned}$$

We can find expectation of $\tilde{\boldsymbol{x}}$ as

$$\begin{aligned}
\mathbb{E}[\tilde{\boldsymbol{x}}] &= -(H^T R^{-1} H)^{-1} H^T R^{-1} \mathbb{E}[\boldsymbol{w}] && (6.63)\\
&= 0 && (6.64)
\end{aligned}$$

Therefore, the LS estimator is unbiased.

We can also find the covariance as

$$\begin{aligned}
P_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}} &= (H^T R^{-1} H)^{-1} H^T R^{-1} \mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^T\right] R^{-1} H (H^T R^{-1} H)^{-1} && (6.65)\\
&= (H^T R^{-1} H)^{-1} && (6.66)
\end{aligned}$$

6.3 Square-Root-Based Least Squares Estimation

Refer to Bierman for more detail on this topic. Recall that for a solution to exist, $H^T R^{-1} H$ must be invertible, which amounts to the parameter $\boldsymbol{x}$ being observable. In some cases, $\boldsymbol{x}$ is only locally observable from the data, meaning that $H^T R^{-1} H$ is **almost** singular. In these cases, it's a bad idea to directly invert $H^T R^{-1} H$; small numerical errors can lead to large errors in $\hat{\boldsymbol{x}}$.

The solution to this problem is to use square root algorithms. They are more numerically robust, and also lead to a more elegant and intuitive interpretation of least squares.

Let $\boldsymbol{z'} = H'\boldsymbol{x} + \boldsymbol{w'}$, $\boldsymbol{w'} \sim \mathcal{N}(\bar{\boldsymbol{x}}, R)$, $R = R^T$, $R > 0$. Use the Cholesky factorization, $R_a^T R_a = R$. In MATLAB:

```
R_{a} = chol(R);
```

Then let

$$\begin{aligned}
\boldsymbol{z} &= (R_a^{-1})^T \boldsymbol{z'} && (6.67)\\
&= (R_a^T)^{-1} \boldsymbol{z'} && (6.68)\\
&= R_a^{-T} \boldsymbol{z'} && (6.69)\\
H &= R_a^{-T} H' && (6.70)\\
\boldsymbol{w} &= R_a^{-T} \boldsymbol{w'} && (6.71)
\end{aligned}$$

Now we have a transformed measurement model,

$$\boldsymbol{z} = H\boldsymbol{x} + \boldsymbol{w} \tag{6.72}$$
$$\mathbb{E}[\boldsymbol{w}] = \mathbb{E}\left[R_a^{-T}\boldsymbol{w'}\right] = 0 \tag{6.73}$$
$$\mathbb{E}[\boldsymbol{w}\boldsymbol{w'}] = \mathbb{E}\left[R_a^{-T}\boldsymbol{w'}\boldsymbol{w'}^{-T}R_a^{-1}\right] \tag{6.74}$$
$$= R_a^{-T}\mathbb{E}\left[\boldsymbol{w'}\boldsymbol{w'}^{-T}\right]R_a^{-1} \tag{6.75}$$
$$= R_a^{-T}R_a^T R_a R_a^{-1} \tag{6.76}$$
$$= I \tag{6.77}$$

Because $\mathbb{E}[\boldsymbol{w}\boldsymbol{w'}] = I$, $\boldsymbol{w} \sim \mathcal{N}(0, I)$. Now our noise is standard uniform and is nicer to work with. The cost function of our problem is

$$J(k) = [\boldsymbol{z} - H\boldsymbol{x}]^T[\boldsymbol{z} - H\boldsymbol{x}] \tag{6.78}$$
$$= \|H\boldsymbol{x} - \boldsymbol{z}\|^2 \tag{6.79}$$

Recall that multiplying a vector by an orthonormal matrix doesn't change its magnitude

$$\|\boldsymbol{v}\| = \boldsymbol{v}^T\boldsymbol{v} \tag{6.80}$$
$$\|Q\boldsymbol{v}\| = \boldsymbol{v}^T Q^T Q\boldsymbol{v} \tag{6.81}$$
$$= \boldsymbol{v}^T\boldsymbol{v} \tag{6.82}$$
$$= \|\boldsymbol{v}\|^2 \tag{6.83}$$

Q: Can we multiply $H\boldsymbol{x} - \boldsymbol{z}$ by some orthonormal matrix and cleverly simplify the cost function?
A: yes we can; use QR factorization of H. ■

Let $\tilde{Q}\tilde{R} = H$, and define $T = \tilde{Q}^T$ to get

$$J(k) = \|\tilde{Q}^T(H\boldsymbol{x} - \boldsymbol{z})\|^2 \tag{6.84}$$
$$= \|\tilde{R}\boldsymbol{x} - \tilde{\boldsymbol{z}}\|^2 \tag{6.85}$$
$$\tilde{\boldsymbol{z}} = \tilde{Q}^T\boldsymbol{z}$$

We can break this up further as

$$J(k) = \left\| \begin{bmatrix} \tilde{R}_o \\ 0 \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \tilde{\boldsymbol{z}}_o \\ \boldsymbol{\epsilon} \end{bmatrix} \right\|^2 \tag{6.86}$$
$$= \|\tilde{R}_o\boldsymbol{x} - \tilde{\boldsymbol{z}}_o\| + \|\boldsymbol{\epsilon}\|^2 \tag{6.87}$$

How do we minimize this? We solve first for $\boldsymbol{x}$, made possible because if $\text{rank}(H) = n_x$ (which is needed for observability), then $\tilde{R}_o \in \mathbb{R}^{n_x \times n_x}$ and is invertible. We also have the solution

$$\boxed{\hat{\boldsymbol{x}}_{\mathbf{LS}} = \tilde{R}_o^{-1}\tilde{\boldsymbol{z}}_o} \tag{6.88}$$

The solution was obtained without squaring anything. Unfortunately, the component norm of $\|\boldsymbol{\epsilon}\|^2$ is the irreducible part of the cost. In other words,

$$J(k)\Big|_{\hat{\boldsymbol{x}}_{\text{LS}}} = \|\boldsymbol{\epsilon}\|^2 \tag{6.89}$$

Recall our expression for $P_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$ from before

$$
\begin{aligned}
P_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}|\boldsymbol{z}} &= (H'^T R^{-1} H')^{-1} && (6.90)\\
&= (H'^T R_a^{-1} R_a^{-T} H')^{-1} && (6.91)\\
&= (H^T H)^{-1} && (6.92)\\
&= \left[\begin{bmatrix} \tilde{R}_o^T & 0 \end{bmatrix} \tilde{Q}^T \tilde{Q} \begin{bmatrix} \tilde{R}_o^T \\ 0 \end{bmatrix} \right]^{-1} && (6.93)\\
&= (\tilde{R}_o^T \tilde{R}_o)^{-1} && (6.94)\\
&= \tilde{R}_o^{-1} \tilde{R}_o^{-T} && (6.95)
\end{aligned}
$$

Remember that $\tilde{Q}$ is orthonormal, so $\tilde{Q}^T \tilde{Q} = I$. Additionally, we know that $\tilde{R}_o$ can be inverted without problems at this point.

The matrix $H^T H = \tilde{R}_o^T \tilde{R}_o$ is called the **information matrix**, and $\tilde{R}_o$ is the square root information matrix. A "large" $H^T H$ indicates that data in $\boldsymbol{z}$ produces a large amount of information about $\boldsymbol{x}$, leading to a small $P_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}|\boldsymbol{z}}$.

6.3.1 Summary of Square-Root Approach to Least-Squares Estimation

Given the data equation

$$\boldsymbol{z}' = H'\boldsymbol{x} + \boldsymbol{w}', \qquad \boldsymbol{w}' \sim \mathcal{N}(0, R)$$

the square-root batch least-squares approach proceeds through the following steps.

1. Perform a change of coordinates to normalize the data equation. Cholesky factorize R as $R_a^T R_a = R$. Then let $\boldsymbol{z} = R_a^{-T}\boldsymbol{z}'$, $H = R_a^{-T}H'$, and $\boldsymbol{w} = R_a^{-T}\boldsymbol{w}'$ so that

$$\boldsymbol{z} = H\boldsymbol{x} + \boldsymbol{w}, \qquad \boldsymbol{w} \sim \mathcal{N}(0, I)$$

2. Set up the cost function

$$J(\boldsymbol{x}) = \|H\boldsymbol{x} - \boldsymbol{z}\|^2$$

   Here, we write $J(\boldsymbol{x})$ to emphasize that the cost is a function of $\boldsymbol{x}$. We could further write $J(k, \boldsymbol{x})$ if we wished to emphasize that this is the cost after having ingested measurements $\boldsymbol{z}(1)$ through $\boldsymbol{z}(k)$.

   Note that because our change of coordinates yielded $\mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^T\right] = I$, the measurement errors are uncorrelated and have equal variance. Thus, our cost function, which weights all measurements equally, is properly weighted in the maximum likelihood sense. The value of $\boldsymbol{x}$ that minimizes $J(\boldsymbol{x})$ will be our least-squares/ML solution $\hat{\boldsymbol{x}}_{\text{LS}}$.

3. Transform the problem again via an orthonormal matrix T:

$$\begin{aligned} J(\boldsymbol{x}) &= \|H\boldsymbol{x} - \boldsymbol{z}\|^2 \\ &= \|T(H\boldsymbol{x} - \boldsymbol{z})\|^2 \end{aligned}$$

   This transformation does not change $J(\boldsymbol{x})$. We choose a special T that makes solving for $\hat{\boldsymbol{x}}_{\text{LS}}$ easier: Let $T = \tilde{Q}^T$ where $\tilde{Q}\tilde{R} = H$ (QR factorization). Then

$$\begin{aligned} J(\boldsymbol{x}) &= \|\tilde{Q}^T(\tilde{Q}\tilde{R}\boldsymbol{x} - \boldsymbol{z})\|^2 \\ &= \|\tilde{R}\boldsymbol{x} - \tilde{Q}^T\boldsymbol{z}\|^2 \\ &= \left\| \begin{bmatrix} \tilde{R}_o \\ 0 \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \tilde{\boldsymbol{z}}_o \\ \boldsymbol{\epsilon} \end{bmatrix} \right\|^2 \\ &= \|\tilde{R}_o\boldsymbol{x} - \tilde{\boldsymbol{z}}_o\|^2 + \|\boldsymbol{\epsilon}\|^2 \end{aligned}$$

4. Minimize $J(\boldsymbol{x})$. Assuming that $\tilde{R}_o$ is invertible, the minimizing value of $\boldsymbol{x}$ is obvious:

$$\begin{aligned} \hat{\boldsymbol{x}}_{\text{LS}} &= \tilde{R}_o^{-1}\tilde{\boldsymbol{z}}_o \\ J(\hat{\boldsymbol{x}}_{\text{LS}}) &= \|\boldsymbol{\epsilon}\|^2 \end{aligned}$$

   Note that $\|\boldsymbol{\epsilon}\|^2$ cannot be eliminated by any choice of $\boldsymbol{x}$: it is the irreducible cost that arises because the measurement data, due to noise, do not exactly fit the model. Note also that if $\tilde{R}_o$ is not invertible, then the problem is not observable: H (and H') have linearly dependent columns.

5. Analyze the solution:

$$\begin{aligned} P_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}} &= (\tilde{R}_o^T\tilde{R}_o)^{-1} \\ &= \tilde{R}_o^{-1}\tilde{R}_o^{-T} \end{aligned}$$

The so-called information matrix

$$H^T H = \tilde{R}_o^T \tilde{R}_o$$

becomes larger (in the positive definite sense) when the measurements carry more information about $\boldsymbol{x}$. One can show that the batch least squares estimator is efficient for the original linear problem posed. In other words, $H^T H = \tilde{R}_o^T \tilde{R}_o$ is equivalent to the Fisher information matrix in this case.

6.4 Recursive Least Squares Estimation

Suppose we have $\hat{\boldsymbol{x}}(k, Z^k)$ and $\hat{P}(k, Z^k) = P_{\hat{\boldsymbol{x}}\hat{\boldsymbol{x}}|Z^k}$ and we get a new data vector

$$\boldsymbol{z}(k+1) = H(k+1)\boldsymbol{x} + \boldsymbol{w}(k+1), \quad \boldsymbol{w}(k+1) \sim \mathcal{N}(0, R(k+1))$$

Naturally, we'd like to find $\hat{\boldsymbol{x}}(k+1, Z^{k+1})$ without starting from scratch. To set up the problem, let us stack the relevant vectors and matrices as

$$Z^{k+1} = \begin{bmatrix} Z^k \\ \boldsymbol{z}(k+1) \end{bmatrix}, \quad H^{k+1} = \begin{bmatrix} H^k \\ H(k+1) \end{bmatrix}, \quad R^{k+1} = \begin{bmatrix} R^k & 0 \\ 0 & R(k+1) \end{bmatrix}, \quad W^{k+1} = \begin{bmatrix} W^k \\ w(k+1) \end{bmatrix}$$

We can show that

$$J(k+1) = J(k) + \left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}\right]^T R^{-1}(k+1)\left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}\right]$$

The key to the recursion is to rewrite $J(k)$ as

$$J(k) = \left[\boldsymbol{x} - \hat{\boldsymbol{x}}(k, Z^k)\right]^T \hat{P}^{-1}(k, Z^k)\left[\boldsymbol{x} - \hat{\boldsymbol{x}}(k, Z^k)\right] + \underbrace{\left[Z^k - H^k\hat{\boldsymbol{x}}(k, Z^k)\right]^T \left(R^k\right)^{-1}\left[Z^k - H^k\hat{\boldsymbol{x}}(k, Z^k)\right]}_{\text{irreducible constant (independent of } \boldsymbol{x})}$$

From which it follows that

$$\begin{aligned} J(k+1) = &\left[\boldsymbol{x} - \hat{\boldsymbol{x}}(k, Z^k)\right]^T \hat{P}^{-1}(k, Z^k)\left[\boldsymbol{x} - \hat{\boldsymbol{x}}(k, Z^k)\right] \\ &+ \left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}\right]^T R^{-1}(k+1)\left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}\right] + \text{const.} \end{aligned}$$

We minimize J by the usual procedure:

$$\begin{aligned} 0 &= \left(\frac{\partial J}{\partial \boldsymbol{x}}\right)^T \\ &= 2\hat{P}^{-1}(k, Z^k)\left[\boldsymbol{x} - \hat{\boldsymbol{x}}(k, Z^k)\right] - 2H^T(k+1)R^{-1}(k+1)\left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}\right] \end{aligned}$$

Solving for $\boldsymbol{x}$ yields the minimizing value

$$\begin{aligned} \hat{\boldsymbol{x}}(k+1, Z^{k+1}) = &\left[\hat{P}^{-1}(k, Z^k) + H^T(k+1)R^{-1}(k+1)H(k+1)\right]^{-1} \\ &\times \left[\hat{P}^{-1}(k, Z^k)\hat{\boldsymbol{x}}(k, Z^k) + H^T(k+1)R^{-1}(k+1)\boldsymbol{z}(k+1)\right] \end{aligned}$$

With some manipulation, we can rewrite this as

$$\hat{\boldsymbol{x}}(k+1, Z^{k+1}) = \hat{\boldsymbol{x}}(k, Z^k) + W(k+1)\left[\boldsymbol{z}(k+1) - H(k+1)\hat{\boldsymbol{x}}(k, Z^k)\right]$$

This equation is in feedback correction, or predictor-corrector, form with gain matrix

$$W(k+1) \triangleq \left[\hat{P}^{-1}(k, Z^k) + H^T(k+1)R^{-1}(k+1)H(k+1)\right]^{-1} H^T(k+1)R^{-1}(k+1)$$

Analysis of Recursive LS Algorithm The estimation error

$$\begin{aligned}\tilde{\boldsymbol{x}}(k+1, Z^{k+1}) &\triangleq \boldsymbol{x} - \hat{\boldsymbol{x}}(k+1, Z^{k+1}) \\ &= [I - W(k+1)H(k+1)]\tilde{\boldsymbol{x}}(k, Z^k) - W(k+1)\boldsymbol{w}(k+1)\end{aligned}$$

can be analyzed as follows. Assume that $\mathbb{E}\left[\tilde{\boldsymbol{x}}(k, \boldsymbol{z}^k)\boldsymbol{w}^T(k+1)\right] = 0$. Then

$$\begin{aligned}\hat{P}(k+1, Z^{k+1}) &\triangleq \mathbb{E}\left[\tilde{\boldsymbol{x}}(k+1, Z^{k+1})\tilde{\boldsymbol{x}}^T(k+1, Z^{k+1})\right] \\ &= [I - W(k+1)H(k+1)]\,\hat{P}(k, Z^k)\,[I - W(k+1)H(k+1)]^T \\ &\quad + W(k+1)R(k+1)W^T(k+1)\end{aligned}$$

We can find alternative formulas for $\hat{P}(k+1, Z^{k+1})$ and $W(k+1)$ using the matrix inversion lemma:

$$\begin{aligned}\hat{P}(k+1, Z^{k+1}) &= \left[\hat{P}^{-1}(k, Z^k) + H^T(k+1)R^{-1}(k+1)H(k+1)\right]^{-1} \\ &= [I - W(k+1)H(k+1)]\,\hat{P}(k, Z^k)\end{aligned}$$

$$\begin{aligned}W(k+1) &= \hat{P}(k+1, Z^{k+1})H^T(k+1)R^{-1}(k+1) \\ &= \hat{P}(k, Z^k)H^T(k+1)\left[\underbrace{H(k+1)\hat{P}(k, Z^k)H^T(k+1) + R(k+1)}_{S(k+1)}\right]^{-1}\end{aligned}$$

Example. Recall that, for our running example,

$$z(j) = \gamma_j x + w(j), \quad w(j) \sim \mathcal{N}(0, \sigma^2), \quad j = 1, \ldots, k$$

As shown before, the ML estimate is

$$\begin{aligned}\hat{x}(k, Z^k) &= \frac{\sum_{j=1}^k \gamma_j z(j)}{\sum_{j=1}^k \gamma_j^2} \\ \hat{P}(k, Z^k) &= \sigma^2_{\text{LS}} \\ &= \frac{\sigma^2}{\sum_{j=1}^k \gamma_j^2}\end{aligned}$$

Casting this problem in our new notation, we have

$$\begin{aligned}H(k+1) &= \gamma_{k+1} \\ R(k+1) &= \sigma^2\end{aligned}$$

It follows that

$$\begin{aligned}W(k+1) &= \left(\frac{\sigma^2}{\sum_{j=1}^k \gamma_j^2}\right)\gamma_{k+1}\left[\frac{1}{\gamma_{k+1}\left(\frac{\sigma^2}{\sum_{j=1}^k \gamma_j^2}\right)\gamma_{k+1} + \sigma^2}\right] \\ &= \frac{\gamma_{k+1}}{\gamma_{k+1}^2 + \sum_{j=1}^k \gamma_j^2} \\ &= \frac{\gamma_{k+1}}{\sum_{j=1}^{k+1} \gamma_j^2}.\end{aligned}$$

Hence,

$$\hat{\boldsymbol{x}}(k+1, Z^{k+1}) = \left(\frac{\sum_{j=1}^{k} \gamma_j z(j)}{\sum_{j=1}^{k} \gamma_j^2} \right) + \left(\frac{\gamma_{k+1}}{\sum_{j=1}^{k+1} \gamma_j^2} \right) \left[z(k+1) - \gamma_{k+1} \underbrace{\left(\frac{\sum_{j=1}^{k} \gamma_j z(j)}{\sum_{j=1}^{k} \gamma_j^2} \right)}_{\hat{x}(k, Z^k)} \right]$$

which we can simplify as

$$\hat{\boldsymbol{x}}(k+1, Z^{k+1}) = \frac{\sum_{j=1}^{k+1} \gamma_j z(j)}{\sum_{j=1}^{k+1} \gamma_j^2}$$

This is the same form as the ML estimate applied to a batch of $k+1$ measurements. Thus, with the recursive approach we get the same result without having to start from scratch.

6.5 Recursive Square-Root Least Squares Estimation

The square-root approach to least squares estimation also has a simple recursion. Assume we have normalized all our data equations so that

$$Z^{k+1} = H^{k+1}\boldsymbol{x} + W^{k+1}, \qquad W^{k+1} \sim \mathcal{N}(0, I)$$

The full cost function after having incorporated measurement $k+1$ is

$$\begin{aligned} J(k+1) &= \|H^{k+1}\boldsymbol{x} - Z^{k+1}\|^2 \\ &= \left\| \begin{bmatrix} H^k \\ H(k+1) \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} Z^k \\ \boldsymbol{z}(k+1) \end{bmatrix} \right\|^2 \\ &= \|H^k\boldsymbol{x} - Z^k\|^2 + \|H(k+1)\boldsymbol{x} - \boldsymbol{z}(k+1)\|^2 \\ &= \|\boldsymbol{\epsilon}(k)\|^2 + \|\tilde{R}_o(k)\boldsymbol{x} - \tilde{\boldsymbol{z}}_o\|^2 + \|H(k+1)\boldsymbol{x} - \boldsymbol{z}(k+1)\|^2 \\ &= \|\boldsymbol{\epsilon}(k)\|^2 + \left\| \begin{bmatrix} \tilde{R}_o(k) \\ H(k+1) \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \tilde{\boldsymbol{z}}_o(k) \\ \boldsymbol{z}(k+1) \end{bmatrix} \right\|^2 \end{aligned}$$

It's clear from this form that to process measurement $k+1$ we don't have to start from scratch—we can begin with the $\tilde{R}_o(k)$ and $\tilde{\boldsymbol{z}}_o$ left over from the last iteration. Note also that the form of $J(k+1)$ suggests an interpretation of $\tilde{\boldsymbol{z}}_o$ as a measurement with data equation

$$\tilde{\boldsymbol{z}}_o(k) = \tilde{R}_o(k)\boldsymbol{x} + \tilde{\boldsymbol{w}}_o(k), \qquad \tilde{\boldsymbol{w}}_o(k) \sim \mathcal{N}(0, I)$$

With this data equation, we can solve for $\hat{\boldsymbol{x}}(k, Z^k)$ as

$$\hat{\boldsymbol{x}}(k, Z^k) = \tilde{R}_o^{-1}\tilde{\boldsymbol{z}}_o$$

and evaluate the precision of our solution in terms of the error covariance matrix as

$$\hat{P}(k, Z^k) = (\tilde{R}_o^T \tilde{R}_o)^{-1}$$

In other words, this data equation encapsulates everything we know about the estimate $\hat{\boldsymbol{x}}(k, Z^k)$. Interpreting this data equation as just another measurement makes the square-root form intuitive: we just stack our new measurement with our old measurement.

Now we QR factorize

$$\tilde{Q}\tilde{R} = \begin{bmatrix} \tilde{R}_o(k) \\ H(k+1) \end{bmatrix}$$

and transform via $T = \tilde{Q}^T$ to get

$$\begin{aligned} J(k+1) &= \|\boldsymbol{\epsilon}(k)\|^2 + \left\| \begin{bmatrix} \tilde{R}_o(k+1) \\ 0 \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \tilde{\boldsymbol{z}}_o(k+1) \\ \boldsymbol{\epsilon}(k+1) \end{bmatrix} \right\|^2 \\ &= \|\boldsymbol{\epsilon}(k)\|^2 + \|\boldsymbol{\epsilon}(k+1)\|^2 + \|\tilde{R}_o(k+1)\boldsymbol{x} - \tilde{\boldsymbol{z}}_o(k+1)\|^2 \end{aligned}$$

The minimizing value of $\boldsymbol{x}$ is now obvious:

$$\hat{\boldsymbol{x}}(k+1, Z^{k+1}) = \tilde{R}_o^{-1}(k+1)\tilde{\boldsymbol{z}}_o(k+1)$$

and our new estimate's error covariance is

$$\begin{aligned} \hat{P}(k+1, Z^{k+1}) = \left[(H^{k+1})^T H^{k+1}\right]^{-1} &= \left[\tilde{R}_o^T(k+1)\tilde{R}_o(k+1)\right]^{-1} \\ &= \tilde{R}_o^{-1}(k+1)\tilde{R}_o^{-T}(k+1) \end{aligned}$$

Chapter 7

Nonlinear Least Squares Estimation

Scribe: Chao Jia

7.1 Basics of nonlinear least squares estimation

Problem model:

$$\boldsymbol{z} = h(\boldsymbol{x} + \boldsymbol{w}) \tag{7.1}$$

where $\boldsymbol{w} \sim \mathcal{N}(0, R)$, $\boldsymbol{w} \in \mathbb{R}^{n_z \times 1}$ and $\boldsymbol{x} \in \mathbb{R}^{n_x \times 1}$. Normally $n_z > n_x$. $h(\boldsymbol{x})$ is an n_x-by-1 vector function of $\boldsymbol{x}$. We can write

$$h(\boldsymbol{x}) = \begin{bmatrix} h_1(\boldsymbol{x}) \\ h_2(\boldsymbol{x}) \\ \vdots \\ h_{n_z}(\boldsymbol{x}) \end{bmatrix}. \tag{7.2}$$

Problem statement: find $\boldsymbol{x}$ to minimize the objective function $J_{NLW}(\boldsymbol{x}$ (nonlinear and weighted).

$$J_{NLW}(\boldsymbol{x}) = [\boldsymbol{z} - h(\boldsymbol{x})]^T R^{-1} [\boldsymbol{z} - h(\boldsymbol{x})]. \tag{7.3}$$

If we can find a unique $\boldsymbol{x}$ that minimizes $J_{NLW}(\boldsymbol{x})$, then this becomes $\hat{\boldsymbol{x}}$, our nonlinear LS estimate.

Note that in the linear case $h(\boldsymbol{x}) = H\boldsymbol{x}$, also note that $p(\boldsymbol{z}|\boldsymbol{x}) = C \cdot \exp\left[-\frac{1}{2} J_{NLW}(\boldsymbol{x})\right]$ (because the noise is additive). Minimizing $J_{NLW}(\boldsymbol{x})$ is equivalent to maximizing the likelihood function. So $\hat{\boldsymbol{x}}$ is also the ML estimate. It does not matter that $h(\boldsymbol{x})$ is nonlinear. Hense, $J_{NLW}(\boldsymbol{x})$ has a rigorous statistical meaning.

Properties of $J_{NLW}(\boldsymbol{x})$:

1. $J_{NLW}(\boldsymbol{x}) \geq 0$ (assumes $R > 0$)
2. $J_{NLW}(\boldsymbol{x}) = 0 \Leftrightarrow h(\boldsymbol{x}) = \boldsymbol{z}$

Use Cholesky factorization to simplify the form of $J_{NLW}(\boldsymbol{x})$: $R_a^T R_a = R$. Then let $\boldsymbol{z}_a = R_a^{-T} \boldsymbol{z}$, $h_a(\boldsymbol{x}) = R_a^{-T} h(\boldsymbol{x})$, and $\boldsymbol{w}_a = R_a^{-T} \boldsymbol{w} \sim \mathcal{N}(0, I)$. Then we have

$$J_{NLW}(\boldsymbol{x}) = (\boldsymbol{z}_a - h_a(\boldsymbol{x}))^T (\boldsymbol{z}_a - h_a(\boldsymbol{x})) \tag{7.4}$$

$$= ||\boldsymbol{z}_a - h_a(\boldsymbol{x})||^2 \tag{7.5}$$

Drop the a's and consider $J_{NLW}(\boldsymbol{x}) = ||\boldsymbol{z}_a - h_a(\boldsymbol{x})||^2$. This will be our generic problem formulation now.

Aside:

The gradient operator is $\nabla_{\boldsymbol{x}} = (\frac{\partial}{\partial \boldsymbol{x}})^T = \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{x}_1} \\ \frac{\partial}{\partial \boldsymbol{x}_2} \\ \vdots \\ \frac{\partial}{\partial \boldsymbol{x}_n} \end{bmatrix}$

$$\nabla_{\boldsymbol{x}} f^T(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{x}_1} \\ \frac{\partial}{\partial \boldsymbol{x}_2} \\ \vdots \\ \frac{\partial}{\partial \boldsymbol{x}_n} \end{bmatrix} [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})] = \begin{bmatrix} \frac{f_1}{\boldsymbol{x}_1} & \frac{f_2}{\boldsymbol{x}_1} & \cdots & \frac{f_m}{\boldsymbol{x}_1} \\ \frac{f_1}{\boldsymbol{x}_2} & \frac{f_2}{\boldsymbol{x}_2} & \cdots & \frac{f_m}{\boldsymbol{x}_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{f_1}{\boldsymbol{x}_n} & \frac{f_2}{\boldsymbol{x}_n} & \cdots & \frac{f_m}{\boldsymbol{x}_n} \end{bmatrix} \in \mathbb{R}^{n \times m}$$

The Jacobian is the transpose of this:
$\frac{\partial f}{\partial \boldsymbol{x}} = [\nabla_{\boldsymbol{x}} f^T(\boldsymbol{x})]^T$

Define $\frac{\partial h}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_{nom}} = H(\boldsymbol{x}_{nom}) = H$. Thus, $H_{ij} = \frac{\partial h_i}{\partial \boldsymbol{x}_j}\Big|_{\boldsymbol{x}_{nom}}$.

First order necessary condition for the minimum J_{NL} is:

$$0 = (\frac{\partial J}{\partial \boldsymbol{x}}\Big|_{\hat{\boldsymbol{x}}})^T = \begin{bmatrix} \frac{\partial J}{\partial \boldsymbol{x}_1} \\ \vdots \\ \frac{\partial J}{\partial \boldsymbol{x}_n} \end{bmatrix}\Big|_{\hat{\boldsymbol{x}}} \tag{7.6}$$

We know that

$$\frac{\partial J}{\partial \boldsymbol{x}}\Big|_{\hat{\boldsymbol{x}}} = 2(\boldsymbol{z} - h(\hat{\boldsymbol{x}})^T \ (-\frac{\partial h}{\partial \boldsymbol{x}})\Big|_{\hat{\boldsymbol{x}}} = -2(\boldsymbol{z} - h(\hat{\boldsymbol{x}})^T H. \tag{7.7}$$

So we need

$$0 = (\frac{\partial J}{\partial \boldsymbol{x}}\Big|_{\hat{\boldsymbol{x}}})^T = -2H^T(\boldsymbol{z} - h(\hat{\boldsymbol{x}}) \tag{7.8}$$

Our goal is to solve this equation for $\boldsymbol{x}$.

Note that if $h(\boldsymbol{x})$ were linear, then from $0 = -2H^T(\boldsymbol{z} - h(\hat{\boldsymbol{x}}))$ we have $\hat{\boldsymbol{x}} = (H^T H)^{-1} H^T \boldsymbol{z}$.

7.2 Newton-Rhaphson method

To find $\hat{\boldsymbol{x}}$ which satisfies the first order necessary condition for minimum J_{NL}, firstly we will consider the Newton-Rhaphson (NR) method.

The NR method is originally used for finding zeros of a nonlinear function.

As shown in the above figure, to find the zero of a nonlinear function $f(x)$, we start from an initial guess x_1 and update it as x_2, where $x_2 = x_1 - \triangle x$.

In Newton-Rhaphson method $\triangle x = \frac{f(x_1)}{f'(x_1)}$. The NR method comes from first order Taylor expansion:

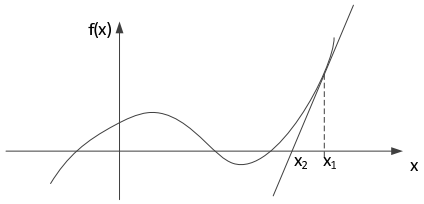


Figure 7.1: Newton-Rhaphson method

$$0 = f(x) = f(x + \triangle x) = f(x_1) + f'(x_1)\triangle x + H.O.T. \tag{7.9}$$
$$\Downarrow$$
$$\triangle x \cong -\frac{f(x_1)}{f'(x_1)}$$

In the current context (minimizing $J_{NL}(\boldsymbol{x})$, suppose we have a guess for $\hat{\boldsymbol{x}}$ and call it $\hat{\boldsymbol{x}}_g$.

Define $\triangle\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_g$, then we have $\hat{\boldsymbol{x}} = \triangle\hat{\boldsymbol{x}} + \hat{\boldsymbol{x}}_g$.

Based on the first order necessary condition we need

$$0 = -H^T(\hat{\boldsymbol{x}}_g + \triangle\hat{\boldsymbol{x}})[\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g + \triangle\hat{\boldsymbol{x}})] \tag{7.10}$$

The vector Taylor series is defined as

$$f(\hat{\boldsymbol{x}}) = f(\hat{\boldsymbol{x}}_g + \triangle\hat{\boldsymbol{x}}) \tag{7.11}$$
$$= f(\hat{\boldsymbol{x}}_g) + [\frac{\partial f}{\partial \boldsymbol{x}}\Big|_{\hat{\boldsymbol{x}}_g}]\triangle\hat{\boldsymbol{x}} + O(\triangle\hat{\boldsymbol{x}}^2) \tag{7.12}$$

Let $f(\boldsymbol{x}) = -H^T(\boldsymbol{x})[\boldsymbol{z} - h(\boldsymbol{x})]$, then based on NR method we need to solve

$$0 = f(\hat{\boldsymbol{x}}_g) + [\frac{\partial f}{\partial \boldsymbol{x}}\Big|_{\hat{\boldsymbol{x}}_g}]\triangle\hat{\boldsymbol{x}}$$
$$\Downarrow$$
$$\frac{\partial f}{\partial \boldsymbol{x}} = -\frac{\partial H^T}{\partial \boldsymbol{x}}[\boldsymbol{z} - h(\boldsymbol{x})] - H^T(-\frac{\partial h}{\partial \boldsymbol{x}})$$
$$= -\frac{\partial^2 h}{\partial \boldsymbol{x}^2}[\boldsymbol{z} - h(\boldsymbol{x})] + H^T H = V$$

where $\frac{\partial^2 h}{\partial \boldsymbol{x}^2}$ is beyond a matrix, it is actually a tensor of three indices. $\frac{\partial f}{\partial \boldsymbol{x}} = V$ is symmetric. Each entry in $\frac{\partial f}{\partial \boldsymbol{x}}$ can be written as

$$[\frac{\partial f}{\partial \boldsymbol{x}}]_{ij} = -\sum_{l=1}^{n_z} \frac{\partial^2 h_l}{\partial \boldsymbol{x}_i \partial \boldsymbol{x}_j}\Big|_{\hat{\boldsymbol{x}}_g} [\boldsymbol{z}_l - h_l(\hat{\boldsymbol{x}}_g)] + (H^T H)_{ij} \tag{7.13}$$

NR method says: solve $0 = -H^T(\hat{\boldsymbol{x}}_g)(\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g)) + V\triangle\hat{\boldsymbol{x}})$. So we have $\triangle\hat{\boldsymbol{x}} = V^{-1}H^T(\hat{\boldsymbol{x}}_g)$. In for NR method just let $\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}}_g + \triangle\hat{\boldsymbol{x}}$, then repeat this until convergence. (i.e., $\triangle\hat{\boldsymbol{x}} \to 0$)

If $\hat{\boldsymbol{x}}_g$ starts "sufficiently close" to the optimum, then $\hat{\boldsymbol{x}}_g$ converges super linearly to the optimum under normal conditions ($h(\boldsymbol{x})$ satisfies several smoothness requirements).

7.3 Gauss-Newton Algorithm (Gill et al. 4.7.2) (with step length algorithm)

Problems with NR method:

1. painful to compute $\frac{\partial H}{\partial \boldsymbol{x}} = \frac{\partial^2 h}{\partial \boldsymbol{x}^2}$

2. NR can diverge if $\hat{\boldsymbol{x}}_g$ is "too far" from solution

If $\hat{\boldsymbol{x}}_{soln}$ produces a small residual error, then $||\boldsymbol{z}-h(\hat{\boldsymbol{x}}_{soln})||$ is small, and it is reasonable to neglect the second order term in V, i.e., let $V = H^T H$.

Therefore, $\triangle\hat{\boldsymbol{x}} = V^{-1} f(\hat{\boldsymbol{x}}_g) = (H^T H) H^T [\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g)]$.

One can arrive at this expression via another straightforward route:

$$J(\hat{\boldsymbol{x}}) = ||\boldsymbol{z} - h(\hat{\boldsymbol{x}})||^2 \cong ||\underbrace{\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g)}_{\triangle \boldsymbol{z}} - H\triangle\hat{\boldsymbol{x}}||^2 \tag{7.14}$$

Let $J(\triangle\hat{\boldsymbol{x}}) = ||\triangle\boldsymbol{z} - H\triangle\hat{\boldsymbol{x}}||^2$. The $\triangle\hat{\boldsymbol{x}}$ that minimize this cost function is the same as $\triangle\hat{\boldsymbol{x}}$ given in (7.14).

Comparison:

NR method applies a Taylor series expansion to the first order necessary conditions. On the other hand, the GN method applies Taylor series to the measurement model: $\boldsymbol{z} = h(\hat{\boldsymbol{x}}_g) + H\triangle\hat{\boldsymbol{x}} + \boldsymbol{w}$.

To avoid divergence, we modify the updating equation in GN method as $\hat{\boldsymbol{x}}_g \leftarrow \hat{\boldsymbol{x}}_g + \alpha\hat{\boldsymbol{x}}$, where $0 < \alpha \leq 1$. Choose α s.t. $J[\hat{\boldsymbol{x}}_g + \alpha\triangle\hat{\boldsymbol{x}}]$ is less than $J[\hat{\boldsymbol{x}}_g]$. This guarantees convergence in virtually all situations because of condition $J(\boldsymbol{x}) \geq 0$.

Q: How do we know $\exists\alpha$: $0 < \alpha \leq 1$ s.t. $J(\hat{\boldsymbol{x}}_g + \alpha\triangle\hat{\boldsymbol{x}}) < J(\hat{\boldsymbol{x}}_g)$?
A: Define $\tilde{J}(\alpha) = J(\hat{\boldsymbol{x}}_g + \alpha\triangle\hat{\boldsymbol{x}})$, call α the step length.

Note: $\tilde{J}(0) = J(\hat{\boldsymbol{x}}_g)$ and $\frac{d\tilde{J}}{d\alpha} = (\frac{\partial J}{\partial \boldsymbol{x}}\big|_{\hat{\boldsymbol{x}}_g + \triangle\hat{\boldsymbol{x}}})\triangle\hat{\boldsymbol{x}}$. (chain rule)

Consider

$$\left.\frac{d\tilde{J}}{d\alpha}\right|_{\alpha=0} = \left.\frac{\partial J}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_g} \triangle\hat{\boldsymbol{x}} \tag{7.15}$$

$$= -2[\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g)]^T H(H^T H)^{-1} H^T (\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g)) \tag{7.16}$$

$$= -2(H^T(\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g)))^T (H^T H)^{-1} (H^T(\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g))) \tag{7.17}$$

This is a quadratic form. In cases where the nonlinear system is observable, $H^T H > 0$, which implies $rank(H) = n_x$ (all columns are linearly independent) and $\left.\frac{d\tilde{J}}{d\alpha}\right|_{\alpha=0} < 0$, with equality only if $\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g) = 0$.

Thus for some small values of α, $\tilde{J}(\alpha) < J(\hat{\boldsymbol{x}}_g)$ is guaranteed! ■

Practical but crude approach for GN algorithm:

1. Set $\alpha = 1$.
2. $J_g = \tilde{J}(\alpha = 0)$, $J_{g_{new}} = \tilde{J}(\alpha = 1)$
3. while $J_{g_{new}} \geq J_g$
   $\alpha = \alpha/2$
   $J_{g_{new}} = \tilde{J}(\alpha)$
   end

This will converge to a local minimum.

7.4 Levenberg-Marquart Method (LM)

In each updating step of LM method, we have $\hat{\boldsymbol{x}}_g \leftarrow \hat{\boldsymbol{x}}_g + \triangle\hat{\boldsymbol{x}}_{LM}$, where

$$\triangle\hat{\boldsymbol{x}}_{LM} = (H^T H + \lambda I)^{-1} H^T [\boldsymbol{z} - h(\hat{\boldsymbol{x}}_g)] \tag{7.18}$$

with $\lambda \geq 0$

If $\lambda = 0$ $\triangle\hat{\boldsymbol{x}}_L M$ is equivalent to $\triangle\hat{\boldsymbol{x}}$ in Gauss-Newton method with $\alpha = 1$.

The LM method does not use step size parameter α, instead it uses λ.

LM	GN
$\lambda = 0$	$\alpha = 1$
$\lambda = \infty$	$\alpha = 0$

Pseudo LM algorithm:

1. $\lambda = 0$
2. check if $H^T H > 0$, if not let $\lambda =$ something small, say $\lambda = ||H|| \cdot 0.001$
3. compute $\triangle\hat{\boldsymbol{x}}_{LM}(\lambda)$
4. measure the cost $J_g = J(\hat{\boldsymbol{x}}_g)$, $J_g^{new} = J(\hat{\boldsymbol{x}}_g + \triangle\hat{\boldsymbol{x}}_{LM}(\lambda))$
5. If $J_g^{new} \geq J_g$, then let $\lambda = \max(2\lambda, ||H|| \cdot 0.001)$, go to step (3)
   Else we accept the new guess

LM achieves fast convergence near the solution if the residuals are small. If the residuals near solution are not small, we may have to use full NR (Newton Rhaphson) method.

Chapter 8

Stochastic Linear System Models

8.1 Continuous-time model for dynamic systems

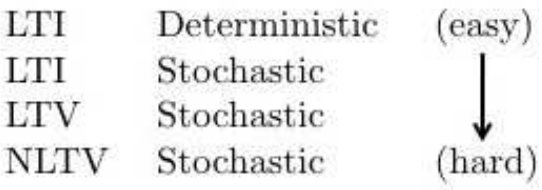


Figure 8.1: We will be dealing with NLTV stochastic systems.

A continuous-time model for a dynamic system is given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u} + \boldsymbol{D}(t)\tilde{\boldsymbol{v}}(t) \tag{8.1}$$

where $\dot{\boldsymbol{x}}$ (the state vector) is $n_x \times 1$, $\boldsymbol{u}$ (the input vector) is $n_u \times 1$, $\tilde{\boldsymbol{v}}$ (the process noise or disturbance) is $n_v \times 1$, and the matrices $\boldsymbol{A}$ (the system matrix), $\boldsymbol{B}$ (the input gain), and $\boldsymbol{D}$ (the noise gain) are appropriately dimensioned. The measurement model is given by

$$\boldsymbol{z}(t) = \boldsymbol{C}(t)\boldsymbol{x}(t) + \tilde{\boldsymbol{w}}(t) \tag{8.2}$$

where $\tilde{\boldsymbol{w}}$ (the measurement noise) is $n_z \times 1$ and $\boldsymbol{C}$ (the measurement matrix) is $n_z \times n_x$.

Note: $\tilde{\boldsymbol{v}}$ is continuous but not differentiable, meaning that it cannot properly be put into a differential equation. However, a more rigorous derivation of the equation still leads to the same result.

Note: If $\tilde{\boldsymbol{v}}(t) = \tilde{\boldsymbol{w}}(t) = \boldsymbol{0}$, then, given $\boldsymbol{x}(t_0)$ and $\boldsymbol{u}(\tau)$ for $t_0 \leq \tau \leq t$, one can predict $\boldsymbol{x}$ and $\boldsymbol{z}$ for the entire time interval. When $\tilde{\boldsymbol{v}}(t)$ and $\tilde{\boldsymbol{w}}(t)$ are not equal to $\boldsymbol{0}$, it may be enough to know the pdf of $\boldsymbol{x}(t_0)$ in order to predict the conditional pdfs of all future $\boldsymbol{x}(t)$ values (conditioned on the data $\boldsymbol{z}(\sigma)$ for $\sigma \leq t_0$), but sometimes this is not the case.

The solution of the above system is

$$\boldsymbol{x}(t) = \boldsymbol{F}(t, t_0)\boldsymbol{x}(t_0) + \int_{t_0}^{t} \boldsymbol{F}(t, \tau)[\boldsymbol{B}(\tau)\boldsymbol{u}(\tau) + \boldsymbol{D}(\tau)\tilde{\boldsymbol{v}}(\tau)]\mathrm{d}\tau \tag{8.3}$$

where $\boldsymbol{F}$ is the state transition matrix, sometimes denoted $\boldsymbol{\Phi}(t, t_0)$.

Note: $\boldsymbol{F}$ is defined by its properties:

$$\frac{\partial}{\partial t}\boldsymbol{F}(t, t_0) = \boldsymbol{A}(t)\boldsymbol{F}(t, t_0) \tag{8.4}$$

$$\boldsymbol{F}(t_0, t_0) = \boldsymbol{I} \tag{8.5}$$

where $\boldsymbol{I}$ is the identity matrix. Other properties of $\boldsymbol{F}$ include

- $\boldsymbol{F}(t, \tau) = \boldsymbol{F}(t, \sigma)\boldsymbol{F}(\sigma, \tau)$
- $\boldsymbol{F}(t, \tau) = [\boldsymbol{F}(\tau, t)]^{-1}$
- If $\boldsymbol{A}$ is constant, then $\boldsymbol{F}(t, \tau) = \boldsymbol{F}(t - \tau, 0) = e^{\boldsymbol{A}\cdot(t-\tau)}$, where the matrix exponential is defined as

  $$e^{\boldsymbol{A}\cdot(t-\tau)} = \boldsymbol{I} + \boldsymbol{A}\cdot(t-\tau) + \frac{1}{2!}\boldsymbol{A}^2(t-\tau)^2 + \ldots \tag{8.6}$$

  (The matrix exponential may be calculated in MATLAB using the expm() function.)
- If $\boldsymbol{A}$ is time-varying, then one must numerically integrate the matrix initial value problem in order to determine $\boldsymbol{F}(t, \tau)$.

8.2 White noise for stochastic systems

$\tilde{\boldsymbol{v}}(t)$ is white noise if $\tilde{\boldsymbol{v}}(t)$ is stochastically independent of $\tilde{\boldsymbol{v}}(\tau)$ for all $t \neq \tau$ and $\mathbb{E}[\tilde{\boldsymbol{v}}(t)] = \boldsymbol{0}$. (The noise must be independent even when t and τ are very close.) A consequence of whiteness is that $\mathbb{E}\left[\tilde{\boldsymbol{v}}(t)\tilde{\boldsymbol{v}}^T(\tau)\right] = \boldsymbol{V}(t)\delta(t-\tau)$, which for $\boldsymbol{V}(t) = \boldsymbol{V} = const$ implies that $\boldsymbol{S}_{vv}(f) =$ power spectral density of $\tilde{\boldsymbol{v}}(t) = \boldsymbol{v}$, i.e. the power spectrum is flat. This implies that white noise is

- independent in time
- zero mean
- has covariance $\boldsymbol{V}(t)\delta(t-\tau)$

This also implies a process that has infinite power because, at $t = \tau$, $\delta(t-\tau) = \infty$. However, we "just go with" the fiction of white noise because it is convenient and can be a good approximation over a frequency band (as opposed to the entire frequency spectrum).

8.3 Prediction of mean and covariance

The prediction of the mean is

$$\mathbb{E}[\boldsymbol{x}(t)] = \boldsymbol{F}(t, t_0)\mathbb{E}[\boldsymbol{x}(t_0)] + \int_{t_0}^{t} \boldsymbol{F}(t, \tau)[\boldsymbol{B}(\tau)\boldsymbol{u}(\tau)]\mathrm{d}\tau \tag{8.7}$$

$$\bar{\boldsymbol{x}}(t) = \boldsymbol{F}(t, t_0)\bar{\boldsymbol{x}}(t_0) + \int_{t_0}^{t} \boldsymbol{F}(t, \tau)\boldsymbol{B}(\tau)\boldsymbol{u}(\tau)\mathrm{d}\tau \tag{8.8}$$

Additionally,

$$\dot{\bar{\boldsymbol{x}}} = \boldsymbol{A}(t)\bar{\boldsymbol{x}}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) \tag{8.9}$$

meaning that the prediction of the mean follows the linear system. If $\mathbb{E}\left[\tilde{\boldsymbol{v}}(t)\right] = \bar{\boldsymbol{v}}$ (not zero-mean), then

$$\dot{\bar{\boldsymbol{x}}} = \boldsymbol{A}(t)\bar{\boldsymbol{x}}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) + \boldsymbol{D}(t)\bar{\boldsymbol{v}} \tag{8.10}$$

which is still deterministic.

The covariance is

$$\boldsymbol{P}_{xx}(t) = \mathbb{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^T\right] \tag{8.11}$$

stituting for $\bar{\boldsymbol{x}}$ gives

$$\boldsymbol{P}_{xx}(t) = \mathbb{E}\left[(\boldsymbol{F}(t,t_0)[\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}(t_0)] + \int_{t_0}^{t} \boldsymbol{F}(t,\tau_1)\boldsymbol{D}(\tau_1)\tilde{\boldsymbol{v}}(\tau_1)\mathrm{d}\tau_1)(\boldsymbol{F}(t,t_0)[\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}(t_0)] + \int_{t_0}^{t} \boldsymbol{F}(t,\tau_2)\boldsymbol{D}(\tau_2)\tilde{\boldsymbol{v}}(\tau_2)\mathrm{d}\tau_2)^T\right] \tag{8.12}$$

Expanding gives

$$\boldsymbol{P}_{xx}(t) = \boldsymbol{F}(t,t_0)E[(\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}(t_0))(\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}(t_0))^T]\boldsymbol{F}^T(t,t_0) + \int_{t_0}^{t}\int_{t_0}^{t} \boldsymbol{F}(t,\tau_1)\boldsymbol{D}(\tau_1)\mathbb{E}\left[\tilde{\boldsymbol{v}}(\tau_1)\tilde{\boldsymbol{v}}^T(\tau_1)\right]\boldsymbol{D}^T(\tau_2)\boldsymbol{F}(t,\tau_2)\mathrm{d}\tau_1\mathrm{d}\tau_2 \tag{8.13}$$

where $\mathbb{E}\left[\tilde{\boldsymbol{v}}(\tau_1)\tilde{\boldsymbol{v}}^T(\tau_1)\right] = \boldsymbol{V}(\tau_1)\delta(\tau_1 - \tau_2)$. Also, cross terms in the covariance go to zero because $\mathbb{E}\left[(\boldsymbol{x}(t_0) - \bar{\boldsymbol{x}}(t_0))(\tilde{\boldsymbol{v}}^T)\right] = \boldsymbol{0}$ for all $\tau > t_0$ due to the whiteness of the noise. The sifting property of the Dirac delta allows us to collapse one integral:

$$\boldsymbol{P}_{xx}(t) = \boldsymbol{F}(t,t_0)\boldsymbol{P}_{xx}(t_0)\boldsymbol{F}^T(t,t_0) + \int_{t_0}^{t} \boldsymbol{F}(t,\tau_1)\boldsymbol{D}(\tau_1)\boldsymbol{V}(\tau_1)\boldsymbol{D}^T(\tau_1)\boldsymbol{F}^T(t,\tau_1)\mathrm{d}\tau_1 \tag{8.14}$$

$$\dot{\boldsymbol{P}}_{xx}(t) = \boldsymbol{A}(t)\boldsymbol{P}_{xx}(t) + \boldsymbol{P}_{xx}(t)\boldsymbol{A}^T(t) + \boldsymbol{D}(t)\boldsymbol{V}(t)\boldsymbol{D}^T(t) \tag{8.15}$$

Note: $\dot{\boldsymbol{P}}_{xx}(t)$ is symmetric and linear in $\boldsymbol{P}_{xx}(t)$.

Note: If $\boldsymbol{A}(t) = \boldsymbol{A} = const$ and if $Re[eig(\boldsymbol{A})] < 0\ \forall$ eigenvalues of $\boldsymbol{A}$ (i.e. the system is stable) and if $\boldsymbol{V}(t) = \boldsymbol{V} = const$, $\boldsymbol{D}(t) = \boldsymbol{D} = const$, and $\boldsymbol{V} > \boldsymbol{0}$, then $\boldsymbol{P}_{xx}(t)$ converges to a constant $\boldsymbol{P}_{xxss}$, the steady-state value. Thus,

$$\boldsymbol{0} = \boldsymbol{A}\boldsymbol{P}_{xxss} + \boldsymbol{P}_{xxss}\boldsymbol{A}^T + \boldsymbol{D}\boldsymbol{V}\boldsymbol{D}^T \tag{8.16}$$

is a linear matrix equation known as the continuous time Lyapunov equation. To solve this equation in MATLAB, use the lyap() function

$$\boldsymbol{P}_{xxss} = \text{lyap}(\boldsymbol{A}, \boldsymbol{D}\boldsymbol{V}\boldsymbol{D}^T) \tag{8.17}$$

8.4 Discrete-time models of stochastic systems

Assume a zero-order hold control input: $\boldsymbol{u}(t) = \boldsymbol{u}(t_k) = \boldsymbol{u}_k$ for $t_k \leq t \leq t_{k+1}$.

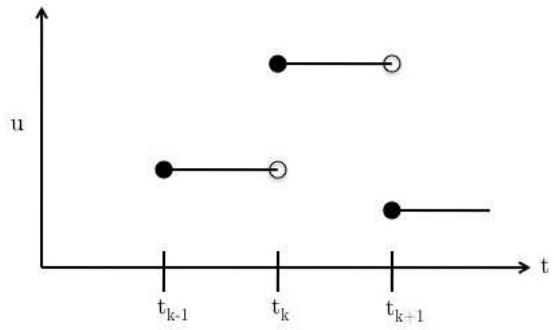


Figure 8.2: A zero-order hold control input holds a constant value for $t \in [t_k, t_{k+1})$.

Then an equivalent discrete-time model of our original continuous system is:

$$\boldsymbol{x}(t_{k+1}) = \boldsymbol{F}(t_{k+1}, t_k)\boldsymbol{x}(t_k) + \boldsymbol{G}(t_{k+1}, t_k)\boldsymbol{u}(t_k) + \boldsymbol{v}(t_k) \tag{8.18}$$

where

$$\boldsymbol{G}(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \boldsymbol{F}(t_{k+1}, \tau)\boldsymbol{B}(\tau)\mathrm{d}\tau \tag{8.19}$$

$$\boldsymbol{v}(t_k) = \int_{t_k}^{t_{k+1}} \boldsymbol{F}(t_{k+1}, \tau)\boldsymbol{D}(\tau)\tilde{\boldsymbol{v}}(\tau)\mathrm{d}\tau \tag{8.20}$$

$\boldsymbol{v}(t_k)$ is the discrete-time process noise disturbance. If $\tilde{\boldsymbol{v}}(t)$ is white noise, then $\mathbb{E}[\boldsymbol{v}(t_k)] = \boldsymbol{0}$ and

$$\mathbb{E}\left[\boldsymbol{v}(t_k)\boldsymbol{v}^T(t_j)\right] = \int_{t_j}^{t_{j+1}} \int_{t_k}^{t_{k+1}} \boldsymbol{F}(t_{k+1}, \tau_1)\boldsymbol{D}(\tau_1)\mathbb{E}\left[\tilde{\boldsymbol{v}}(\tau_1)\tilde{\boldsymbol{v}}^T(\tau_2)\right]\boldsymbol{D}^T(\tau_2)\boldsymbol{F}^T(t_{j+1}, \tau_2)\mathrm{d}\tau_2\mathrm{d}\tau_2 \tag{8.21}$$

where $\mathbb{E}\left[\tilde{\boldsymbol{v}}(\tau_1)\tilde{\boldsymbol{v}}^T(\tau_2)\right] = \boldsymbol{V}(\tau_1)\delta(\tau_1 - \tau_2)$. Thus,

$$\mathbb{E}\left[\boldsymbol{v}(t_k)\boldsymbol{v}^T(t_j)\right] = \delta_{jk}\int_{t_k}^{t_{k+1}} \boldsymbol{F}(t_{k+1}, \tau_1)\boldsymbol{D}(\tau_1)\boldsymbol{V}(\tau_1)\boldsymbol{D}^T(\tau_1)\boldsymbol{F}^T(t_{k+1}, \tau_1)\mathrm{d}\tau_1 \tag{8.22}$$

$$= \delta_{jk}\boldsymbol{Q}_k \tag{8.23}$$

(The Kronecker delta δ_{jk} appears because, if the t windows do not overlap, then the intervals are independent unless $j = k$.)

We may now simplify the notation:

$$\boldsymbol{x}(t_k) \to \boldsymbol{x}(k) \tag{8.24}$$
$$\boldsymbol{u}(t_k) \to \boldsymbol{u}(k) \tag{8.25}$$
$$\boldsymbol{v}(t_k) \to \boldsymbol{v}(k) \tag{8.26}$$
$$\boldsymbol{F}(t_{k+1}, t_k) \to \boldsymbol{F}(k) \tag{8.27}$$
$$\boldsymbol{G}(t_{k+1}, t_k) \to \boldsymbol{G}(k) \tag{8.28}$$
$$\boldsymbol{V}(t_{k+1}, t_k) \to \boldsymbol{V}(k) \tag{8.29}$$

The dynamics model then becomes:

$$\boldsymbol{x}(k+1) = \boldsymbol{F}(k)\boldsymbol{x}(k) + \boldsymbol{G}(k)\boldsymbol{u}(k) + \boldsymbol{v}(k) \tag{8.30}$$

where $\mathbb{E}[\boldsymbol{v}(k)] = \boldsymbol{0}$ and $\mathbb{E}\left[\boldsymbol{v}(k)\boldsymbol{v}^T(j)\right] = \delta_{kj}\boldsymbol{Q}_k$. For a time-invariant system ($\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{D}, \boldsymbol{V}$ constant), and if $t_{k+1} - t_k = \Delta t = const$, then

$$\boldsymbol{F}(k) = \boldsymbol{F} = e^{\boldsymbol{A}\Delta t} \tag{8.31}$$
$$\boldsymbol{G}(k) = \boldsymbol{G} = \int_0^{\Delta t} e^{\boldsymbol{A}\tau}\boldsymbol{B}\mathrm{d}\tau \tag{8.32}$$
$$\boldsymbol{Q}(k) = \int_0^{\Delta t} e^{\boldsymbol{A}\tau}\boldsymbol{D}\boldsymbol{V}\boldsymbol{D}^T e^{\boldsymbol{A}^T\tau}\mathrm{d}\tau \tag{8.33}$$

8.5 Discrete-time measurement model

The discrete-time measurement model is

$$\boldsymbol{z}(k) = \boldsymbol{H}(k)\boldsymbol{x}(k) + \boldsymbol{w}(k) \tag{8.34}$$

where $\boldsymbol{w}(k)$ is discrete-time white measurement noise. This implies

$$\mathbb{E}[\boldsymbol{w}(k)] = \boldsymbol{0} \tag{8.35}$$
$$\mathbb{E}\left[\boldsymbol{w}(k)\boldsymbol{w}^T(j)\right] = \delta_{kj}\boldsymbol{R}(k) \tag{8.36}$$

where $\boldsymbol{R}(k) = \boldsymbol{R}^T(k) > \boldsymbol{0}$. We think of $\boldsymbol{z}(k)$ as a sample from $\boldsymbol{z}(t) = \boldsymbol{C}(t)\boldsymbol{x}(t) + \tilde{\boldsymbol{w}}(t)$, but it is not correct to say that $\boldsymbol{z}(k) = \boldsymbol{z}(t_k)$. The problem lies in the assumption of whiteness, and therefore infinite power, for $\tilde{\boldsymbol{w}}(t)$. Because of this, $\mathbb{E}\left[\tilde{\boldsymbol{w}}(t_k)\tilde{\boldsymbol{w}}^T(t_k)\right] = \delta(0)\boldsymbol{Q} = \infty$. The correct way to obtain $\boldsymbol{z}(k)$ is to assume an anti-aliasing filter is used to low-pass filter $\boldsymbol{z}(t)$ before sampling. This can be modeled as an average-and-sample operation:

By the Nyquist sampling theorem, we must sample at twice the bandwidth of the anti-aliasing filter ($f_{samp} = \frac{1}{\Delta t}$).

Now, $\boldsymbol{R}(k)$ becomes

$$\boldsymbol{R}(k) = \frac{1}{\delta t^2}\int_{t_k-\delta t}^{t_k}\int_{t_k-\delta t}^{t_k} E[\tilde{\boldsymbol{w}}(\tau_1)\tilde{\boldsymbol{w}}^T(\tau_2)]\mathrm{d}\tau_1\mathrm{d}\tau_2 \approx \frac{\boldsymbol{R}\tilde{\boldsymbol{w}}(t_k)}{\delta t} \tag{8.37}$$

where $\mathbb{E}\left[\tilde{\boldsymbol{w}}(\tau_1)\tilde{\boldsymbol{w}}^T(\tau_2)\right] = \boldsymbol{R}\tilde{\boldsymbol{w}}(\tau_1)\delta(\tau_1-\tau_2)$.

8.6 Full discrete-time model

Combining the discrete-time dynamics and measurement models, we obtain the full discrete-time model.

$$\boldsymbol{x}(k+1) = \boldsymbol{F}(k)\boldsymbol{x}(k) + \boldsymbol{G}(k)\boldsymbol{u}(k) + \boldsymbol{v}(k) \tag{8.38}$$
$$\boldsymbol{z}(k) = \boldsymbol{H}(k)\boldsymbol{x}(k) + \boldsymbol{w}(k) \tag{8.39}$$

with $\mathbb{E}[\boldsymbol{v}(k)] = \boldsymbol{0}$, $\mathbb{E}\left[\boldsymbol{v}(k)\boldsymbol{v}^T(j)\right] = \boldsymbol{Q}(k)\delta_{kj}$, $\mathbb{E}[\boldsymbol{w}(k)] = \boldsymbol{0}$, $\mathbb{E}\left[\boldsymbol{w}(k)\boldsymbol{w}^T(j)\right] = \boldsymbol{R}(k)\delta_{kj}$, and $\mathbb{E}\left[\boldsymbol{w}(k)\boldsymbol{v}^T(j)\right] = \boldsymbol{0}\ \forall k, j$.

We can solve the discrete-time dynamics equation.

$$\boldsymbol{x}(k) = [\boldsymbol{F}(k-1)\boldsymbol{F}(k-2)\cdots\boldsymbol{F}(0)]\boldsymbol{x}(0) + \sum_{i=0}^{k-1}\{[\boldsymbol{F}(k-1)\boldsymbol{F}(k-2)\cdots\boldsymbol{F}(i+1)][\boldsymbol{G}(i)\boldsymbol{u}(i)+\boldsymbol{v}(i)]\} \tag{8.40}$$

Note: When $i = k-1$, the identity matrix $\boldsymbol{I}$ should be used in place for $[\boldsymbol{F}(k-1)\boldsymbol{F}(k-2)\cdots\boldsymbol{F}(i+1)]$ in the summation.

We can also predict the statistics of $\boldsymbol{x}(k)$.

$$\bar{\boldsymbol{x}}(k+1) = \mathbb{E}[\boldsymbol{x}(k+1)] = \boldsymbol{F}(k)\bar{\boldsymbol{x}}(k) + \boldsymbol{G}(k)\boldsymbol{u}(k) \tag{8.41}$$
$$\boldsymbol{P}_{xx} = \mathbb{E}\left[(\boldsymbol{x}(k+1)-\bar{\boldsymbol{x}}(k+1))(\boldsymbol{x}(k+1)-\bar{\boldsymbol{x}}(k+1))^T\right] = \boldsymbol{F}(k)\boldsymbol{P}_{xx}\boldsymbol{F}^T k + \boldsymbol{Q}(k) \tag{8.42}$$

The simplification of $\boldsymbol{P}_{xx}$ exploits $\mathbb{E}\left[(\boldsymbol{x}(k)-\bar{\boldsymbol{x}}(k))\boldsymbol{v}^T(k)\right] = \boldsymbol{0}$ to cancel cross terms.

If $\boldsymbol{F}(k) = \boldsymbol{F} = const$, and if $\boldsymbol{Q}(k) = \boldsymbol{Q} = const$ (LTI system), and if $max(abs(eig(\boldsymbol{F}))) < 1$ (asymptotically stable discrete-time system), then

$$\boldsymbol{P}(k)_{xx} \to \boldsymbol{P}_{xxss} \tag{8.43}$$

where $\boldsymbol{P}_{xxss}$ is a steady-state covariance. We then have the discrete-time Lyapunov equation

$$\boldsymbol{P}_{xxss} = \boldsymbol{F}\boldsymbol{P}_{xxss}\boldsymbol{F}^T + \boldsymbol{Q} \tag{8.44}$$

In MATLAB:

$$\boldsymbol{P}_{xxss} = \text{dlyap}(\boldsymbol{F}, \boldsymbol{Q}) \tag{8.45}$$

Also, $\boldsymbol{Q} > \boldsymbol{0} \Rightarrow \boldsymbol{P}_{xxss} > \boldsymbol{0}$.

Q: Where does the requirement that $max(abs(eig(\boldsymbol{F}))) < 1$ for stability come from?
A: $\boldsymbol{F} = e^{\boldsymbol{A}\Delta t}$. One can show that $|\lambda| = e^{Re(\beta)\Delta t}$, where λ is the eigenvalue of $\boldsymbol{F}$ corresponding to β, an eigenvalue of $\boldsymbol{A}$. Iff $Re(\beta) < 0$, then $|\lambda| < 1$. ■

Chapter 9

Kalman filter for discrete-time linear systems

9.1 The Kalman Filter

1) Dynamics and measurement model

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k) \tag{9.1}$$
$$z(k) = H(k)x(k) + w(k) \tag{9.2}$$

$$\mathbb{E}\left[v(k)\right] = 0 \qquad \mathbb{E}\left[v(k)v(j)^T\right] = Q_k\delta_{jk} \tag{9.3}$$
$$\mathbb{E}\left[w(k)\right] = 0 \qquad \mathbb{E}\left[w(k)w(j)^T\right] = R_k\delta_{jk} \tag{9.4}$$

$$\mathbb{E}\left[w(k)v(j)^T\right] = 0 \ \forall k, j \tag{9.5}$$

Note: w and v uncorrelated is not required, but will be assumed here to simplify analysis.

2) A priori information about initial state

$$\mathbb{E}\left[x(0)|Z^0\right] = \hat{x}(0) \tag{9.6}$$
$$\mathbb{E}\left[(x(0) - \hat{x}(0))(x(0) - \hat{x}(0))^T|Z^0\right] = P(0) \tag{9.7}$$
$$\mathbb{E}\left[w(k)(x(0) - \hat{x}(0))^T\right] = 0 \ \forall k \geq 0 \tag{9.8}$$
$$\mathbb{E}\left[v(k)(x(0) - \hat{x}(0))^T\right] = 0 \ \forall k \geq 0 \tag{9.9}$$

3) $v(k)$, $w(k)$, $x(0)$ are all Gaussian random variables

Here we have chosen some specific conditions for setting up the Kalman filter. Later, we will relax some of these conditions, or investigate the implications of them being violated.

There are several generic estimation problems:

Filtering: Determine $\hat{x}(k|Z^k) = \hat{x}(k|z0, ..., z_k)$ - Use measurements up to time k, to estimate state at time k. This can be done in real time, and causally - it does not depend on future states.
Smoothing: Determine $\hat{x}(j|Z^k)$, for $j < k$ - Use future data to find an improved estimate of a historical state. This is noncausal.
Prediction: Determine $\hat{x}(j|Z^k)$, for $j > k$ - Estimation of a future state - this gives the worst estimate of the three types.

"Prediction is always hard, especially when it's about the future" - Groucho Marx

We will concentrate on the filtering problem for now.

There are several different standard notations for the filtering problem. Bar-Shalom's notation is unambiguous, but cumbersome, so we will use a leaner alternative.

Bar-shalom	Humphreys	others	name
$\hat{x}(k\|Z^k) = \hat{x}(k\|k)$	$\hat{x}(k)$	$\hat{x}^+(k)$	a posteriori state estimate
$\hat{x}(k+1\|Z^k) = \hat{x}(k+1\|k)$	$\bar{x}(k+1)$ †	$\hat{x}^-(k+1)$	a priori state estimate
$P(k\|Z^k) = P(k\|k)$	$P(k)$	$P^+(k)$	a posteriori state estimate error covariance
$P(k+1\|Z^k) = P(k+1\|k)$	$\bar{P}(k+1)$	$P^-(k+1)$	a priori state estimate error covariance

†: This notation is nice because it corresponds to the prior in the static estimation equations.

Filtering steps (derivation based on MMSE)
0) set $k = 0$, then $\hat{x}(k)$, $P(k)$ known

1) state and covariance propagation: predict state and error covariance at step $k+1$, conditioned on data through $z(k)$
state estimate propagation:

$$\bar{x}(k+1) = \mathbb{E}\left[x(k+1)|Z^k\right] \tag{9.10}$$
$$\bar{x}(k+1) = \mathbb{E}\left[F(k)x(k) + G(k)u(k) + v(k)|Z^k\right] \tag{9.11}$$
$$\bar{x}(k+1) = F(k)\mathbb{E}\left[x(k)|z^k\right] + G(k)u(k) + \cancel{\mathbb{E}\left[v(k)|Z^k\right]} \tag{9.12}$$
$$\bar{x}(k+1) = F(k)\hat{x}(k) + G(k)u(k) \tag{9.13}$$

state covariance propagation:

$$\bar{P}(k+1) = \mathbb{E}\left[(x(k+1) - \hat{x}(k+1))(x(k+1) - \hat{x}(k+1))^T|Z^k\right] \tag{9.14}$$
$$\bar{P}(k+1) = F(k)P(k)F^T(k) + Q(k) \tag{9.15}$$

Some additional steps for this can be found on page 204 of Bar-Shalom. The cross-terms are zero due to the fact that $v(k)$ is zero mean and white, and orthogonal to $\tilde{x}(k)$.
The first term tends to decrease (since the absolute value of the eigenvalues of F are less than one), but because of the additive Q term, the overall covariance grows (in the positive definite sense).

2) measurement update: use $\bar{x}(k+1)$ and $\bar{P}(k+1)$ (a priori info) and the measurement $z(k+1)$ to get an improved state estimate with a reduced estimation error covariance, due to our measurement update. This next step has been solved previously in class, in the review of linear algebra.

Bar-Shalom derivation: get distribution of $[x(k+1)\ z(k+1)]^T$, conditioned on Z^{k+1}, and solve by analogy to static case.

$$\bar{z}(k+1) = \mathbb{E}\left[z(k+1)|Z^k\right] \tag{9.16}$$
$$\bar{z}(k+1) = \mathbb{E}\left[H(k+1)x(k+1) + w(k+1)|Z^{k+1}\right] \tag{9.17}$$
$$\bar{z}(k+1) = H(k+1)\mathbb{E}\left[x(k+1)|z^{k+1}\right] + \cancel{\mathbb{E}\left[w(k+1)|Z^{k+1}\right]} \tag{9.18}$$
$$\bar{z}(k+1) = H(k+1)\bar{x}(k+1) \tag{9.19}$$

$$\bar{P}_{zz}(k+1) = \mathbb{E}\left[(z(k+1) - \bar{z}(k+1))(z(k+1) - \bar{z}(k+1))^T|Z^k\right]$$
$$\bar{P}_{zz}(k+1) = \mathbb{E}\left[\{H(k+1)(x(k+1) - \bar{x}(k+1)) + w(k+1)\}\{H(k+1)(x(k+1) - \bar{x}(k+1)) + w(k+1)\}^T|Z^k\right]$$
$$\bar{P}_{zz}(k+1) = H(k+1)\bar{P}(k+1)H^T(k+1) + R(k+1)$$

$$\bar{P}_{xz}(k+1) = \mathbb{E}\left[(x(k+1) - \bar{x}(k+1))(z(k+1) - \bar{z}(k+1))^T|Z^k\right]$$
$$\bar{P}_{xz}(k+1) = \mathbb{E}\left[(x(k+1) - \bar{x}(k+1))(x(k+1) - \bar{x}(k+1))^T\right]H^T(k+1) +$$
$$\cancel{\mathbb{E}\left[(x(k+1) - \bar{x}(k+1))w^T(k+1)\right]}$$
$$\bar{P}_{xz}(k+1) = \bar{P}(k+1)H^T(k+1)$$

Thus, we have all the moments required to specify $P([x(k+1)z(k+1)]^T|z^k)$, so

$$\begin{bmatrix}x(k+1)\\z(k+1)\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\bar{x}(k+1)\\H(k+1)\bar{x}(k+1)\end{bmatrix}, \begin{bmatrix}\bar{P} & \bar{P}H^T\\H\bar{P} & H\bar{P}H^T + R\end{bmatrix}\right) \tag{9.20}$$

where the $(k+1)$ index on the elements of the covariance matrix are suppressed for brevity, and the distribution is conditioned on z^k.

We seek:

$$\begin{aligned}\hat{x}(k+1) &= \mathbb{E}\left[x(k+1)|Z^{k+1}\right]\\ &= \mathbb{E}\left[x(k+1)|z(k+1), Z^k\right]\\ &= \mathbb{E}\left[x(k+1)|z(k+1)\right]\end{aligned}$$

Here, the conditioning on Z^k is implicit; by suppressing this conditioning, the form of this problem is made to resemble the static case. So, the problem has now been reduced to one we've already solved (in Linear Estimation of Static Systems):

$$\hat{x}(k+1) = \hat{x}(k+1) + \bar{P}_{xz}(k+1)\bar{P}_{zz}^{-1}(k+1)[z(k+1) - \bar{z}(k+1)]$$
$$\bar{P}(k+1) = \bar{P}(k+1) - \bar{P}_{xz}(k+1)\bar{P}_{zz}^{-1}(k+1)\bar{P}_{xz}^T(k+1)$$

substituting:

$$\hat{x} = \hat{x} + \bar{P}H^T \left[H\bar{P}H^T + R\right]^{-1} [z - H\bar{x}] \tag{9.21}$$
$$\bar{P} = \bar{P} - \bar{P}H^T[H\bar{P}H^T + R]^{-1}H\bar{P} \tag{9.22}$$

where the $(k+1)$ index is suppressed on each term in both expressions for brevity.

Using Bar-Shalom notation:

$$\begin{aligned}
S(k+1) &= P_{zz} = H(k+1)\bar{P}(k+1)H^T(k+1) + R(k+1) && \text{(“innovation covariance”)}\\
\nu(k+1) &= z(k+1) - H(k+1)\bar{x}(k+1) && \text{(“innovation”)}\\
W(k+1) &= \bar{P}(k+1)H^T(k+1)S(k+1) && \text{(“Kalman gain matrix”)}\\
\hat{x}(k+1) &= \bar{x}(k+1) + W(k+1)\nu(k+1)\\
P(k+1) &= \bar{P}(k+1) - W(k+1)S(k+1)W^T(k+1)
\end{aligned}$$

Summary:
given $\hat{x}(0)$, $P(0)$,
0) set $k = 0$
1) propagate state and covariance (compute $\bar{x}(k+1)$, $\bar{P}(k+1)$
2) measurement update of state and covariance:
compute:

$$\begin{aligned}
\nu(k+1) &= z(k+1) - H(k+1)\bar{x}(k+1) && \text{(“innovation”)}\\
S(k+1) &= H(k+1)\bar{P}(k+1)H^T(k+1) + R(k+1) && \text{(“innovation covariance”)}\\
W(k+1) &= \bar{P}(k+1)H^T(k+1)S(k+1) && \text{(“gain”)}\\
\hat{x}(k+1) &= \bar{x}(k+1) + W(k+1)\nu(k+1) && \text{(“a posteriori” or “filtered” state estimate)}\\
P(k+1) &= \bar{P}(k+1) - W(k+1)S(k+1)W^T(k+1) && \text{(“a posteriori state error covariance”)}
\end{aligned}$$

3) $k \leftarrow k+1$
4) goto (1)

9.2 Alternative formulas for covariance and gain of Kalman filter

Note: These are algebraically equivalent to the formulas given previously

$$P(k+1) = [\bar{P}^{-1}(k+1) + H^T(k+1)R^{-1}(k+1)H(k+1)]^{-1} \tag{9.23}$$
$$\begin{aligned}
P(k+1) &= [I - W(K+1)H(k+1)]\bar{P}(k+1)[I - W(K+1)H(k+1)]^T\\
&\quad \ldots + W(K+1)R(k+1)W^T(K+1)
\end{aligned} \tag{9.24}$$

The second of these is called the “Joseph Form” of the state covariance update. It guarantees that $P(k+1) > 0$, even with the limited precision of a computer. The Kalman filter assumes that $P(k+1)$ is always positive definite, so this can help implementation.

$$W(k+1) = P(k+1)H^T(k+1)R^{-1}(k+1) \tag{9.25}$$

Since $P(k+1)$ is usually calculated using $W(k+1)$, this last form is more useful for derivation and analysis than as part of the (causal) filter.

9.3 Interpretation of the Kalman gain

What is the $W(k+1)$? Again repeat the definition of it:

$$W(k+1) = \bar{P}(k+1)H^T(k+1)[H(k+1)\bar{P}(k+1)H^T(k+1) + R(k+1)]^{-1} \tag{9.26}$$

$$\hat{x}(k+1) = \bar{x}(k+1) + W(k+1)\nu(k+1) \tag{9.27}$$

Remarks

- As $W(k+1) \to 0$ then new measurements are not taken into account in $\hat{x}(k+1)$
- If $\bar{P}(k+1)$ is very small then $W(k+1)$ is also small and $\hat{x}(k+1)$ differs little from $\bar{x}(k+1)$ ($\bar{P}(k+1)$ being small implies that $\bar{x}(k+1)$ is a good estimate)
- If $\bar{P}(k+1)$ is so big that $H(k+1)\bar{P}(k+1)H^T(k+1) \gg R(k+1)$ then $W(k+1)$ approaches an upper limit. In this case we trust the measurements almost entirely in the subspace in which they give information.

The subspace in which the measurements provide information

If $n_z < n_x$ then the measurements provide information in an n_z-dimension subspace that is determined by $H(k+1)$. Specifically, this subspace is the complement of the null space of $H(k+1)$, or the range of $H(k+1)$:

$$z(k+1) \in range[H(k+1)] = (null[H(k+1)])^{\perp}$$

$\bar{P}(k+1)$ gives the weighting of the update in this subspace.

This can be visualized in two dimensions:

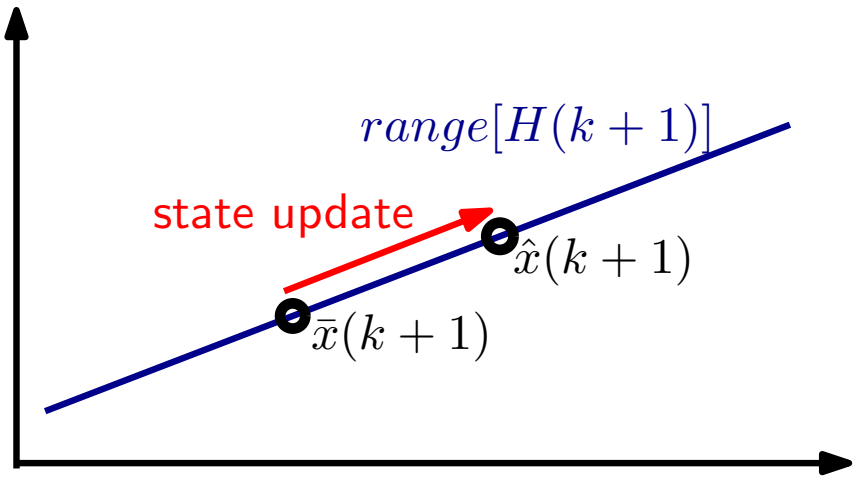


Figure 9.1: The state update occurs in the subspace in which the measurements provide information

9.3.1 Generalization for Weighted Process Noise

We can modify the dynamics equations to have weightings on the noise such that:

$$x(k+1) = F(k)x(k) + G(k)u(k) + \Gamma(k)v(k) \tag{9.28}$$

Here setting $\Gamma = I$ recovers the previous (unweighted) form. With this generalization, the only change in the Kalman filter is in the covariance propagation, which is now given by:

$$\bar{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k) \tag{9.29}$$

9.4 MAP-based Kalman Filter Derivation

For the linear Kalman filter under our discrete-time stochastic-linear-time-varying (SLTV) problem, the pdfs are all Gaussian. As a result, $\hat{x}_{\text{MAP}}(k) = \hat{x}_{\text{MMSE}}(k) = \mathbb{E}\left[x(k)|Z^k\right]$ (i.e., MAP and MMSE are equivalent). For nonlinear systems, MAP and MMSE are not necessarily equivalent, and MAP is generally a more practical approach.

MAP benefits:

- More practical than MMSE for nonlinear systems
- Aids interpretation of square-root filtering
- Aids in statistical hypothesis testing of the Kalman filter

We will derive the MAP estimator for SLTV systems.

9.4.1 Setting up the cost function

In the MAP derivation, we estimate the process noise $v(k)$ along with the state $x(k+1)$. Conditioning on Z^k is implied throughout the derivation but not shown explicitly for brevity. We begin with the conditional distribution

$$p[x(k+1), v(k)|z(k+1)] = \frac{p[z(k+1)|x(k+1), v(k)]p[x(k+1), v(k)]}{p[z(k+1)]}$$

The MAP approach tells us to find the $x(k+1)$ and $v(k)$ that maximize $p[x(k+1), v(k)|z(k+1)]$. We can ignore $p[z(k+1)]$, since it doesn't depend on $x(k+1)$ or $v(k)$, and just maximize the numerator. This is equivalent to minimizing the cost function

$$J[x(k+1), v(k)] = -\log(p[z(k+1)|x(k+1), v(k)]) - \log(p[x(k+1), v(k)])$$

where log() denotes the natural logarithm.

In the next step, we'll exploit the fact that

$$p_{x(k+1),v(k)}[x(k+1), v(k)] = c \cdot p_{x(k),v(k)}[x(k), v(k)]$$

where c is a constant and the subscripts make clear that these are two distinct probability distributions.

Q: Why can we make the above transformation from $p[x(k+1), v(k)]$ to $p[x(k), v(k)]$?
A: Because $x(k+1)$ is a function of $x(k)$; specifically,

$$x(k+1) = F(k)x(k) + G(k)u(k) + \Gamma(k)v(k)$$

In general, for any invertible 1-to-1 scalar function $Y = g(X)$, it can be shown that

$$p_Y[y] = \frac{p_X[g^{-1}(y)]}{|\frac{dy}{dx}|}$$

For vector-valued functions, the scaling factor in the denominator is the determinant of the Jacobian $[\partial y/\partial x]$. Thus, in the case of $x(k+1)$ and $v(k)$ above, $c = |F^{-1}(k)|$ ■

Applying this transformation, the cost function becomes the sum of three components plus a constant, which we can ignore:

$$\begin{aligned} J[x(k+1), v(k)] = &\tfrac{1}{2}[z(k+1) - H(k+1)x(k+1)]^T R^{-1}(k+1)[\dots]\} \quad \text{from } p[z(k+1)|x(k+1), v(k)] \\ &+ \tfrac{1}{2}[x(k) - \hat{x}(k)]^T P^{-1}(k)[x(k) - \hat{x}(k)] + \tfrac{1}{2}v^T(k)Q^{-1}(k)v(k)\} \quad \text{from } p[x(k), v(k)] \\ &+ \text{const} \quad \text{(ignore)} \end{aligned}$$

Substituting $x(k) = F^{-1}[x(k+1) - G(k)u(k) - \Gamma(k)v(k)]$ yields

$$\begin{aligned} J[x(k+1), v(k)] = &\tfrac{1}{2}[z(k+1) - H(k+1)x(k+1)]^T R^{-1}(k+1)[\dots] \\ &+ \tfrac{1}{2}\left\{F^{-1}[x(k+1) - G(k)u(k) - \Gamma(k)v(k)] - \hat{x}(k)\right\}^T P^{-1}(k)\{\dots\} \\ &+ \tfrac{1}{2}v^T(k)Q^{-1}(k)v(k) \end{aligned}$$

Recall that $\bar{x}(k+1) = F(k)\hat{x}(k) + G(k)u(k)$; then

$$\begin{aligned} J[x(k+1), v(k)] = &\tfrac{1}{2}[z(k+1) - H(k+1)x(k+1)]^T R^{-1}(k+1)[\dots] \\ &+ \tfrac{1}{2}\left\{F^{-1}[x(k+1) - \Gamma(k)v(k) - \bar{x}(k+1)]\right\}^T P^{-1}(k)\{\dots\} \\ &+ \tfrac{1}{2}v^T(k)Q^{-1}(k)v(k) \end{aligned}$$

9.4.2 Solving for the minimum-cost estimate

We minimize $J[x(k+1), v(k)]$ by finding the $x(k+1)$ and $v(k)$ that satisfy the first-order necessary conditions

$$\begin{aligned} \left[\frac{\partial J}{\partial v(k)}\right]^T = 0 = &-\Gamma^T(k)F^{-T}(k)P^{-1}(k)F^{-1}(k)[x(k+1) - \Gamma(k)v(k) - \bar{x}(k+1)] + Q^{-1}(k)v(k) \\ \left[\frac{\partial J}{\partial x(k+1)}\right]^T = 0 = &-H^T(k+1)R^{-1}(k+1)[z(k+1) - H(k+1)x(k+1)] \\ &+ F^{-T}(k)P^{-1}(k)F^{-1}(k)[x(k+1) - \Gamma(k)v(k) - \bar{x}(k+1)] \end{aligned}$$

We could put these two equations into the form

$$A\begin{bmatrix} x(k+1) \\ v(k) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

and solve by taking the inverse of A. But instead we'll solve by subsitution. We begin by isolating $v(k)$ in the first equation:

$$v(k) = [\Gamma^T(k)F^{-T}(k)P^{-1}(k)F^{-1}(k)\Gamma(k) + Q^{-1}(k)]^{-1}\Gamma^T(k)F^{-T}(k)P^{-1}(k)F^{-1}(k)[x(k+1) - \bar{x}(k+1)]$$

Applying the matrix inversion lemma and the definition of $\bar{P}(k+1)$ we obtain

$$\begin{aligned} v(k) &= Q(k)\Gamma^T(k)[\underbrace{F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k)}_{\bar{P}(k+1)}]^{-1}[x(k+1) - \bar{x}(k+1)] \\ &= Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)[x(k+1) - \bar{x}(k+1)] \end{aligned}$$

We substitute this expression for $v(k)$ into the $[\partial J/\partial x(k+1)]^T = 0$ equation to get

$$\begin{aligned}
0 =& - H^T(k+1)R^{-1}(k+1)[z(k+1) - H(k+1)x(k+1)] \\
&+ F^{-T}(k)P^{-1}(k)F^{-1}(k)\left\{x(k+1) - \Gamma(k)Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)[x(k+1) - \bar{x}(k+1)] - \bar{x}(k+1)\right\} \\
0 =& - H^T(k+1)R^{-1}(k+1)[z(k+1) - H(k+1)x(k+1)] \\
&+ F^{-T}(k)P^{-1}(k)F^{-1}(k)[\underbrace{I}_{\bar{P}\bar{P}^{-1}} - \Gamma(k)Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)][x(k+1) - \bar{x}(k+1)] \\
0 =& - H^T(k+1)R^{-1}(k+1)[z(k+1) - H(k+1)x(k+1)] \\
&+ \underbrace{F^{-T}(k)P^{-1}(k)F^{-1}(k)[\bar{P}^{-1}(k+1) - \Gamma(k)Q(k)\Gamma^T(k)]}_{I}\bar{P}^{-1}(k+1)[x(k+1) - \bar{x}(k+1)]
\end{aligned}$$

By replacing $\bar{P}(k+1)$ with its definition, the underbraced quantity can be recognized as the identity matrix. We can then solve the simplified equation for $x(k+1)$ to obtain $\hat{x}(k+1)$:

$$\begin{aligned}
\hat{x}(k+1) = &[\bar{P}^{-1}(k+1) + H^T(k+1)R^{-1}(k+1)H(k+1)]^{-1} \\
&\times [\bar{P}^{-1}(k+1)\bar{x}(k+1) + H^T(k+1)R^{-1}(k+1)z(k+1)]
\end{aligned}$$

This is identical to the recursive least squares form. Further manipulation yields

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

with

$$W(k+1) = [\bar{P}^{-1}(k+1) + H^T(k+1)R^{-1}(k+1)H(k+1)]^{-1}H^T(k+1)R^{-1}(k+1)$$

which is equivalent to our MMSE formula for the *a posteriori* state estimate. Substituting this formula into the one for $v(k)$ gives

$$\begin{aligned}
\hat{v}(k) \triangleq \mathbb{E}\left[v(k)|Z^{k+1}\right] &= Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)[\hat{x}(k+1) - \bar{x}(k+1)] \\
&= Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)W(k+1)[z(k+1) - H(k+1)\bar{x}(k+1)]
\end{aligned}$$

This is the MAP estimate of the process noise "looking back" after having taken the measurement $z(k+1)$.

9.4.3 Summary

For the SLTV system, the MMSE and MAP estimates for $x(k+1)$ given Z^{k+1} are identical. But the MAP derivation has yields as a by-product an estimate of the process noise that drove the system from k to $k+1$:

$$\begin{aligned}
\hat{x}(k+1) &= \bar{x}(k+1) + W(k+1)[z(k+1) - H(k+1)\bar{x}(k+1)] \\
\hat{v}(k) &= Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)W(k+1)[z(k+1) - H(k+1)\bar{x}(k+1)]
\end{aligned}$$

where $W(k+1)$ is the Kalman gain matrix. Note that, even with a perfect model, $\hat{v}(k)$ is not necessarily equal to $v(k)$ because the former is conditioned on $z(k+1)$. The conditioning also causes $\hat{v}(k)$ to be correlated with $\hat{v}(j)$ for $j \neq k$ so that, unlike $v(k)$, $\hat{v}(k)$ is not white.

9.5 Stability of Kalman Filter

The general question of the stability of a Kalman filter applied to a stochastic linear time-varying (SLTV) system has not yet been resolved. Our approach to KF stability analysis will be to "cheat" in two different ways. First, we'll examine KF in the "zero-input" case of no stochastic inputs to our dynamical system. Second, we'll examine the case of a linear time invariant (LTI) system with stochastic inputs.

9.5.1 Zero-Input Stability

Assume $\boldsymbol{v}(k) = \boldsymbol{w}(k) = 0 \ \forall k$ (zero-input stability). We want to show that the error vector

$$\boldsymbol{e}(k) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k)$$

decays to zero.

(Almost) proof:

$$\begin{aligned}\boldsymbol{e}(k+1) &= \boldsymbol{x}(k+1) - \hat{\boldsymbol{x}}(k+1) \\ &= F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k) - [\bar{\boldsymbol{x}}(k+1) + W(k+1)\{\boldsymbol{z}(k+1) - H(k+1)\bar{\boldsymbol{x}}(k+1)\}]\end{aligned}$$

But

$$\begin{aligned}\boldsymbol{z}(k+1) &= H(k+1)\boldsymbol{x}(k+1) = H(k+1)[F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k)] \\ \bar{\boldsymbol{x}}(k+1) &= F(k)\hat{\boldsymbol{x}}(k) + G(k)\boldsymbol{u}(k)\end{aligned}$$

Substituting these causes the $\boldsymbol{u}(k)$'s to cancel, giving the error dynamics for $\boldsymbol{v}(k) = \boldsymbol{w}(k) = 0 \ \forall k$:

$$\boldsymbol{e}(k+1) = [I - W(k+1)H(k+1)]F(k)\boldsymbol{e}(k)$$

It is tricky to prove stability because the system implied by the error dynamics is time varying. If it were time invariant, it would be possible to look at the modulus of the system's eigenvalues. Instead, we will use a Lyapunov-type energy method to prove stability. Define an energy-like function

$$V[k, \boldsymbol{e}(k)] = \tfrac{1}{2}\ \boldsymbol{e}^T(k)P^{-1}(k)\boldsymbol{e}(k)$$

Note that $\sqrt{V}$ is a weighted 2-norm of $\boldsymbol{e}(k)$ because $P(k) > 0$. Therefore, $V \geq 0$, with equality if and only if $\boldsymbol{e}(k) = 0$. We wish to show that V always gets smaller as k increases.

$$\begin{aligned}V[k+1, \boldsymbol{e}(k+1)] &= \tfrac{1}{2}\boldsymbol{e}^T(k+1)P^{-1}(k+1)\boldsymbol{e}(k+1) \\ &= \tfrac{1}{2}\boldsymbol{e}^T(k)[P(k) + D(k)]^{-1}\boldsymbol{e}(k)\end{aligned}$$

where

$$D(k) \triangleq F^{-1}(k)[\Gamma(k)Q(k)\Gamma(k)^T + \bar{P}(k+1)H^T(k+1)R^{-1}(k+1)H(k+1)\bar{P}(k+1)]F^{-T}(k)$$

Note that $D(k) \geq 0$, which implies $[P(k)+D(k)]^{-1} < P^{-1}(k)$. Thus $V[k+1, e(k+1)] \leq V[k, e(k)]$. But the fact that V always gets smaller as k increases isn't quite enough for our proof because V could be getting smaller due to $P(k)$ increasing, not $e(k)$ decreasing.

It turns out that, under suitable conditions on $Q(k)$, $R(k)$, $F(k)$, and $H(k)$ (i.e. all unstable or neutrally stable subspaces of original system are observable, Q and R are not too big or small, system is controllable with respect to points of entry of process noise, or "stochastic controllability and observability") we can show that

1. $P^{-1}(k) \geq \epsilon > 0$ for some bound ϵ.
2. $V[k+N, \boldsymbol{e}(k+N)] < \alpha V[k, \boldsymbol{e}(k)]$ for some α: $0 \leq \alpha < 1$ and some N.

Thus, V is decreasing because $e(k)$ is decreasing, not because $P(k)$ is increasing. Also note that *nowhere in this proof did we assume the original system was stable.* Indeed, it need not be for the *estimation error dynamics* to be stable.

9.5.2 Matrix Riccati Equation

Recall that

$$\bar{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k)$$

We substitute for $P(k)$ to get

$$\begin{aligned}\bar{P}(k+1) = F(k)\left\{\bar{P}(k) - \bar{P}(k)H^T(k)[H(k)\bar{P}(k)H^T(k) + R(k)]^{-1}H(k)\bar{P}(k)\right\}F^T(k) \\ + \Gamma(k)Q(k)\Gamma^T(k)\end{aligned}$$

This dynamic model for $\bar{P}(k)$, is nonlinear in $\bar{P}(k)$, is called the Matrix Riccati equation (MRE). Beware, a general-case analysis of MRE is not easy! We "cheat" by limiting ourselves to the special case of an LTI system: $F(k) = F$, $\Gamma(k) = \Gamma$, $H(k) = H$, $Q(k) = Q$, $R(k) = R$.

If the pair (F, H) is observable, and if (F, Γ) is controllable, and if Q, $R > 0$, and $\bar{P}(0) > 0$, then the MRE converges to a steady-state solution $\bar{P}_{ss} > 0$, which can be determined by solving the Algebraic Riccati Equation (ARE):

$$\bar{P}_{ss} = F\left\{\bar{P}_{ss} - \bar{P}_{ss}H^T[H\bar{P}_{ss}H^T + R]^{-1}H\bar{P}_{ss}\right\}F^T + \Gamma Q\Gamma^T$$

This form is called the ARE because it's algebraic, not dynamic like the MRE; but it remains nonlinear in $\bar{P}_{ss}$. Note that the form is similar to the discrete-time Lyapunov equation for the steady-state covariance of a linear dynamic system.

The MATLAB function `kalman` can be used to solve the ARE and provide the Kalman gain matrix W_{ss}:

$$\begin{aligned}\texttt{sys} &= \texttt{ss}(F, [B\ \Gamma], H, 0, -1); \\ [\texttt{km}, L, \bar{P}_{ss}, W_{ss}] &= \texttt{kalman}(\texttt{sys}, Q, R);\end{aligned}$$

Apart from $\bar{P}_{ss}$ and W_{ss}, this command returns the state-space model of the Kalman filter in `km` and a gain matrix L that can be used in an alternate form of the Kalman filter.

9.5.3 Steady-State KF Equations

The steady-state Kalman Filter equations are given by

$$\begin{aligned}\bar{\boldsymbol{x}}(k+1) &= F\hat{\boldsymbol{x}}(k) + G\boldsymbol{u}(k) \\ \hat{\boldsymbol{x}}(k+1) &= \bar{\boldsymbol{x}}(k+1) + W_{ss}[\boldsymbol{z}(k+1) - H(k+1)\bar{\boldsymbol{x}}(k+1)]\end{aligned}$$

9.5.4 Steady-State Error Dynamics

The steady-state error dynamics are given by

$$\boldsymbol{e}(k+1) = \underbrace{[I - W_{ss}H]F}_{A_{ss}}\boldsymbol{e}(k)$$

We know that the dynamics of $\boldsymbol{e}(k)$ are stable—they must be for a steady-state solution to exist. Thus, it must be the case that

$$\max_i |\lambda_i| < 1$$

where $\{\lambda_i\}_i^{n_x}$ are the eigenvalues of A_{ss}.

Again, keep in mind that this holds whether or not the original system dynamics are stable.

9.6 Filter Initialization and Modeling Errors

9.6.1 Filter Initialization

The Kalman filter requires an initial estimates $\boldsymbol{x}$, which we denote as $\hat{\boldsymbol{x}}(0)$, and an associated error covariance matrix, $P(0)$.

How do we arrive at values for $\hat{\boldsymbol{x}}(0)$ and $P(0)$?

1. We may be able to measure $\boldsymbol{x}(0)$ directly; $P(0)$ reflects the accuracy of the sensor used for this measurement.

2. We may have some a priori knowledge of $\hat{\boldsymbol{x}}(0)$ and $P(0)$.

3. We can set $\hat{\boldsymbol{x}}(0)$ arbitrarily and let $P(0) = c \cdot I$, with c very large. Problems with this approach:

   (a) In linear problems, the wrong $\hat{\boldsymbol{x}}(0)$ and $P(0)$ will make initial filter performance suboptimal.

   (b) In nonlinear problems, the filter may diverge and never recover.

4. We can apply non-Bayesian estimation to a block of early data to obtain $\hat{\boldsymbol{x}}$ and P at the end of the block.

Let us further examine method 4. Measurements $\boldsymbol{z}(k),\ k = 1, 2, \ldots$ can be written in terms of the state, input, and noises as

$$\boldsymbol{z}(1) = H(1)\boldsymbol{x}(1) + \boldsymbol{w}(1) = H(1)[F(0)\boldsymbol{x}(0) + G(0)\boldsymbol{u}(0) + \Gamma(0)\boldsymbol{v}(0)] + \boldsymbol{w}(1)$$
$$\boldsymbol{z}(2) = H(2)\{F(1)[F(0)\boldsymbol{x}(0) + G(0)\boldsymbol{u}(0) + \Gamma(0)\boldsymbol{v}(0)] + G(1)\boldsymbol{u}(1) + \Gamma(1)\boldsymbol{v}(1)\} + \boldsymbol{w}(2)$$

etc. up to the kth measurement. These measurements can be stacked in the form

$$\boldsymbol{z}^k = H^k\boldsymbol{x}(0) + G^{k-1}\boldsymbol{u}^{k-1} + \Gamma^{k-1}\boldsymbol{v}^{k-1} + \boldsymbol{w}^k$$

where

$$\underbrace{\boldsymbol{z}^k = \begin{bmatrix} \boldsymbol{z}(1) \\ \boldsymbol{z}(2) \\ \vdots \\ \boldsymbol{z}(k) \end{bmatrix}}_{(n_z \cdot k \times 1)}, \quad \underbrace{H^k = \begin{bmatrix} H(1)F(0) \\ H(2)F(1)F(0) \\ \vdots \\ H(k)F(k-1)F(k-2)\ldots F(0) \end{bmatrix}}_{(n_z \cdot k) \times n_x}, \quad \underbrace{\boldsymbol{u}^{k-1} = \begin{bmatrix} \boldsymbol{u}(0) \\ \boldsymbol{u}(1) \\ \vdots \\ \boldsymbol{u}(k-1) \end{bmatrix}}_{(n_u \cdot k) \times 1}$$

$$G^{k-1} = \underbrace{\begin{bmatrix} H(1)G(0) & 0 & \dots & 0 \\ H(2)F(1)G(0) & H(2)G(1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \underbrace{H(k)F(k-1)\dots F(1)G(0)}_{\boldsymbol{u}(0) \text{ columns}} & \underbrace{H(k)F(k-1)\dots F(2)G(1)}_{\boldsymbol{u}(1) \text{ columns}} & \dots & \underbrace{H(k)G(k-1)}_{\boldsymbol{u}(k-1) \text{ columns}} \end{bmatrix}}_{(n_z \cdot k) \times (n_u \cdot k)}$$

$$\boldsymbol{v}^{k-1} = \underbrace{\begin{bmatrix} \boldsymbol{v}(0) \\ \boldsymbol{v}(1) \\ \vdots \\ \boldsymbol{v}(k-1) \end{bmatrix}}_{(n_v \cdot k) \times 1}, \quad \boldsymbol{w}^k = \underbrace{\begin{bmatrix} \boldsymbol{w}(1) \\ \boldsymbol{w}(2) \\ \vdots \\ \boldsymbol{w}(k) \end{bmatrix}}_{(n_z \cdot k) \times 1}, \quad \Gamma^{k-1} = \underbrace{\begin{bmatrix} \text{same as } G^{k-1} \text{ but with} \\ G(j) \text{ replaced by } \Gamma(j) \end{bmatrix}}_{(n_z \cdot k) \times (n_v \cdot k)}$$

Let

$$R^k = \mathbb{E}\left[\boldsymbol{w}^k \boldsymbol{w}^{k^T}\right] = \begin{bmatrix} R(1) & & & \\ & R(2) & & \\ & & \ddots & \\ & & & R(k) \end{bmatrix}, \quad Q^{k-1} = \mathbb{E}\left[v^{k-1} v^{k-1^T}\right] = \begin{bmatrix} Q(0) & & & \\ & Q(1) & & \\ & & \ddots & \\ & & & Q(k-1) \end{bmatrix}$$

with the remaining elements zero.

Our stacked measurement equation, together with R^k and Q^{k-1}, define a linear least-squares batch estimation problem that does not depend on any *a priori* $\hat{\boldsymbol{x}}(0)$ or $P(0)$. Our cost function can be written as

$$J\left[\boldsymbol{x}(0), \boldsymbol{v}^{k-1}\right] = \tfrac{1}{2}\left[\boldsymbol{z}^k - H^k \boldsymbol{x}(0) - G^{k-1}\boldsymbol{u}^{k-1} - \Gamma^{k-1}\boldsymbol{v}^{k-1}\right]^T (R^k)^{-1}[\dots] + \tfrac{1}{2}\left[\boldsymbol{v}^{k-1}\right]^T (Q^{k-1})^{-1}[\dots]$$

The cost function properly weights errors in measurement and process noise. We minimize the cost function by solving

$$0 = \left[\frac{\partial J}{\partial \boldsymbol{x}(0)}\right]^T, \quad 0 = \left[\frac{\partial J}{\partial \boldsymbol{v}^{k-1}}\right]^T$$

Note that

$$\begin{bmatrix} P_{\boldsymbol{x}\boldsymbol{x}}(0) & P_{\boldsymbol{x}\boldsymbol{v}^{k-1}} \\ P^T_{\boldsymbol{x}\boldsymbol{v}^{k-1}} & P_{\boldsymbol{v}^{k-1}\boldsymbol{v}^{k-1}} \end{bmatrix} = \left[\frac{\partial^2 J}{\partial \begin{pmatrix} \boldsymbol{x}(0) \\ \boldsymbol{v}^{k-1} \end{pmatrix}^2}\right]^T$$

Remarks:

1. We must make k large enough so that the minimum is unique: H^k should have more rows than columns and the n_x columns of H^k should be linearly independent. (Note that H^k functions like an observability matrix.)
2. The resulting $\hat{\boldsymbol{x}}(0)$ is $E[\boldsymbol{x}(0)|\boldsymbol{z}^k]$, a smoothed estimate of $vbx(0)$ using future data.
3. We usually want $E[\boldsymbol{x}(k)|\boldsymbol{z}^k]$. We can iterate the initial estimate forward using our knowledge of the dynamics and our estimates for $\boldsymbol{v}(j)$ to get $\hat{\boldsymbol{x}}(k|k)$. We can also iterate $P_{\boldsymbol{x}\boldsymbol{x}}(0)$ forward to get an estimate for $P(k)$. We then start our Kalman filter at time k.
4. For a linear system, this is equivalent to choosing $\hat{\boldsymbol{x}}(0)$ arbitrarily, setting $P(0) = \infty$ and running the Kalman filter from 0 to k.

9.6.2 Modeling Errors

Suppose we implement a Kalman filter according to the model

$$\begin{aligned}\boldsymbol{x}(k+1) &= F_f(k)\boldsymbol{x}(k) + G_f(k)\boldsymbol{u}(k) + \boldsymbol{v_f}(k)\\ \boldsymbol{z}(k) &= H_f(k)\boldsymbol{x}(k) + \boldsymbol{w_f}(k)\end{aligned}$$

$$\begin{aligned}&E[\boldsymbol{v}_f] = 0, && E[\boldsymbol{w}_F] = 0\\ &E[\boldsymbol{v}_f(j)\boldsymbol{v}_f^T(k)] = \delta_{jk}Q_f(k) && E[\boldsymbol{w}_f(j)\boldsymbol{w}_f^T(k)] = \delta_{jk}R_f(k)\end{aligned}$$

leading to a Kalman filter with $S_f(k)$, $W_f(k)$, etc.

But suppose the real system obeys

$$\begin{aligned}\boldsymbol{x}(k+1) &= F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k) + \boldsymbol{v}(k)\\ \boldsymbol{z}(k) &= H(k)\boldsymbol{x}(k) + \boldsymbol{w}(k)\end{aligned}$$

$$\begin{aligned}&E[\boldsymbol{v}] = 0, && E[\boldsymbol{w}] = 0\\ &E[\boldsymbol{v}(j)\boldsymbol{v}^T(k)] = \delta_{jk}Q(k) && E[\boldsymbol{w}(j)\boldsymbol{w}^T(k)] = \delta_{jk}R(k)\end{aligned}$$

with $F(k) \neq F_f(k)$, $G(f) \neq G_f(k)$, etc. (modeling errors). Large errors can yield unpleasant effects.

Let

$$\begin{aligned}\bar{\boldsymbol{e}}(k) &\triangleq \boldsymbol{x}(k) - \bar{\boldsymbol{x}}(k) \quad \text{(a priori error)}\\ \boldsymbol{e}(k) &\triangleq \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k) \quad \text{(a posteriori error)}\end{aligned}$$

We can develop a dynamics model for these errors:

$$\begin{aligned}\bar{\boldsymbol{e}}(k+1) &= F_f(k)\boldsymbol{e}(k) + [F(k) - F_f(k)]\boldsymbol{x}(k) + [G(k) - G_f(k)]\boldsymbol{u}(k) + \boldsymbol{v}(k)\\ \boldsymbol{e}(k+1) &= [I - W_f(k+1)H_f(k+1)]\bar{e}(k+1) + W_f(k+1)[H_f(k+1)\\ &\quad - H(k+1)]\boldsymbol{x}(k+1) + W_f(k+1)\boldsymbol{w}(k+1)\end{aligned}$$

Substituting $\bar{\boldsymbol{e}}(k+1)$ into $\boldsymbol{e}(k+1)$, we get

$$\boldsymbol{e}(k+1) = [I - W_f(k+1)H_f(k+1)]F_f(k)\boldsymbol{e}(k) + T_1 + T_2$$

where T_1 is a function of our usual noise terms $\boldsymbol{v}(k)$ and $\boldsymbol{w}(k)$, and T_2 includes terms multiplying $\boldsymbol{x}(k)$, $\boldsymbol{u}(k)$, and $\boldsymbol{v}(k)$ whose matrix coefficients include the model mismatch errors $[F-F_f]$, $[G-G_f]$, and $[H - H_f]$. These terms do nasty things to the Kalman filter behavior and complicate our analysis.

A full analysis requires computation of $E[\boldsymbol{e}(k+1)\boldsymbol{e}^T(k+1)] = P^{\text{true}}(k+1)$, which depends on the joint statistics of $\boldsymbol{x}(k)$, $\boldsymbol{u}(k)$, $\boldsymbol{v}(k)$, and $\boldsymbol{w}(k)$ (see Bar-Shalom 5.6.1).

Upshot: if the errors $[F(k) - F_f(k)]$, etc. are not too large, then the modeling errors in the filter will be small and the filter can still perform well: $P^{true}(k+1)$ will not be much larger than $P(k+1)$. But if the modeling errors are too large then the system will perform poorly.

- May want to improve system model via system identification techniques.
- May want to perform a "what-if?" simulation analysis.

9.7 Stability and Consistency of Kalman Filter

9.7.1 Stability of KF

Assume $\boldsymbol{v}(k) = \boldsymbol{w}(k) = 0\ \forall k$ (zero-input stability). We want to show that the error vector decays to zero:

$$\boldsymbol{e}(k) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k) \tag{9.30}$$

Pseudo Proof:

$$\begin{aligned}\boldsymbol{e}(k+1) &= \boldsymbol{x}(k+1) - \hat{\boldsymbol{x}}(k+1) &(9.31)\\ &= F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k) - [\bar{\boldsymbol{x}}(k+1) + W(k+1)\{\boldsymbol{z}(k+1) - H(k+1)\bar{\boldsymbol{x}}(k+1)\}] &(9.32)\end{aligned}$$

However,

$$\begin{aligned}\boldsymbol{z}(k+1) &= H(k+1)\boldsymbol{x}(k+1) = H(k+1)[F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k)] &(9.33)\\ \bar{\boldsymbol{x}}(k+1) &= F(k)\hat{\boldsymbol{x}}(k) + G(k)\boldsymbol{u}(k) &(9.34)\end{aligned}$$

Substituting the above causes the $\boldsymbol{u}(k)$'s to cancel, giving the error dynamics for $\boldsymbol{v}(k) = \boldsymbol{w}(k) = 0$ $\forall k$:

$$\boldsymbol{e}(k+1) = [I - W(k+1)H(k+1)]F(k)\boldsymbol{e}(k) \tag{9.35}$$

It is a little tricky to prove stability because this is a time varying system. If the system was not time varying, it would be possible to look at the modulus of the eigenvalues. To analyze, we will use a Lyapunov-type energy method to prove stability. Define an energy-like function

$$V[k, \boldsymbol{e}(k)] = \frac{1}{2}\ \boldsymbol{e}^T(k)P^{-1}(k)\boldsymbol{e}(k) \tag{9.36}$$

Then, $\sqrt{V}$ is a weighted 2-norm of $\boldsymbol{e}(k)$ because $P(k) > 0$. Therefore, $V \geq 0$, with equality if and only if $\boldsymbol{e}(k) = 0$. We wish to show that V always gets smaller as k increases.

$$\begin{aligned}V[k+1, \boldsymbol{e}(k+1)] &= \frac{1}{2}\boldsymbol{e}^T(k+1)P^{-1}(k+1)\boldsymbol{e}(k+1) &(9.37)\\ &= \frac{1}{2}\boldsymbol{e}^T(k)[P(k) + D(k)]^{-1}\boldsymbol{e}(k) &(9.38)\end{aligned}$$

where

$$D(k) \triangleq F^{-1}(k)[\Gamma Q(k)\Gamma^T + \bar{P}(k+1)H^T(k+1)R^{-1}(k+1)H(k+1)\bar{P}(k+1)]F^{-T}(k) \tag{9.39}$$

Note $D(k) \geq 0$, which implies $[P(k) + D(k)]^{-1} < P^{-1}(k)$. Thus $V[k+1, e(k+1)] \leq V[k, e(k)]$ under suitable conditions on $Q(k)$, $R(k)$, $F(k)$, and $H(k)$ (i.e. observable, Q and R are not too big or small, controllable with respect to points of entry of process noise, or "stochastic controllability and observability").

Here observable implies that all unstable or neutrally stable subspaces of original system are observable. However, the original system need not be stable! Then, we can show that:

1. $P^{-1}(k) \geq \epsilon > 0$ for some bound ϵ.
2. $V[k+N, \boldsymbol{e}(k+N)] < \alpha V[k, \boldsymbol{e}(k)]$ for some α: $0 \leq \alpha < 1$ and some N.

Thus, V is decreasing but not because $P(k)$ is increasing.

9.7.2 Matrix Riccati Equation

Recall that

$$\bar{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k) \tag{9.40}$$

We substituted for $P(k)$ to get:

$$\bar{P}(k+1) = F(k)\{\bar{P}(k) - \bar{P}(k)H^T(k)[H(k)\bar{P}(k)H^T(k) + R(k)]^{-1}H(k)\bar{P}(k)\}F^T(k)+ \tag{9.41}$$

$$\Gamma(k)Q(k)\Gamma^T(k) \tag{9.42}$$

This gives us a dynamic model for $\bar{P}(k)$, which is nonlinear in $\bar{P}(k)$. This is called the Matrix Riccati equation (MRE). Beware: Analysis of MRE is not easy! Special case: Steady state solution for LTI systems: $F(k) = F$, $G(k) = G$, etc. If the pair (F, H) is observable, (F, Γ) is controllable, and if Q, $R > 0$ then MRE converges to a steady-state solution $\bar{P}_{ss} > 0$, which can be determined by solving the Algebraic Riccati Equation (ARE):

$$\bar{P}_{ss} = F\bar{P}_{ss} - \bar{P}_{ss}H^T[H\bar{P}_{ss}H^T + R]^{-1}H\bar{P}_{ss}F^T + \Gamma Q\Gamma^T \tag{9.43}$$

where $\bar{P}(0) > 0$. The MATLAB function for solving the ARE is:

$$[W_{ss}, \bar{P}_{ss}, P_{ss}] = dlqe(F, \Gamma, H, Q, R) \tag{9.44}$$

9.7.3 Steady-State KF Equations

The steady-state Kalman Filter equations are given by

$$\bar{\boldsymbol{x}}(k+1) = F\hat{\boldsymbol{x}}(k) + G\boldsymbol{u}(k) \tag{9.45}$$

$$\hat{\boldsymbol{x}}(k+1) = \bar{\boldsymbol{x}}(k+1) + W_{ss}[\boldsymbol{z}(k+1) - H(k+1)\bar{\boldsymbol{x}}(k+1)] \tag{9.46}$$

9.7.4 Steady-State Error Dynamics

The steady-state error dynamics are given by

$$\boldsymbol{e}(k+1) = A_{ss}HF\boldsymbol{e}(k) \tag{9.47}$$

where

$$A_{ss} \triangleq I - W_{ss}H \tag{9.48}$$

We know that the dynamics of $\boldsymbol{e}(k)$ are stable from $max(abs(eig(A_{ss}))) < 1$.
NB: Original system dynamics may not have been stable.

9.7.5 Properties of KF Innovations

The KF innovation vector is given by $\boldsymbol{\nu}(k) = \boldsymbol{z}(k) - H(k)\bar{\boldsymbol{x}}(k)$, where $\mathbb{E}[\boldsymbol{\nu}(k)] = 0$. We can show that $\mathbb{E}\left[\boldsymbol{\nu}(k)\boldsymbol{\nu}^T(j)\right] = S(k)\delta_{kj}$ (i.e. the innovation is white), where we recall that

$$S(k) \triangleq H(k)\bar{P}(k)H^T(k) + R(k) \tag{9.49}$$

The whiteness of $\boldsymbol{\nu}(k)$ follows from the whiteness of the process noise:

$$\boldsymbol{v}(k) = [\Gamma^T F^{-T} P^{-1} F^{-1} \Gamma + Q^{-1}]^{-1} \Gamma^T F^{-T} P^{-1} F^{-1} [\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)] \tag{9.50}$$

Using the Matrix Inversion Lemma,

$$\boldsymbol{v}(k) = Q(k)\Gamma^T(k)[\bar{P}^{-1}(k+1)[\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)] \tag{9.51}$$

where we recall that

$$\bar{P}(k+1) \triangleq FPF^T + \Gamma Q \Gamma^T \tag{9.52}$$

Then,

$$\begin{aligned}
0 = &-H^T R^{-1}[\boldsymbol{z}(k+1) - H\boldsymbol{x}(k+1)] + && (9.53)\\
&F^{-T}P^{-1}F^{-1}\{\boldsymbol{x}(k+1) - \Gamma Q \Gamma^T \bar{P}^{-1}(k+1)[\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)] - \bar{\boldsymbol{x}}(k+1)\} && (9.54)\\
= &-H^T R^{-1}[\boldsymbol{z}(k+1) - H\boldsymbol{x}(k+1)] + && (9.55)\\
&F^{-T}P^{-1}F^{-1}[\bar{P}\bar{P}^{-1} - \Gamma Q \Gamma^T \bar{P}^{-1}(k+1)][\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)] && (9.56)\\
= &-H^T R^{-1}[\boldsymbol{z}(k+1) - H\boldsymbol{x}(k+1)] + && (9.57)\\
&F^{-T}P^{-1}F^{-1}[\bar{P}(k+1) - \Gamma Q \Gamma^T]\bar{P}^{-1}(k+1)[\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)] && (9.58)\\
= &-H^T R^{-1}[\boldsymbol{z}(k+1) - H\boldsymbol{x}(k+1)] + I\bar{P}^{-1}(k+1)[\boldsymbol{x}(k+1) - \bar{\boldsymbol{x}}(k+1)] && (9.59)\\
& && (9.60)
\end{aligned}$$

where the last step is achieved using the definition of $\bar{P}(k+1)$. We can then solve for $\hat{\boldsymbol{x}}(k+1)$ such that

$$\hat{\boldsymbol{x}}(k+1) = [\bar{P}^{-1} + H^T R^{-1} H]^{-1}[\bar{P}^{-1}\bar{\boldsymbol{x}}(k+1) + H^T R^{-1}\boldsymbol{z}(k+1)] \tag{9.61}$$

This is just like the recursive least-squares form. With further manipulation we can get:

$$\begin{aligned}
\hat{\boldsymbol{x}}(k+1) &= \bar{\boldsymbol{x}}(k+1) + [\bar{P}^{-1} + H^T(k+1)R^{-1}(k+1)H(k+1)]^{-1} \cdot && (9.62)\\
&\quad H^T(k+1)R^{-1}(k+1)[\boldsymbol{z}(k+1) - H(k+1)\bar{\boldsymbol{x}}(k+1)] && (9.63)\\
&= \bar{\boldsymbol{x}}(k+1) + W(k+1)[\boldsymbol{z}(k+1) - H(k+1)\bar{\boldsymbol{x}}(k+1)] && (9.64)
\end{aligned}$$

Substituting this into the equation for $\boldsymbol{v}(k)$ gives

$$\begin{aligned}
\hat{\boldsymbol{v}}(k) &= \mathbb{E}\left[\boldsymbol{v}(k)|Z^{k+1}\right] = Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)[\hat{\boldsymbol{x}}(k+1) - \bar{\boldsymbol{x}}(k+1)] && (9.65)\\
&= Q(k)\Gamma^T(k)\bar{P}^{-1}(k+1)W(k+1)[\boldsymbol{z}(k+1) - H(k+1)\bar{\boldsymbol{x}}(k+1)] && (9.66)
\end{aligned}$$

This gives us more information about the process noise.

Q: How do we know if the filter is working properly?
A: If it is, then $\mathbb{E}\left[\boldsymbol{\nu}(k)\boldsymbol{\nu}^T(j)\right]$ should behave as $\delta_{kj}S(k)$. If not, then there could be a modeling or coding error, or the system may be subject to colored noise (i.e. non-white noise).

NB: This is only a necessary condition, not sufficient. That is, satisfying these conditions does not imply that the filter will work properly. ■

9.7.6 Likelihood of a Filter Model

Consider the MAP form of the KF:

$$\begin{aligned}
\min\{J[\boldsymbol{x}(k+1), \boldsymbol{\nu}(k)]\} &= J[\hat{\boldsymbol{x}}(k+1), \hat{\boldsymbol{\nu}}(k)] && (9.67)\\
&= \frac{1}{2}\boldsymbol{\nu}^T(k+1)S^{-1}(k+1)\boldsymbol{\nu}(k+1) && (9.68)
\end{aligned}$$

Use this to get the likelihood function for the correctness of the KF model.

$$P[Z^k|\text{KF mode}] = C\exp\{-J[\hat{\boldsymbol{x}}(1),\hat{\boldsymbol{\nu}}(0)]\}\cdot\exp\{-J[\hat{\boldsymbol{x}}(2),\hat{\boldsymbol{\nu}}(1)]\}\cdots\exp\{-J[\hat{\boldsymbol{x}}(k),\hat{\boldsymbol{\nu}}(k-1)]\} \tag{9.69}$$

$$= C\exp\{-\frac{1}{2}\sum_{j=1}^{k}\boldsymbol{\nu}^T(j)S^{-1}(j)\boldsymbol{\nu}(j)\} \tag{9.70}$$

Suppose we have multiple KF models with different F, Γ, Q, R, H, $\hat{\boldsymbol{x}}(0)$, $P(0)$.

Q: Which model do we trust the most?
A: The one with largest $P[Z^k|\text{KF mode}]$. This amounts to finding the KF with the minimum

$$J' \triangleq \sum_{j=1}^{k}\boldsymbol{\nu}^T(j)S^{-1}(j)\boldsymbol{\nu}(j) \tag{9.71}$$

That is, the one with the minimum weighted least-square error. This leads to the Multiple Model approach. ■

9.8 Kalman Filter Consistency

Scribe: Anamika Dubey

Recall the definition of consistency:

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}(k)\tilde{\boldsymbol{x}}(k)^T\right] = 0 \tag{9.72}$$

This doesn't hold in most causes as evidence by the fact that $P(k)$ does not go to zero in most case. The culprit is the process noise. Hence, for our purpose we decrease our requirements for consistency.

9.8.1 Consistency Definition

A state estimator (KF) is called consistent if its state estimation errors satisfy the conditions given below. This is finite-sample consistency property, which means analysis for the estimation errors based on a finite number of samples.

1. Unbiased estimator - Have zero mean

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}(k)\right] = 0 \tag{9.73}$$

2. Efficient estimator - Have covariance matrix as calculated by the filter

$$\mathbb{E}\left[\tilde{\boldsymbol{x}}(k)\tilde{\boldsymbol{x}}(k)^T\right] = P(k) = J^{-1}(k) \tag{9.74}$$

where, $J^{-1}(k)$ is the fisher information matrix

In contradiction, the parameter estimator consistency is an asymptotic (infinite size sample) property. Typically the consistency criteria of the filter are as follows:

1. The state errors should have zero mean and have covariance matrix as calculated by the filter.
2. The innovations should also have the same property as mentioned in 1.
3. The innovation should be acceptable as white.

The first criteria, which is most important one, can be tested only in simulation (Monte-carlo simulations). The last two criteria can be tested on real data (single-run / multiple-runs). In theory these properties should hold but in practice they might not. Following are the reasons for this:

- Modeling error - can be solved using tuning the filter
- Numerical error - can be solved using square root information (SRI) technique
- Programming error - solve by fixing the bugs

9.8.2 Statistical test for KF performance

Consistency of a filter can be tested in two ways. First, off-line-test - Using multiple runs (Monte-carlo simulations); Second, Real-time test - using real-time measurements by single-run or multiple-runs of the experiment (if the experiment can be repeated).

Monte Carlo Simulation based test

Consider a truth model simulation

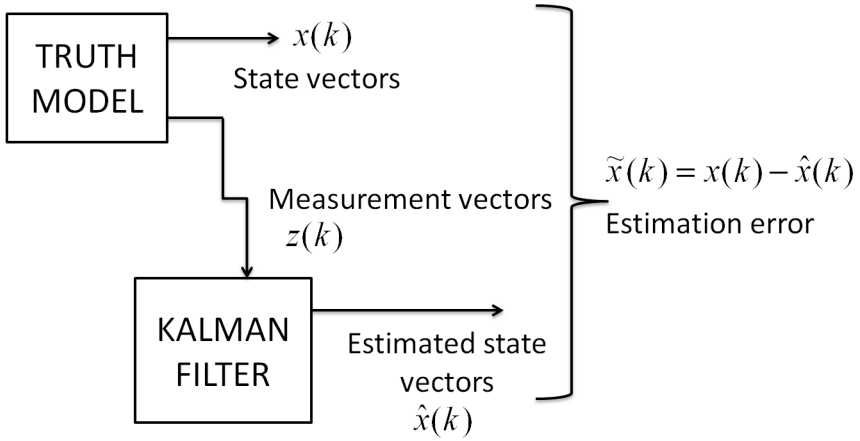


Figure 9.2: Truth model simulation for monte-carlo test

The truth model uses the dynamics equations to generate measurement $\boldsymbol{z(k)}$ and state vectors $\boldsymbol{x(k)}$. The measurement vectors are then used as input to the kalman filter (under evaluation) and estimated states $\hat{\boldsymbol{x}}(k)$ are generated.

Using the notation,

$$\tilde{\boldsymbol{x}}(k) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k) \tag{9.75}$$

define normalized estimation error squared (NEES) as

$$\tilde{\boldsymbol{\epsilon}}(k) = \tilde{\boldsymbol{x}}^T(k) P^{-1}(k) \tilde{\boldsymbol{x}}(k) \tag{9.76}$$

If KF is working properly then $\epsilon(k)$ should be distributed as chi-square with n_x degrees of freedom.

To demonstrate this further, We decompose $P^{-1}(k) = V(k)\Lambda(k)V^T(k)$ where, $V(k)V^T(k) = I$ and $\Lambda(k)$ is a diagonal matrix
let,

$$\tilde{\boldsymbol{y}}(k) = V^T(k)\tilde{\boldsymbol{x}}(k) \tag{9.77}$$

then,

$$\boldsymbol{\epsilon}(k) = \tilde{\boldsymbol{y}}^T(k)\Lambda^{-1}(k)\tilde{\boldsymbol{y}}(k) = \sum_{i=1}^{n_x} \left(\frac{\tilde{\boldsymbol{y}}_{\boldsymbol{i}}}{\sigma_{y_i}} \right)^2 \tag{9.78}$$

which is distributed as χ^2 with n_x degree of freedom, i.e. $\text{mean}(\boldsymbol{\epsilon}(k)) = n_x$

We can do N monte carlo simulations of our truth model and filter the measurements and see if average of the normalized estimation error squared (NEES) approaches n_x.

Let,

$$\bar{\boldsymbol{\epsilon}_{\boldsymbol{k}}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\epsilon}^{\boldsymbol{i}}(k) \tag{9.79}$$

where i denote i^{th} monte carlo simulation model. Then, if the filter is working properly then $\bar{\boldsymbol{\epsilon}_{\boldsymbol{k}}}$ approaches to n_x as N approached to ∞. Note that $N\bar{\boldsymbol{\epsilon}_{\boldsymbol{k}}}$ is distributed as $\chi^2_{Nn_x}$.

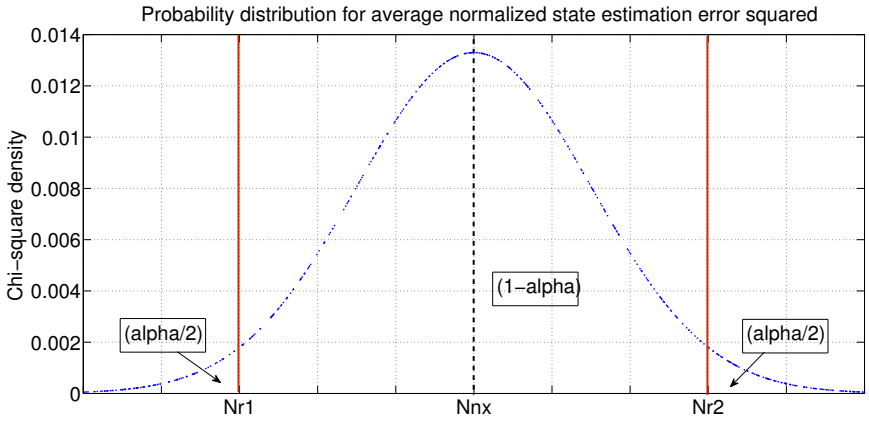


Figure 9.3: Hypothesis test for average NEES

If α is false alarm probability. Then for χ^2 hypothesis test, under H_0 hypothesis that the filter is consistent, $\bar{\boldsymbol{\epsilon}_{\boldsymbol{k}}}$ should be limited to $r_1 \leq \bar{\boldsymbol{\epsilon}_{\boldsymbol{k}}} \leq r_2$; $(1-\alpha) \times 100\%$ of the times. α s given by:

$$\int_{Nr_1}^{Nr_2} p(\xi) d\xi = (1-\alpha) \tag{9.80}$$

We usually choose r_1 and r_2 such that $\alpha = 0.01$ or $\alpha = 0.05$

In MATLAB:

$$r_1 = \frac{chi2inv(\frac{\alpha}{2}, N.n_x)}{N} \text{ and } r_2 = \frac{chi2inv(1-\frac{\alpha}{2}, N.n_x)}{N}$$

Note: If these limits are violated then something is wrong with the filter.

Real-Time (Multiple-Runs) Tests

This is test is done on filter (KF) based on the real time data for the dynamic model under evaluation. The test is applicable for the experiments that can be repeated in real world. Hence the dynamic model under evaluation should available for N real-time runs.

First compute,

$$\boldsymbol{\epsilon_\nu}(k) = \boldsymbol{\nu^T}(k) S^{-1}(k) \boldsymbol{\nu}(k) \tag{9.81}$$

If KF is working properly then this is distributed as $\chi^2_{n_z}$. We can do N, independent rum of the real experiment (if it can be repeated) and calculate :

$$\bar{\boldsymbol{\epsilon}}_{\nu_k} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\epsilon_\nu^i}(k) \tag{9.82}$$

Note that $N\bar{\boldsymbol{\epsilon}}_{\nu_k}$ is distributed as $\chi^2_{Nn_z}$.

If α is false alarm probability, then for χ^2 hypothesis test for filter consistency, $\bar{\boldsymbol{\epsilon}}_{\nu_k}$ should be limited to $r_1 \leq \bar{\boldsymbol{\epsilon}_k} \leq r_2$, $(1-\alpha) \times 100\%$ of the times. r_1 and r_2 are given by

In MATLAB:

$$r_1 = \frac{chi2inv(\frac{\alpha}{2}, N.n_z)}{N} \text{ and } r_2 = \frac{chi2inv(1-\frac{\alpha}{2}, N.n_z)}{N}$$

Note: If these limits are violated then something is wrong with the filter.

In addition to testing size of $\nu(k)$, we can test for its whiteness as well. Compute the auto-correlation function,

$$\bar{\boldsymbol{\rho}}_{lm}(k,j) = \frac{\sum_{i=1}^{N} \boldsymbol{\nu_l^i}(k) \boldsymbol{\nu_m^i}(j)}{\sqrt{\left(\sum_{i=1}^{N} \boldsymbol{\nu_l^i}(k)\right)^2 \left(\sum_{i=1}^{N} \boldsymbol{\nu_m^i}(j)\right)^2}} \tag{9.83}$$

If $k = j$ then $\bar{\boldsymbol{\rho}_{lm}}(k,k) = 1$
When $k \neq j$, then we expect $\bar{\boldsymbol{\rho}}_{lm}(k,k)$ to be small as compared to 1,

For N large enough and $k \neq j$, this statistics can be approximated as normal distribution with zero mean and $\frac{1}{N}$ as variance.

$$\mathbb{E}\left[\bar{\boldsymbol{\rho}}_{lm}(k,j)\right] = 0 \tag{9.84}$$

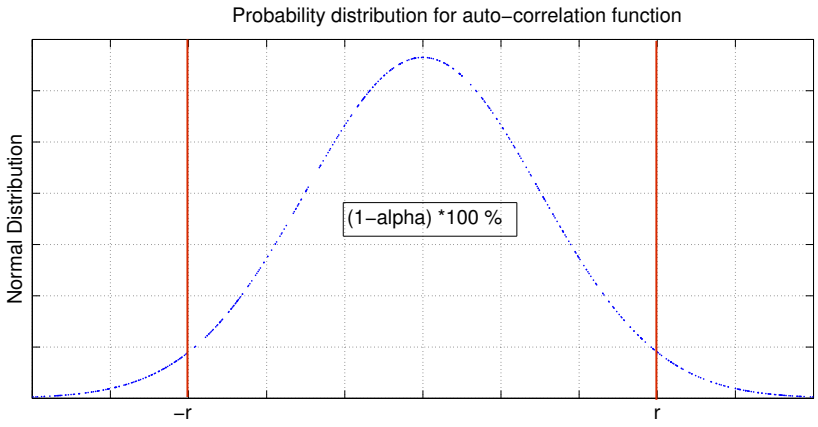


Figure 9.4: Hypothesis test for auto-correlation function

$$\mathbb{E}\left[\bar{\boldsymbol{\rho}}_{lm}(k,j)\bar{\boldsymbol{\rho}}_{lm}^{T}(k,j)\right] = \frac{1}{N} \tag{9.85}$$

hence, $-r \leq \bar{\boldsymbol{\rho}}_{lm}(k,j) \leq r$, $(1-\alpha)\times 100\%$ of the time. where, r is given as : $r = norminv\left(1-\frac{\alpha}{2}, 0, \frac{1}{\sqrt{(N)}}\right)$

Note: Typically to do the whiteness test, we look at $l = m$ and just look for k and $k+1$

Real-Time (Single-Run) Tests

All the above tests assume that N independent runs have been made. While they can be used on a single run, they might have a high variability. The question is whether one can achieve a low variability based on a single run, as a real-time implementation. This test for filter consistency is called *real-time consistency test.*

These test are based on replacing the ensemble averages by time averages based on the *ergodicity* of the innovation sequence.

Time-averaged normalized statistics is given by :

$$\bar{\boldsymbol{\epsilon}}_{\nu} = \frac{1}{\kappa}\sum_{k=1}^{\kappa} \boldsymbol{\nu}^{\boldsymbol{T}}(k) S^{-1}(k) \boldsymbol{\nu}(k) \tag{9.86}$$

then,$\bar{\boldsymbol{\epsilon}}_{\nu}\boldsymbol{\kappa}$ is distributed as $\chi^2_{\kappa n_z}$. Hence, same hypothesis test applies, with only one experiment.

Similarly whiteness test can be done. The whiteness test statistics for innovations, which are j time-step apart can be written as time-average autocorrelation.

$$\bar{\boldsymbol{\rho}}_{lm}(j) = \frac{\sum_{k=1}^{\kappa} \boldsymbol{\nu_l}(k)\boldsymbol{\nu_m}(k+j)}{\sqrt{\left(\sum_{k=1}^{\kappa} \boldsymbol{\nu_l}(k)\right)^2 \left(\sum_{k=1}^{\kappa} \boldsymbol{\nu_m}(k+j)\right)^2}} \tag{9.87}$$

This statistics is normally distributed for large κ. Furthermore it can be shown that for large κ

$$\mathbb{E}\left[\bar{\boldsymbol{\rho}}_{lm}(j)\right] = 0 \tag{9.88}$$

$$\mathbb{E}\left[\bar{\boldsymbol{\rho}}_{lm}(j)\bar{\boldsymbol{\rho}}_{lm}^{T}(j)\right] = \frac{1}{\kappa} \tag{9.89}$$

Question: What if one of the above tests fail?

Answer: Consider filter tuning.

Strategy for filter tuning

Assume $F(k), H(k), \Gamma(k), R(k)$ are correct. Since we know least about process noise, Hence, vary $Q(k)$ to pass the consistency test.

1). If $\bar{\boldsymbol{\epsilon_\nu}}(k)$ is too small then it means model fits measurement too well.

This is the case if $Q(k)$ is too big. In this case the system is making too much adjustment to $\hat{\boldsymbol{x}}(k)$ in response to each measurement.

Recall $\bar{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k)$

If $Q(k)$ is too big then so is $\bar{P}(k+1)$

Since, $S(k+1) = H(k+1)\bar{P}(k+1)H^T(k+1) + R(k+1)$ then $S^{-1}(k+1)$ will be too small, which will make $\bar{\boldsymbol{\epsilon_\nu}}(k)$ too small.

Solution is to decrease Q(k) such that $Q_{new}(k) = \dfrac{Q_{old}(k)}{10}$

2). If $\bar{\boldsymbol{\epsilon_\nu}}(k)$ is too big, then increase $Q(k)$

Chapter 10

Stochastic Control

10.1 Feedback Control from Estimated States

Recall our linear system model

$$\begin{aligned} \boldsymbol{x}(k+1) &= F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k) + \Gamma(k)\boldsymbol{v}(k) && \text{Dynamics model} \\ \boldsymbol{z}(k) &= H(k)\boldsymbol{x}(k) + w(k) && \text{Measurement model} \end{aligned}$$

and the Kalman filter equations

$$\begin{aligned} \bar{\boldsymbol{x}}(k+1) &= F(k)\hat{\boldsymbol{x}}(k) + G(k)\boldsymbol{u}(k) && \textit{A priori} \text{ state est.} \\ \bar{P}(k+1) &= F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k) && \textit{A priori} \text{ state error cov.} \\ \boldsymbol{\nu}(k+1) &= \boldsymbol{z}(k+1) - H(k+1)\bar{\boldsymbol{x}}(k+1) && \text{Innovation} \\ S(k+1) &= H(k+1)\bar{P}(k+1)H^T(k+1) + R(k+1) && \text{Innovation covariance} \\ W(k+1) &= \bar{P}(k+1)H^T(k+1)S^{-1}(k+1) && \text{Kalman gain matrix} \\ \hat{\boldsymbol{x}}(k+1) &= \bar{\boldsymbol{x}}(k+1) + W(k+1)\boldsymbol{\nu}(k+1) && \textit{A posteriori} \text{ state est.} \\ P(k+1) &= \bar{P}(k+1) - W(k+1)S(k+1)W^T(k+1) && \textit{A posteriori} \text{ state error cov.} \end{aligned}$$

Suppose we wish to perform feedback control with control law $\boldsymbol{u}(k) = -C(k)\boldsymbol{x}(k)$ (full state feedback), for which the closed-loop dynamics model is

$$\boldsymbol{x}(k+1) = [F(k) - G(k)C(k)]\boldsymbol{x}(k) + \Gamma(k)\boldsymbol{v}(k)$$

Linear system theory tells us that if the pair (F, G) is controllable then we can design a stabilizing controller (stable closed-loop system). However, it may be expensive, impractical, or impossible to measure all of the states. What if we use $\hat{\boldsymbol{x}}(k)$ instead?

Q: Can we substitute $\hat{\boldsymbol{x}}(k)$ for $x(k)$ and still get a stable CL system?
A: ■

Let $\boldsymbol{u}(k) = -C(k)\hat{\boldsymbol{x}}(k)$ so that the system model becomes

$$\begin{aligned} \boldsymbol{x}(k+1) &= F(k)\boldsymbol{x}(k) - G(k)C(k)\hat{\boldsymbol{x}}(k) + \Gamma(k)\boldsymbol{v}(k) \\ \hat{\boldsymbol{x}}(k+1) &= [I - W(k+1)H(k+1)][F(k)\hat{\boldsymbol{x}}(k) - G(k)C(k)\hat{\boldsymbol{x}}(k)] \\ &\quad + W(k+1)\underbrace{\{H(k+1)[F(k)\boldsymbol{x}(k) - G(k)C(k)\hat{\boldsymbol{x}}(k) + \Gamma(k)\boldsymbol{v}(k)] + \boldsymbol{w}(k+1)\}}_{\boldsymbol{z}(k+1)} \end{aligned}$$

(The second equation is the full KF state update written as one step.)

Trick: Perform a change of coordinates:

$$\begin{bmatrix}\boldsymbol{x}(k)\\ \hat{\boldsymbol{x}}(k)\end{bmatrix} \to \begin{bmatrix}\boldsymbol{x}(k)\\ \boldsymbol{e}(k)\end{bmatrix} = \begin{bmatrix}\boldsymbol{x}(k)\\ \boldsymbol{x}(k)-\hat{\boldsymbol{x}}(k)\end{bmatrix}$$

With this change of coordinates, the dynamics of the overall controller-estimator system are given by

$$\begin{aligned}\boldsymbol{x}(k+1) &= [F(k)-G(k)C(k)]\boldsymbol{x}(k)+G(k)C(k)\boldsymbol{e}(k)+\Gamma(k)\boldsymbol{v}(k)\\ \boldsymbol{e}(k+1) &= [I-W(k+1)H(k+1)]F(k)\boldsymbol{e}(k)\\ &\quad +[I-W(k+1)H(k+1)]\Gamma(k)\boldsymbol{v}(k)-W(k+1)\boldsymbol{w}(k+1)\end{aligned}$$

Is this system stable? For analysis, ignore exogenous inputs. Also, recognize that the error dynamics do not depend on the state. (This is unfortunately not true for nonlinear systems.) We already showed that $\boldsymbol{e}(k)\to 0$ as $k\to\infty$ for any linear KF that satisfies the stochastic observability, control, etc. Therefore, after a long time $\boldsymbol{e}(k)\approx 0$ and the state dynamics obey

$$\boldsymbol{x}(k+1) = [F(k)-G(k)C(k)]\boldsymbol{x}(k) \tag{10.1}$$

which is stable by our choice of $C(k)$.

Thus, we can design a full-state feedback control law and the KF *separately*—this is the *separation principle*. When connected, the system will be stable. The combined system ends up with the properties of the two independent systems (union of eigenvalues of $[F(k)-G(k)C(k)]$ and $[I-W(k+1)H(k+1)]F(k)]$). In practice, the poles of the estimator should be to the left (i.e., faster) than those of the controlled plant or process.

Note: If the estimator has modeling errors (e.g., F or G not perfectly known), then the system may not decouple!

10.2 Duality of Estimation and Control

LQ (linear quadratic) control: An optimal control technique where cost to be minimized is quadratic in $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$.

System dynamics:

$$\boldsymbol{x}(k+1) = F(k)\boldsymbol{x}(k)+G(k)\boldsymbol{u}(k)$$

The initial state $\boldsymbol{x}(0)$ and initial control input $\boldsymbol{u}(0)$ are assumed to be known. A common measure of performance is the quadratic cost

$$\begin{aligned}J(N) &= [\boldsymbol{x}(N)-\bar{\boldsymbol{x}}(N)]^T P_c(N)[\boldsymbol{x}(N)-\bar{\boldsymbol{x}}(N)]\\ &\quad +\sum_{k=0}^{N-1}[\boldsymbol{x}(k)-\bar{\boldsymbol{x}}(k)]^T Q_c(k)[\boldsymbol{x}(k)-\bar{\boldsymbol{x}}(k)]+[\boldsymbol{u}(k)-\bar{\boldsymbol{u}}(k)]^T R_c(k)[\boldsymbol{u}(k)-\bar{\boldsymbol{u}}(k)]\end{aligned}$$

where the bars indicate nominal (i.e., desired) values for the associated quantities, $P_c(N)>0$ reflects the penalty imposed on the terminal state, $Q_c(k)\geq 0$ is the penalty on the intermediate states, and $R_c(k)>0$ is the penalty on the control input.

Note the similarity in form between this cost and the estimation cost given in the MAP-based KF

derivation (given as handout; also prob. 1 in problem set 5):

$$\begin{aligned} J[\boldsymbol{x}(k), \boldsymbol{v}(k), \boldsymbol{x}(k+1), k] = & \tfrac{1}{2} \left[\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k)\right]^T P^{-1}(k) \left[\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k)\right] \\ & + \tfrac{1}{2} \boldsymbol{v}^T(k) Q^{-1}(k) \boldsymbol{v}(k) \\ & + \tfrac{1}{2} \left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}(k+1)\right]^T R^{-1}(k+1) \left[\boldsymbol{z}(k+1) - H(k+1)\boldsymbol{x}(k+1)\right] \end{aligned}$$

In the estimation problem, we penalize deviation of $\boldsymbol{x}(k)$ from our estimate $\hat{\boldsymbol{x}}(k)$ with a weighting $P^{-1}(k)$, processing noise with a weighting $Q^{-1}(k)$, and measurement residuals with a weighting $R^{-1}(k)$.

As it turns out, deterministic LQ control and linear MMSE estimation (with additive noise) are *dual problems*:

- The optimal performace of linear MMSE estimation starting at $k = 0$ is given at $k > 0$ by $P(k)$. Similarly, the optimal performance of the LQ control problem starting at $k < k_1$ and running up to $k = k_1$ is determined by an *optimal cost matrix* $K(k)$.
- The equations describing the dynamics of $P(k)$ and $K(k)$ are identical (both are matrix Riccati equations) except that $P(k)$ propagates forward from k_0 whereas $K(k)$ propagates backwards from k_1.
- The filter gain $W(k)$ has a counterpart of exactly the same form in the control problem called the control feedback gain $W_c(k)$.
- The dynamics of the optimal state estimate are the *adjoint* of the dynamics of the closed-loop control system under the optimal control feedback.

LQG(linear quadratic Gaussian) control: LQ control with a linear quadratic estimator (i.e., KF) supplying the state estimates for feedback.

10.3 Correlated Process and Measurement Noise

Scribe: Henri Kjellberg

Base assumptions of the Kalman Filter:

1. Process noise is white
2. Measurement noise is white
3. The two noises are uncorrelated

Q: What if one or more of these assumptions is violated?
A: Let's back up and consider our noise sources in the frequency domain. Consider continuous time noise. Usually by "white" we mean specifically that: $S_{\tilde{v}\tilde{v}}(f)$ = power spectrum of $\tilde{v}$=Const $= V$. By the Weiner-Khonchin theorem, take the inverse Fourier transform to recover the auto-correlation function:

$$E\left[\tilde{v}(t)\,\tilde{v}(t+\tau)\right] = R_{\tilde{\nu}\tilde{\nu}}\tau = \mathcal{F}^{-1}\left[\left(S_{\tilde{\nu}\tilde{\nu}}(f)\right)\right] = V\delta(\tau) \tag{10.2}$$

Spectrally, flat $\rightleftarrows$ uncorrelated in time ■
This implies that a nonuniform power spectra leads to auto-corelated noise:

Strategy: the power spectrum of the auto-correlated noise can be approximated as closely as desired by the output of a linear subsystem driven by white noise.

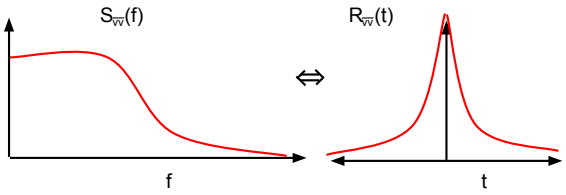


Figure 10.1: An example of a low-pass process

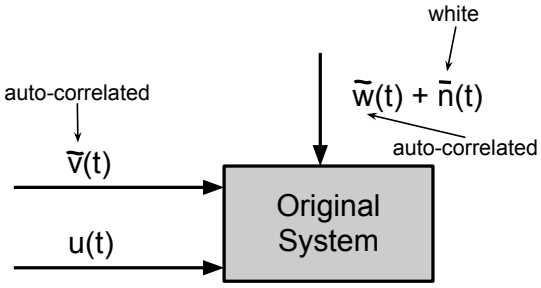


Figure 10.2: Power spectrum approximation linear system

$$\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t) + D(t)\,\tilde{v}(t) \tag{10.3}$$

$$z(t) = C(t)\,x(t) + \tilde{w}(t) + \bar{n}(t) \tag{10.4}$$

Assume $E[\tilde{v}(t)] = E[\tilde{w}(t)] = E[\bar{n}(t)] = 0$

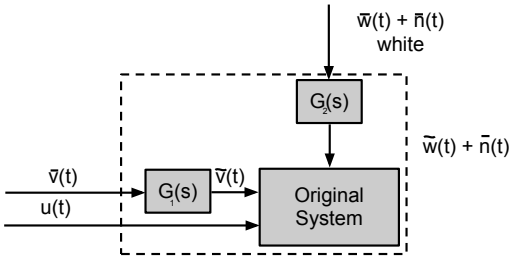


Figure 10.3: Augmented system

Let $\eta(t) = \begin{bmatrix} \tilde{v}(t) \\ \tilde{w}(t) + n(t) \end{bmatrix}$

The shaping or pre-whitening filters can be implemented in state space as follows:

$$\dot{x}_\eta(t) = A_\eta x_\eta(t) + B_\eta \bar{n}(t) \tag{10.5}$$

$$\eta(t) = C_\eta x_\eta(t) + D_\eta \bar{\eta}(t) \tag{10.6}$$

where

$$\bar{\eta}(t) = \begin{bmatrix} \bar{v}(t) \\ \bar{w}(t) \\ \bar{n}(t) \end{bmatrix} \tag{10.7}$$

The output of the shaping system can be used to drive the original systems. The augmented dynamics become:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_\eta(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A(t) & D(t) \begin{bmatrix} I \\ 0 \end{bmatrix} C_\eta \\ 0 & A_\eta \end{bmatrix}}_{\text{new A matrtix}} \begin{bmatrix} x(t) \\ x_\eta(t) \end{bmatrix} + \begin{bmatrix} B(t) \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} D(t) \begin{bmatrix} I \\ 0 \end{bmatrix} D_\eta \\ B_\eta \end{bmatrix} \bar{\eta}(t) \tag{10.8}$$

$$z(t) = \begin{bmatrix} C(t) & \begin{bmatrix} 0 & I \end{bmatrix} C_\eta \end{bmatrix} \begin{bmatrix} x(t) \\ x_\eta(t) \end{bmatrix} + \begin{bmatrix} 0 & I \end{bmatrix} D_\eta \bar{\eta}(t) \tag{10.9}$$

Q: How do we develop the shaping filters $G_1(s)$ and $G_2(s)$?
A: Estimate the power spectrum as a rational model (e.g., $S_{\tilde{v}\tilde{v}}(f) = \frac{N(f)}{D(f)}$) and derive the transfer function from $S_{\tilde{v}\tilde{v}}(f)$ and $S_{\tilde{w}\tilde{w}}(f)$ via spectral factorization. ■
(See the standard texts on power spectrum parametrization).

One can often build up the desired power spectrum as a combination of several "building blocks" - prototypical Gauss-Markov processes. The above covers only auto-correlated noise. See Bar Shalom 8.3 for cross-correlated measurement and process noise.

10.3.1 Aside: Realization Problem

We can recall the **Realization Problem**, where one attempts to go from an input-output relationship governed by a transfer function such as

$$\hat{y} = \hat{G}(s)\,\hat{u}(s) \tag{10.10}$$

into the state space model

$$\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t) \tag{10.11}$$
$$y(t) = C(t)\,x(t) + D(t)\,u(t) \tag{10.12}$$

For strictly proper transfer functions (the degree of the numerator is less than the degree of the denominator) one can create a controllable canonical form or an observable canonical form. We do this by exposing the coefficients of the numerator and denominator of the transfer function. As an example:

Given a transfer function:

$$G(s) = \frac{n_1 s^2 + n_2 s + n_3}{s^3 + d_1 s^2 + d_2 s + d_3} \tag{10.13}$$

A state space model that is guaranteed to be controllable will take the form:

$$\dot{x}(t) = \begin{bmatrix} -d_1 & -d_2 & -d_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(t) \tag{10.14}$$

$$y(t) = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} x(t) \tag{10.15}$$

Related MATLAB functions to investigate are: tf, ss, zpk, frd, ssdata, tf2ss.

10.3.2 Aside: Spectral Factorization

Spectral factorization involves taking a transfer function such as the one shown in Bar Shalom on p. 67:

$$S_{\tilde{v}\tilde{v}}(\omega) = S_0 \frac{1}{a^2 + \omega^2} \tag{10.16}$$

And splitting it into two functions, one part with all the Right Hand Plane (RHP) roots and the other including all the Left Hand Plane (LHP) roots.

$$S_{\tilde{v}\tilde{v}}(\omega) = \frac{1}{a + j\omega} S_0 \frac{1}{a - j\omega} \tag{10.17}$$

$$H(\omega) = \frac{1}{a + j\omega} \tag{10.18}$$

Is identified as the causal transfer function.

Chapter 11

Information Filtering/SRIF

Recall that the *a posteriori* state estimate error covariance matrix $P(k+1)$ can be defined in terms of the following (alternative) update formula:

$$P(k+1) = \left[\bar{P}^{-1}(k+1) + \underbrace{H^{\mathrm{T}}(k+1)R^{-1}(k+1)H(k+1)}_{\text{matrix squaring operation}} \right]^{-1}$$

The problem with this traditional approach to updating the error covariance matrix is that the matrix squaring operation is a bad idea numerically—it squares condition number of the H matrices[1]. Note that $\bar{P}(k+1)$ may also be ill-conditioned for the same reason; so may $R(k+1)$.

Let's write $P(k+1)$ in terms of another update formula:

$$P(k+1) = \bar{P}(k+1) - W(k+1)S^{-1}(k+1)W^{\mathrm{T}}(k+1)$$

Here, infinite numerical precision ensures $P(k+1) > 0$ but roundoff errors (e.g., from non-infinite numerical precision) can make $P(k+1)$ indefinite (not positive definite) or asymmetric.

One can use Joseph's form of $P(k+1)$ update to ensure $P(k+1) = P^{\mathrm{T}}(k+1) > 0$, but this doesn't help the accuracy of $P(k+1)$. A deterioration in accuracy of $P(k+1)$ can lead to garbage results.

To address these problems, Bar Shalom introduces the square root covariance filter, which keeps track of the square root of the covariance matrix. But this requires updating a Cholesky factorization. As it turns out, it's easier to work with the square root of the information matrix (the inverse of the covariance matrix). The resulting filter, the square-root information filter (SRIF), is more efficient than the square-root covariance filter when $n_z > n_x, n_v$; otherwise the square-root covariance filter is faster.

As a prelude the the SRIF, we'll first look at the traditional (non-square-root) information filter.

[1] Note that the notion of condition number applies to non-square matrices such as $H(k+1)$—the condition number is just the ratio of the largest to the smallest singular value.

11.0.1 Information Filter

Strategy: Instead of keeping track of $\hat{x}(k+1)$, $P(k+1)$, $\bar{x}(k+1)$, $\bar{P}(k+1)$, we keep track of

$$\begin{aligned}\bar{y}(k) &= \bar{P}^{-1}(k)\bar{x}(k)\\ \hat{y}(k) &= P^{-1}(k)\hat{x}(k)\\ \bar{\mathscr{I}}(k) &= \bar{P}^{-1}(k)\\ \mathscr{I}(k) &= P^{-1}(k)\end{aligned}$$

where $\mathscr{I}(k)$ is known as the information matrix and is equal to the inverse of the covariance matrix $P(k)$; $\mathscr{I}(k)$ is analagous to the Fisher information matrix—the bigger it is, the more certain the associated estimates.

We can substitute these definitions into the Kalman filter to get the information filter. Let

$$A(k) = F^{-\mathrm{T}}(k)\mathscr{I}(k)F^{-1}(k)$$

After much algebra, including the matrix inversion lemma, we arrive at the following propagation and update equations.

Propagation

$$\bar{y}(k+1) = \left\{I - A(k)\Gamma(k)\left[\Gamma^T(k)A(k)\Gamma(k) + Q^{-1}(k)\right]^{-1}\Gamma^T(k)\right\}\left[F^{-\mathrm{T}}(k)\hat{y}(k) + A(k)G(k)u(k)\right]$$

$$\bar{\mathscr{I}}(k+1) = A(k)\underbrace{-A(k)\Gamma(k)\left[\Gamma^T(k)A(k)\Gamma(k) + Q^{-1}(k)\right]^{-1}\Gamma^T(k)A(k)}_{\text{Information is decreasing due to process noise}}$$

Process noise decreases the information during the propagation step. This is similar to a hole in a metaphorical information bucket. If $Q(k) = 0$, i.e., there is no process noise, then there is no information leaking out and

$$\bar{\mathscr{I}}(k+1) = A(k) = F^{-\mathrm{T}}(k)\mathscr{I}(k)F^{-1}(k)$$

Update

$$\hat{y}(k+1) = \bar{y}(k+1) + H^{\mathrm{T}}(k+1)R^{-1}(k+1)z(k+1)$$

$$\mathscr{I}(k+1) = \bar{\mathscr{I}}(k+1) + \underbrace{H^{\mathrm{T}}(k+1)R^{-1}(k+1)H(k+1)}_{\text{measurements add information}}$$

Here the information is increasing due to new measurements. This is similar to an information pipe filling the metaphorical information bucket.

We can recover $\hat{x}$ and P by

$$\begin{aligned}\hat{x}(k+1) &= \mathscr{I}^{-1}(k+1)\hat{y}(k+1)\\ P(k+1) &= \mathscr{I}^{-1}(k+1)\end{aligned}$$

Remarks.

1. The information filter is more efficient than Kalman Filter if $n_z > n_x$, n_v and if $R(k)$ is diagonal, but usually this is not the case.

2. For linear systems we can pick the initial state estimate arbitrarily so long as we set $\mathscr{I}(0) = 0$. This represents the diffuse prior, i.e., it indicates we no idea of our initial state. A diffuse initial prior cannot be as easily done with the regular Kalman Filter. We could set the error covariance matrix to infinity, but finite numerical precision limits our ability to do so in real systems. Note that we cannot compute $\hat{x}(k) = \mathscr{I}^{-1}(k)\hat{y}(k)$ until $\mathscr{I}(k)$ becomes invertible. If the system is observable, then $\mathscr{I}(k)$ will eventually become invertible. Waiting for $\mathscr{I}(k)$ to become invertible is like waiting large enough k so that H^k has full column rank in the batch initialization problem.

3. This form of the information filter still involves squaring in the $H^T R^{-1} H$ terms.

11.0.2 Square Root Information Filter

The square root information filter (SRIF) is a particular implementation of the information filter that involves no squaring of terms.

Define

$$\begin{aligned} R_{xx}^T(k)R_{xx}(k) &= \mathscr{I}(k) \\ \bar{R}_{xx}^T(k)\bar{R}_{xx}(k) &= \bar{\mathscr{I}}(k) \\ z_x(k) &= R_{xx}(k)\hat{x}(k) \\ \bar{z}_x(k) &= \bar{R}_{xx}(k)\bar{x}(k) \end{aligned}$$

where $R_{xx}(k)$ and $\bar{R}_{kk}(k)$ are the Cholesky factorizations of $\mathscr{I}(k)$ and $\bar{\mathscr{I}}(k)$, respectively. Also let

$$\begin{aligned} R_a^T(k)R_a(k) &= R(k) \\ H_a(k) &= R_a^{-T}(k)H(k) \\ z_a(k) &= R_a^{-T}(k)z(k) \\ w_a(k) &= R_a^{-T}(k)w(k) \end{aligned}$$

so that the transformed measurement equation becomes

$$z_a(k) = H_a(k)x(k) + w_a(k)$$

where

$$\begin{aligned} E\left[w_a(k)\right] &= 0 \\ E\left[w_a(k)w_a^T(j)\right] &= I\delta_{kj} \end{aligned}$$

The dynamics model remains unchanged

$$x(k+1) = F(k)x(k) + G(k)u(k) + \Gamma(k)v(k)$$

$$\begin{aligned} \hat{v}(k) &\triangleq E\left[v(k)\right] = 0 \\ E\left[v(k)v^T(j)\right] &= \delta_{kj}Q(k) \end{aligned}$$

but Cholesky factorize the $Q(k)$ matrix:

$$R_{vv}^T(k)R_{vv}(k) = Q^{-1}(k)$$

Note that this factorization-and-inverse operation is most accurate numerically when we perform the Cholesky factorization first, followed by the inverse. In Matlab:

$$R_{vv}(k) = [\texttt{inv}(\texttt{chol}(Q(k)))]^{\mathrm{T}}$$

With the previous definitions, we can write out the following so-called square-root information (SRI) equations (also known as data equations):

$$z_x(k) = R_{xx}(k)x(k) + w_x(k), \qquad w_x(k) \sim (0, I) \tag{11.1}$$
$$z_v(k) = 0 = R_{vv}(k)v(k) + w_v(k), \qquad w_v(k) \sim (0, I) \tag{11.2}$$
$$\mathbb{E}\left[w_x(k)w_v^T(k)\right] = 0 \tag{11.3}$$

One can think of these SRI equations as storing, or "encoding," the state and process noise estimates and their covariances. We can recover our estimates $\hat{x}(k)$ and $\hat{v}(k)$, and their corresponding covariance matrices $P(k)$ and $Q(k)$, from the SRI equations so long as R_{xx} is invertible.[2] The state estimate $\hat{x}(k)$ can be recovered from (11.1) as follows

$$\begin{aligned}
x(k) &= R_{xx}^{-1}(k)\left[z_x(k) - w_x(k)\right] \\
\hat{x}(k) &= E\left[x(k)|k\right] \\
&= R_{xx}^{-1}(k)\underbrace{E\left[z_x(k)|k\right]}_{z_x(k)} - R_{xx}^{-1}(k)\underbrace{E\left[w_x(k)|k\right]}_{0} \\
&= R_{xx}^{-1}(k)z_x(k)
\end{aligned}$$

$P(k)$ can be recovered from the same SRI equation by

$$\begin{aligned}
\tilde{x}(k) &= x(k) - \hat{x}(k) = -R_{xx}^{-1}(k)w_x(k) \\
P(k) &= E\left[\tilde{x}(k)\tilde{x}^T(k)|k\right] \\
&= R_{xx}^{-1}(k)\underbrace{E\left[w_x(k)w_x^T(k)\right]}_{I}R_{xx}^{-T}(k) \\
&= R_{xx}^{-1}(k)R_{xx}^{-T}(k) \\
&= \mathscr{I}^{-1}(k)
\end{aligned}$$

Likewise, $\hat{v}(k)$ and $Q(k)$ can be recovered from (11.2) by

$$\begin{aligned}
&\hat{v}(k) = E\left[v(k)|k\right] = R_{vv}^{-1}(k)\mathbb{E}\left[z_v(k)|k\right] - R_{vv}^{-1}(k)\mathbb{E}\left[w_v(k)|k\right] = 0 \\
&E\left[(v(k) - \hat{v}(k))(v(k) - \hat{v}(k))^T|k\right] = Q(k)
\end{aligned}$$

Propagation Step and Measurement Update

The SRIF is nothing more than a way of performing a propagation step and a measurement update on the SRI equations (11.1) and (11.2). For our derivation, we'll adopt the MAP approach, in which we seek to minimize the negative natural log of the *a posteriori* conditional probability density function. This amounts to minimizing the cost function

$$\begin{aligned}
J_a\left[x(k), v(k), x(k+1), k\right] &= -\log\left\{p[x(k+1), v(k)|Z^{k+1}]\right\} \\
&= \tfrac{1}{2}[x(k) - \hat{x}(k)]^T P^{-1}(k)[...] + \tfrac{1}{2}v^T(k)Q^{-1}(k)v(k) \\
&\quad + \tfrac{1}{2}[z(k+1) - H(k+1)x(k+1)]^T R^{-1}(k+1)[...]
\end{aligned}$$

[2] If $R_{xx}(k)$ is not invertible then the system is not observable from the data through time k and the *a priori* info at time 0. Note that $R_{xx}(k)$ is upper triangular.

After substitution of the previously-defined square-root quantities, we have the following equivalent cost:

$$J_a\left[x(k), v(k), x(k+1), k\right] = \tfrac{1}{2}\underbrace{\left\|R_{xx}(k)x(k) - z_x(k)\right\|^2}_{\text{a priori x}(k)} + \tfrac{1}{2}\underbrace{\left\|R_{vv}(k)v(k) - z_v(k)\right\|^2}_{\text{a priori v}(k)} + \tfrac{1}{2}\underbrace{\left\|H_a(k+1)x(k+1) - z_a(k+1)\right\|^2}_{\text{measurement at k+1}}$$

Notice how the first and second terms in the cost function now have the same form as the third term, the measurement cost term. In other words, $z_x(k)$ acts as another set of measurements on $x(k)$ with measurement equation (11.1), and $z_v(k)$ acts as a set of measurements on $v(k)$ with measurement equation (11.2). This is a key insight that the SRIF makes clear: prior estimates of $x(k)$ and $v(k)$ can be written in the form of measurements on $x(k)$ and $v(k)$ and can thus be intuitively packaged together with the true measurements in the cost function.

Our task is to minimize J_a subject to the dynamics model, which relates $x(k)$, $v(k)$, and $x(k+1)$. We first solve the dynamics model for $x(k)$ in terms of $v(k)$ and $x(k+1)$:

$$x(k) = F^{-1}(k)\left[x(k+1) - G(k)u(k) - \Gamma(k)v(k)\right]$$

We then simultaneously eliminate $x(k)$ and enforce the dynamics constraint by substituting $x(k)$ into J_a. We'll call the resulting equivalent cost function J_b:

$$\begin{aligned} &J_b\left[v(k), x(k+1), k\right] \\ &= \tfrac{1}{2}\left\|\begin{bmatrix} R_{vv}(k) & \mathbf{0} \\ -R_{xx}(k)F^{-1}(k)\Gamma(k) & R_{xx}(k)F^{-1}(k)\end{bmatrix}\begin{bmatrix} v(k) \\ x(k+1)\end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ z_x(k) + R_{xx}(k)F^{-1}(k)G(k)u(k)\end{bmatrix}\right\|^2 \\ &\quad + \tfrac{1}{2}\left\|H_a(k+1)x(k+1) - z_a(k+1)\right\|^2 \end{aligned}$$

In packaging the equation above, we invoked the following identity:

$$\|a\|^2 + \|b\|^2 = \left\|\begin{bmatrix} a \\ b\end{bmatrix}\right\|^2$$

Recall that $\|Tv\|^2 = \|v\|^2$ for T orthonormal. Let $T_a(k) = Q_a^T(k)$, where $Q_a(k)$ is the orthonormal matrix that results from QR factorization of the big block matrix in the equation above. (T_a is also orthonormal.) Multiplying the insides of first term by $T_a(k)$ and rewriting yields

$$\begin{aligned} J_b\left[v(k), x(k+1), k\right] = \quad & \tfrac{1}{2}\left\|\begin{bmatrix} \overline{R}_{vv}(k) & \overline{R}_{vx}(k+1) \\ \mathbf{0} & \overline{R}_{xx}(k+1)\end{bmatrix}\begin{bmatrix} v(k) \\ x(k+1)\end{bmatrix} - \begin{bmatrix} \overline{z}_v(k) \\ \overline{z}_x(k+1)\end{bmatrix}\right\|^2 \\ & + \tfrac{1}{2}\left\|H_a(k+1)x(k+1) - z_a(k+1)\right\|^2 \end{aligned}$$

This was the propagation step! Why? Because we can now isolate the *a priori* SRI equation for $x(k+1)$. Note that, just as we took a set of SRI equations and packed them into a cost function, so can we take the cost function and unpack it into a set of SRI equations. Unpacking the first term into its square root information equations, we obtain

1. The *a posteriori* SRI equation for $v(k)$ as a function of $x(k+1)$:

$$\overline{z}_v(k) = \overline{R}_{vv}(k)v(k) + \overline{R}_{vx}(k+1)x(k+1) + \overline{w}_v(k), \quad \overline{w}_v(k) \sim (0, I)$$

   This equation is a by-product of the filtering process. We ignore it in filtering (since it is non-causal and filtering implies causality), but we will make use of it in smoothing.

2. The *a priori* SRI equation for the state at $k+1$:

$$\overline{z}_x(k+1) = \overline{R}_{xx}(k+1)x(k+1) + \overline{w}_x(k+1), \qquad \overline{w}_x(k) \sim (0, I)$$

As a next step, let us minimize J_b with respect to $v(k)$. The necessary condition

$$0 = \left[\frac{\partial J_b}{\partial v(k)}\right]^{\mathrm{T}} = \underbrace{\overline{R}_{vv}^{T}(k)}_{\text{non-singular}} \left[\overline{R}_{vv}(k)v(k) + \overline{R}_{vx}(k+1)x(k+1) - \overline{z}_v(k)\right]$$

requires

$$v(k) = \overline{R}_{vv}^{-1}(k)\left[\bar{z}_v(k) - \overline{R}_{vx}(k+1)x(k+1)\right]$$

Substituting this into the cost function and stacking the remaining terms we get

$$J_c\left[x(k+1), k+1\right] = \tfrac{1}{2}\left\| \underbrace{\begin{bmatrix}\overline{R}_{xx}(k+1)\\ H_a(k+1)\end{bmatrix}}_{A} x(k+1) - \begin{bmatrix}\overline{z}_x(k+1)\\ z_a(k+1)\end{bmatrix} \right\|^2 \tag{11.4}$$

If A were square and non-singular we could just take its inverse to compute $\hat{x}(k+1)$ the filter's best estimate of $x(k+1)$. But because A is not square, we instead transform the insides of the norm by QR factorizing, as before. The transformation has the effect of decoupling the cost function into a component that depends on $x(k+1)$ and one that does not:

$$J_c[x(k+1), k+1] = \left\| \begin{bmatrix}R_{xx}(k+1)\\ 0\end{bmatrix} x(k+1) - \begin{bmatrix}z_x(k+1)\\ z_r(k+1)\end{bmatrix} \right\|^2$$

This was the update step! Why? Because we can now isolate the *a posteriori* SRI equation for $x(k+1)$. The lack of bars above the terms indicates that we have gone from *a priori* to *a posteriori*. Unstack to get

$$J_c\left[x(k+1), k+1\right] = \tfrac{1}{2}\left\|R_{xx}(k+1)x(k+1) - z_x(k+1)\right\|^2 + \tfrac{1}{2}\left\|z_r(k+1)\right\|^2$$

Now unpack the implicit SRI equations from this cost function to get

1. The *a posteriori* SRI equation for for the state at $k+1$:

$$z_x(k+1) = R_{xx}(k+1)x(k+1) + w_x(k+1), \qquad w_x(k+1) \sim (0, I)$$

2. The residual error equation:

$$z_r(k+1) = w_r(k+1), \qquad w_r(k+1) \sim (0, I)$$

Aside: An inside view of the measurement update step
Unpack 11.4 into its implicit square root information equations:

$$\overline{z}_x(k+1) = \overline{R}_{xx}(k+1)x(k+1) + \overline{w}_x(k+1)$$

$$z_a(k+1) = H_a(k+1)x(k+1) + w_a(k+1)$$

Let $T_a(k+1) = Q_a^T(k+1)$, where $Q_a(k+1)$ is the orthonormal matrix that results from QR factorization of A. Multiplying terms by $T_a(k)$ yields

$$\begin{bmatrix} R_{xx}(k+1) \\ 0 \end{bmatrix} = T_a(k+1) \begin{bmatrix} \bar{R}_{xx}(k+1) \\ H_a(k+1) \end{bmatrix}$$

$$\begin{bmatrix} z_x(k+1) \\ z_r(k+1) \end{bmatrix} = T_a(k+1) \begin{bmatrix} \bar{z}_x(k+1) \\ z_a(k+1) \end{bmatrix}$$

$$\begin{bmatrix} w_x(k+1) \\ w_r(k+1) \end{bmatrix} = T_a(k+1) \begin{bmatrix} \bar{w}_x(k+1) \\ w_a(k+1) \end{bmatrix}$$

Because $T_a(k+1)$ is orthonormal, $w_x(k+1)$ and $w_r(k+1)$ retain the same distribution as $\bar{w}_x(k+1)$ and $w_a(k+1)$; i.e.,

$$w_x(k+1) \sim (0, I), \qquad w_r(k+1) \sim (0, I)$$

We can now minimize J_c by inspection:

$$\hat{x}(k+1) = R_{xx}^{-1}(k+1)z_x(k+1)$$

The minimum value of J_c is

$$J_c[\hat{x}(k+1), k+1] = \tfrac{1}{2} \left\| z_r(k+1) \right\|^2$$

It can be shown that

$$\left\| z_r(k+1) \right\|^2 = z_r^T(k+1)z_r(k+1) = \nu^T(k+1)S(k+1)\nu(k+1) \sim \chi^2_{n_z}$$

where $\nu(k+1)$ and $S(k+1)$ are the innovation and its covariance matrix, which were defined earlier.

Remarks.

1. The SRIF avoids matrix squaring. It involves only QR factorization and inversion of R matrices when necessary. This makes it numerically robust.
2. $P(k)$is guaranteed to be symmetric and positive definite because $P(k) = R_{xx}^{-1}(k)R_{xx}^{-T}(k)$
3. For the SRIF presented here, $F(k)$ must be invertible; if not, an SRIF implementation is possible but more complicated.

Chapter 12

Smoothing

12.1 Overview

Problem: Estimate $x(k)$ based on Z^j with $j > k$.

3 Classic Types:

1. Fixed point smoothing: Estimate $x(k)$ for a fixed k based on Z^j with j increasing.
2. Fixed lag smoothing: Estimate $x(k)$ based on Z^{k+l}, then $x(k+1)$ based on Z^{k+1+l}, etc.
3. Fixed interval smoothing: Estimate $x(k), k = 1, 2, \ldots, N$ based on Z^N.

Our focus will be on fixed interval smoothing. We introduce the following new notation:

$$\begin{aligned} x^*(k) &= \hat{\boldsymbol{x}}(k|N) \\ P^*(k) &= P(k|N) \\ v^*(k) &= \hat{\boldsymbol{v}}(k|N) \end{aligned}$$

Bar Shalom develops a smoother based on the traditional KF equations in Section 8.6. We will instead develop the square-root information smoother (SRIS), which is intuitive and more numerically stable.

Key observation: Smoother equations fall out of the MAP estimation approach.

12.2 Preliminaries

Suppose we have performed an SRIF forward (filtering) pass on measurements Z^N. From this will be left three groups of SRI equations:

1. Equations relating the state and process noise:

$$\begin{aligned} \bar{\boldsymbol{z}}_v(0) &= \bar{\boldsymbol{R}}_{vv}(0)v(0) + \bar{\boldsymbol{R}}_{vx}(1)x(1) + \bar{\boldsymbol{w}}_v(0) \\ &\vdots \\ \bar{\boldsymbol{z}}_v(N-1) &= \bar{\boldsymbol{R}}_{vv}(N-1)v(N-1) + \bar{\boldsymbol{R}}_{vx}(N)x(N) + \bar{\boldsymbol{w}}_v(N-1) \end{aligned}$$

   where $\bar{\boldsymbol{w}}_v(k) \sim (0, I), \quad k = 0, \ldots, N-1.$

2. Equations for the residuals:

$$z_r(1) = \bar{\boldsymbol{w}}_r(1)$$
$$\vdots$$
$$z_r(N) = \bar{\boldsymbol{w}}_r(N)$$

where $\bar{\boldsymbol{w}}_r(k) \sim (0, I), \quad k = 1, \ldots, N$. These residuals equations can be discarded; they are not needed for the SRIS.

3. An equation for the terminal state:

$$z_x(N) = R_{xx}(N)x(N) + w_x(N)$$

The SRIF enforces the dynamics model

$$x(k+1) = F(k)x(k) + G(k)u(k) + \Gamma(k)v(k)$$

going *forward* through the data.

Q: Do you suppose the filtered estimates $\hat{\boldsymbol{x}}(k)$ actually conform to this dynamics model?
A: Interestingly, they do not. A time history of $\hat{\boldsymbol{x}}(k)$ doesn't necessarily look like it could have come from the dynamics model because the estimates get jostled around by the innovations $\boldsymbol{\nu}(k)$ (see [4]). ■

A smoother's job is to enforce the dynamics model going *backward* in time.

Q: Does a smoother need to take in the measurements Z^N?
A: No, the filter already extracted all possible information from the measurements. ■

12.3 SRIS Algorithm

1. Let

$$z_x^*(N) = z_x(N)$$
$$R_{xx}^*(N) = R_{xx}(N)$$
$$w_x^*(N) = w_x(N)$$

2. If a smoothed state estimate and covariance is needed (e.g., for control, etc.), then compute

$$x^*(N) = R_{xx}^{*-1}(N)z^*(N)$$
$$P^*(N) = R_{xx}^{*-1}(N)R_{xx}^{*-T}(N)$$

3. Set $k = N - 1$. The cost function associated with the SRI equations at k can be written as

$$J_b^*[v(k), x(k+1), k] = \tfrac{1}{2}||\bar{\boldsymbol{R}}_{vv}(k)v(k) + \bar{\boldsymbol{R}}_{vx}(k+1)x(k+1) - \bar{\boldsymbol{z}}_v(k)||^2 + \tfrac{1}{2}||R_{xx}^*(k+1)x(k+1) - z_x^*(k+1)||^2$$

We seek to minimize this cost subject to the dynamics constraint.

4. Use the dynamics equation to eliminate $x(k+1)$ in favor of $x(k)$. Substituting for $x(k+1)$ and stacking yields a cost of the form

$$J_a^*[v(k), x(k), k] = \tfrac{1}{2}\left\| \begin{bmatrix} & \end{bmatrix} \begin{bmatrix} v(k) \\ x(k) \end{bmatrix} - \begin{bmatrix} & \end{bmatrix} \right\|^2$$

whose implied SRI equations are

$$\begin{bmatrix} \bar{\boldsymbol{z}}_v(k) - \bar{\boldsymbol{R}}_{vx}(k+1)G(k)u(k) \\ z_x^*(k+1) - R_{xx}^*(k+1)G(k)u(k) \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{R}}_{vv}(k) + \bar{\boldsymbol{R}}_{vx}(k+1)\Gamma(k) & \bar{\boldsymbol{R}}_{vx}(k+1)F(k) \\ R_{xx}^*(k+1)\Gamma(k) & R_{xx}^*(k+1)F(k) \end{bmatrix} \begin{bmatrix} v(k) \\ x(k) \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{w}}_v(k) \\ \boldsymbol{w}_x^*(k+1) \end{bmatrix}, \qquad \begin{bmatrix} \bar{\boldsymbol{w}}_v(k) \\ \boldsymbol{w}_x^*(k+1) \end{bmatrix} \sim (0, I)$$

Multiply both sides by $T_a(k) = Q_a^T(k)$ from QR-factorization of the block matrix. This decouples the SRI equations

$$\begin{bmatrix} z_v^*(k) \\ z_x^*(k) \end{bmatrix} = \begin{bmatrix} R_{vv}^*(k) & R_{vx}^*(k) \\ 0 & R_{xx}^*(k) \end{bmatrix} \begin{bmatrix} v(k) \\ x(k) \end{bmatrix} + \begin{bmatrix} w_v^*(k) \\ w_x^*(k) \end{bmatrix}, \qquad \begin{bmatrix} w_v^*(k) \\ w_x^*(k) \end{bmatrix} \sim (0, I)$$

without changing the associated cost:

$$\begin{aligned} J_a^*[v(k), x(k), k] = \tfrac{1}{2} \left\| R_{vv}^*(k)v(k) + R_{vx}^*(k)x(k) - z_v^*(k) \right\|^2 \\ + \tfrac{1}{2} \left\| R_{xx}^*(k)x(k) - z_x^*(k) \right\|^2 \end{aligned}$$

5. Choose $x^*(k)$ and $v^*(k)$ to minimize the cost:

$$\begin{aligned} x^*(k) &= R_{xx}^{*-1}(k) z_x^*(k) = \mathbb{E}\left[x(k) | Z^N\right] \\ v^*(k) &= R_{vv}^{*-1}(k)[z_v^*(k) - R_{vx}^*(k)x^*(k)] = \mathbb{E}\left[v(k) | Z^N\right] \end{aligned}$$

The associated error covariance matrices are

$$\begin{aligned} P^*(k) &= R_{xx}^{*-1}(k) R_{xx}^{*-T}(k) \\ P_{vv}^*(k) &= R_{vv}^{*-1}(k)[I + R_{vx}^*(k) R_{xx}^{*-1}(k) R_{xx}^{*-T}(k) R_{vx}^{*T}(k)] R_{vv}^{*-T}(k) \\ P_{vx}^*(k) &= -R_{vv}^{*-1}(k) R_{vx}^*(k) R_{xx}^{*-1}(k) R_{xx}^{*-T}(k) \end{aligned}$$

Note that $J_a[v^*(k), x^*(k), k] = 0$ because in forming the smoother's cost function we discarded the irreducible component from the SRIF.

6. If $k = 0$, stop; otherwise, decrement k by 1 and go to step 4, now using the SRI equations

$$\begin{aligned} \bar{\boldsymbol{z}}_v(k) &= \bar{\boldsymbol{R}}_{vv}(k)v(k) + \bar{\boldsymbol{R}}_{vx}(k+1)x(k+1) + \bar{\boldsymbol{w}}_v(k) \\ z_x^*(k+1) &= R_{xx}^*(k+1)x(k+1) + w_x^*(k+1) \end{aligned}$$

12.4 Nonlinear Difference Equations from ZOH Nonlinear Differential Equations

This lecture is a prelude to the EKF. It shows the basic flow from a continuous-time nonlinear dynamics and measurement model: we first discretize, then linearize. It further shows how continuous-time random processes can be properly converted to discrete time.

We wish to find an approximate Kalman Filter algorithm for non-linear systems of the form

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{f}\left[k, \boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{v}(k)\right] \\ \boldsymbol{z}(k) &= \boldsymbol{h}\left[k, \boldsymbol{x}(k)\right] + \boldsymbol{w}(k) \end{aligned}$$

which we could call the nonlinear difference equation form. We will construct this from the continuous-time non-linear model

$$\begin{aligned}\dot{\tilde{\boldsymbol{x}}}(t) &= \tilde{\boldsymbol{f}}\left[t, \tilde{\boldsymbol{x}}(t), \tilde{\boldsymbol{u}}(t)\right] + D(t)\tilde{\boldsymbol{v}}(t)\\ \tilde{\boldsymbol{z}}(t) &= \tilde{\boldsymbol{h}}\left[t, \tilde{\boldsymbol{x}}(t)\right] + \tilde{\boldsymbol{w}}(t)\end{aligned}$$

Recall that we already did this in Chapter 8 for linear systems when we transformed

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t) + D(t)\tilde{\boldsymbol{v}}(t)\\ \boldsymbol{z}(t) &= C(t)\boldsymbol{x}(t) + \boldsymbol{w}(t)\end{aligned}$$

into

$$\begin{aligned}\boldsymbol{x}(k+1) &= F(k)\boldsymbol{x}(k) + G(k)\boldsymbol{u}(k) + \Gamma(k)\boldsymbol{v}(k)\\ \boldsymbol{z}(k) &= H(k)\boldsymbol{x}(k) + \boldsymbol{w}(k)\end{aligned}$$

We use the tilde on, for example, $\tilde{\boldsymbol{x}}(t)$, to denote continuous time parameters. Our earlier abuse of notation (e.g., mixing $\boldsymbol{x}(t)$ and $\boldsymbol{x}(k)$) might cause confusion.

Let Δt be our (uniform) sampling interval and let $t_k \triangleq k\Delta t$. Then $\boldsymbol{z}(k)$ represents the sample $\boldsymbol{z}(t_k)$ after anti-aliasing filtering to avoid infinite variance in $\boldsymbol{w}(k)$.

12.4.1 Zero-Order-Hold (ZOH) Assumption

Under the ZOH assumption, we assume the sampling interval Δt is small enough that over the interval $t_k \le t < t_{k+1}$ we can approximate $\boldsymbol{u}(t)$ as constant:

$$\boldsymbol{u}(t) = \begin{cases} \boldsymbol{u}(k-1) & t_{k-1} \le t < t_k \\ \boldsymbol{u}(k) & t_k \le t < t_{k+1} \\ \text{etc.} & \end{cases}$$

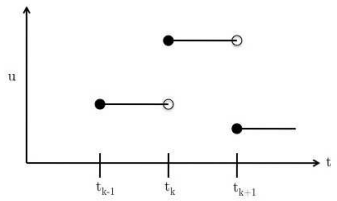


Figure 12.1: The control input under ZOH assumptions

This is illustrated in Fig. 12.1, which shows that the value of the control input is assumed constant (i.e. "held over") over the sampling interval Δt.

We will similarly assume that $\tilde{\boldsymbol{v}}(t) = \boldsymbol{v}(k)$ over $t_k \le t < t_{k+1}$ and we will later work out an expression for $Q(k) \triangleq \mathbb{E}\left[\boldsymbol{v}(k)\boldsymbol{v}^T(k)\right]$.

Let $\boldsymbol{x}_k(t)$ be the continuous-time state in the interval $t_k \le t < t_{k+1}$. Define the initial-value problem

$$\begin{aligned}\dot{\boldsymbol{x}}_k(t) &= \tilde{\boldsymbol{f}}\left[t, \boldsymbol{x}_k(t), \tilde{\boldsymbol{u}}(t)\right] + D(t)\tilde{\boldsymbol{v}}(t)\\ \tilde{\boldsymbol{z}}(t) &= \tilde{\boldsymbol{h}}\left[t, \tilde{\boldsymbol{x}}(t)\right] + \tilde{\boldsymbol{w}}(t)\\ \boldsymbol{x}_k(t_k) &= \tilde{\boldsymbol{x}}(t_k) = \boldsymbol{x}(k)\end{aligned}$$

where

$$\left.\begin{array}{l} \boldsymbol{u}(t) = \boldsymbol{u}(k) \\ \tilde{\boldsymbol{v}}(t) = \boldsymbol{v}(k) \end{array}\right\} \quad \text{on } t_k \le t < t_{k+1}$$

We can solve for $\boldsymbol{x}_k(t)$ on $t_k \le t < t_{k+1}$. The solution depends on $\boldsymbol{x}(k)$, $\boldsymbol{u}(k)$, and $\boldsymbol{v}(k)$:

$$\boldsymbol{x}_k(t_{k+1}) = f\left[k, \boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{v}(k)\right]$$

Here, $f[\cdot]$ is some procedure for integrating forward to t_{k+1}. This integration procedure could be Matlab's `ode45` solver, which is based on a Runge-Kutta (4,5) formula, or any appropriate numerical integration scheme.

12.4.2 Variance of the ZOH Process Noise

Q: Given

$$E\left[\tilde{\boldsymbol{v}}(t)\tilde{\boldsymbol{v}}^T(\tau)\right] = \delta(t-\tau)\tilde{Q}(t)$$
$$E\left[\boldsymbol{v}(k)\boldsymbol{v}^T(j)\right] = \delta_{kj}Q(k)$$

How do we relate $Q(k)$ and $\tilde{Q}(t)$?
A: If Δt is small, then

$$f\left[k, \boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{v}(k)\right] \approx \boldsymbol{x}(k) + \Delta t\left\{\tilde{\boldsymbol{f}}\left[t_k, \boldsymbol{x}(k), \boldsymbol{u}(k)\right] + D(t_k)\boldsymbol{v}(k)\right\}$$

This is simple Euler integration. The term of $f[\cdot]$ corresponding to the process noise is

$$\Delta t D(t_k)\boldsymbol{v}(k) \tag{12.1}$$

An alternative expression for the random component of $f[\cdot]$ that is valid for small Δt is

$$\int_{t_k}^{t_{k+1}} D(\tau)\tilde{\boldsymbol{v}}(\tau)\, d\tau \tag{12.2}$$

These two forms must be equivalent as $\Delta t \to 0$. We note that they are both zero mean. We equate their covariances to relate $Q(k)$ to $\tilde{Q}(t_k)$.

Covariance of (12.1):

$$\Delta t^2 D(t_k)Q(k)D^T(t_k)$$

Covariance of (12.2):

$$\int_{t_k}^{t_{k+1}} D(\tau)\tilde{Q}(\tau)D^T(\tau)\, d\tau \approx \Delta t D(t_k)\tilde{Q}(t_k)D^T(t_k)$$

Equating these two yields

$$Q(k) = \frac{\tilde{Q}(t_k)}{\Delta t} \quad \text{(for small } \Delta t\text{)}$$

Note that $\lim_{\Delta t \to 0} Q(k) = \infty$. This result comes from the whiteness of $\tilde{\boldsymbol{v}}(t)$. ■

Q: What if the measurement interval Δt is too large for the ZOH assumption to hold?
A: One can take m intermediate steps of length $\frac{\Delta t}{m}$ between each measurement. Choose m such that $\frac{\Delta t}{m}$ is small enough that

$$Q(k) = \frac{\tilde{Q}(k\frac{\Delta t}{m})}{\frac{\Delta t}{m}}$$

is a reasonable approximation. (In a Kalman filter, this entails doing m covariance propagations between measurement updates.) ■

12.4.3 Partial Derivatives of Difference Equations

To design a nonlinear estimation algorithm, we'll need a linearized model of the system dynamics in order to propagate our state error covariance matrix. Let $\bar{\boldsymbol{x}}(k)$ and $\bar{\boldsymbol{v}}(k) = 0$ be guesses of the state $\boldsymbol{x}(k)$ and process noise $\boldsymbol{v}(k)$ about which we wish to linearize. Then by Taylor expansion of the dynamics we have

$$\begin{aligned}\boldsymbol{x}(k+1) &= \boldsymbol{f}\left[k, \boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{v}(k)\right] \\ &\approx \boldsymbol{f}\left[k, \bar{\boldsymbol{x}}(k), \boldsymbol{u}(k), 0\right] + F(k)\left[\boldsymbol{x}(k) - \bar{\boldsymbol{x}}(k)\right] + \Gamma(k)\boldsymbol{v}(k)\end{aligned}$$

where

$$F(k) = \left[\left.\frac{\partial f[\centerdot]}{\partial \boldsymbol{x}(k)}\right|_{k,\bar{\boldsymbol{x}}(k),\boldsymbol{u}(k),0}\right], \quad \Gamma(k) = \left[\left.\frac{\partial f[\centerdot]}{\partial \boldsymbol{v}(k)}\right|_{k,\bar{\boldsymbol{x}}(k),\boldsymbol{u}(k),0}\right]$$

Our task is to develop a procedure for calculating these matrices. To begin, recall that

$$\begin{aligned}\dot{\boldsymbol{x}}_k(t) &= \tilde{\boldsymbol{f}}\left[t, \boldsymbol{x}_k(t), \boldsymbol{u}(t)\right] + D(t)\tilde{\boldsymbol{v}}(t) \\ \boldsymbol{x}_k(t_k) &= \boldsymbol{x}(k)\end{aligned}$$

Taking the partial of $vbdx_k(t)$ with respect to $\boldsymbol{x}(k)$ yields

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{x}(k)}\left[\dot{\boldsymbol{x}}_k(t)\right] &= \left[\left.\frac{\partial \tilde{\boldsymbol{f}}(\centerdot)}{\partial \boldsymbol{x}_k(t)}\right|_{t,\boldsymbol{x}_k(t),\boldsymbol{u}(k)}\right]\left[\left.\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{x}(k)}\right|_t\right] \\ &= A(t)\left[\left.\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{x}(k)}\right|_t\right]\end{aligned}$$

We now recognize that $\boldsymbol{x}_k(t_k) = \boldsymbol{x}(k)$ and exchange the order of differentiation to arrive at the following initial-value problem for the state transition matrix:

$$\begin{aligned}\frac{d}{dt}\left[\frac{\partial \boldsymbol{x}_k(t)}{\partial \boldsymbol{x}(k)}\right] &= A(t)\frac{\partial \boldsymbol{x}_k(t)}{\partial \boldsymbol{x}(k)} \\ \frac{\partial \boldsymbol{x}_k(t_k)}{\partial \boldsymbol{x}(k)} &= I_{n_x \mathrm{x} n_x}\end{aligned} \tag{12.3}$$

We recognize

$$\frac{\partial \boldsymbol{x}_k(t_k)}{\partial \boldsymbol{x}(k)}$$

as equivalent to the state-transition matrix $F(t, t_k)$ from the discussions of the continuous-time linear systems in Section 8.

Similarly, for $\Gamma(k)$, we have

$$\begin{aligned}\frac{d}{dt}\left[\frac{\partial \boldsymbol{x}_k(t)}{\partial \boldsymbol{v}(k)}\right] &= A(t)\frac{\partial \boldsymbol{x}_k(t)}{\partial \boldsymbol{v}(k)} + D(t) \\ \frac{\partial \boldsymbol{x}_k(t_k)}{\partial \boldsymbol{v}(k)} &= 0\end{aligned} \tag{12.4}$$

But we want to know the derivatives of the discrete $\boldsymbol{f}[\centerdot]$, not the continuous $\tilde{\boldsymbol{f}}[\centerdot]$. These derivatives are given by

$$\begin{aligned}\frac{\partial f[\centerdot]}{\partial \boldsymbol{x}(k)} &= \frac{\partial \boldsymbol{x}_k(t_{k+1})}{\partial \boldsymbol{x}(k)} \\ \frac{\partial f[\centerdot]}{\partial \boldsymbol{v}(k)} &= \frac{\partial \boldsymbol{x}_k(t_{k+1})}{\partial \boldsymbol{v}(k)}\end{aligned}$$

To obtain these, we integrate (12.3) and (12.4) from t_k to t_{k+1}, with their corresponding initial conditions. This can be accomplished using numerical integration (e.g. Matlab's `ode45`). In fact, we can integrate the two matrix differential equations *at the same time we're integrating the* $\boldsymbol{x}_k(t)$ *differential equation.* Let

$$\begin{aligned}\frac{\partial \boldsymbol{x}_k(t)}{\partial \boldsymbol{x}(k)} &= [\boldsymbol{\phi}_1(t), \boldsymbol{\phi}_2(t), \ldots, \boldsymbol{\phi}_{n_x}(t)] \\ \frac{\partial \boldsymbol{x}_k(t)}{\partial \boldsymbol{v}(k)} &= [\boldsymbol{\gamma}_1(t), \boldsymbol{\gamma}_2(t), \ldots, \boldsymbol{\gamma}_{n_v}(t)] \\ D(t) &= [\boldsymbol{d}_1(t), \boldsymbol{d}_2(t), \ldots, \boldsymbol{d}_{n_v}(t)]\end{aligned}$$

Then the matrix initial-value problems can be broken down column-by-column as

$$\begin{aligned}\dot{\boldsymbol{\phi}}_i(t) &= A(t)\boldsymbol{\phi}_i(t), \quad i \in \{1, 2, \ldots, n_x\} \\ \boldsymbol{\phi}_i(t_k) &= [0, 0, \ldots, 0, \underbrace{1}_{\text{ith row}}, 0, \ldots, 0]^T \\ \dot{\boldsymbol{\gamma}}_i(t) &= A(t)\boldsymbol{\gamma}_i(t) + d_i(t), \quad i \in \{1, 2, \ldots, n_v\} \\ \boldsymbol{\gamma}_i(t_k) &= \mathbf{0}\end{aligned}$$

Then define

$$\boldsymbol{x}_{\text{big}} = \left[\boldsymbol{x}_k^T, \boldsymbol{\phi}_1^T, \boldsymbol{\phi}_2^T, \ldots, \boldsymbol{\phi}_{n_x}^T, \boldsymbol{\gamma}_1^T, \boldsymbol{\gamma}_2^T, \ldots, \boldsymbol{\gamma}_{n_v}^T\right]^T$$

This state vector has dimension $n_x(n_x + n_v + 1) \times 1$. Finally, a large numerical integration routine can be written, with the appropriate previously-defined initial conditions, to solve

$$\dot{\boldsymbol{x}}_{\text{big}} = \tilde{\boldsymbol{f}}_{\text{big}}\left[t, \boldsymbol{x}_{\text{big}}, \boldsymbol{u}(k), \boldsymbol{v}(k)\right]$$

From the solution $vbx(t_{k+1})$ we can extract $\bar{\boldsymbol{x}}(k+1) = \boldsymbol{f}\left[k, \bar{\boldsymbol{x}}(k), \boldsymbol{u}(k), 0\right]$, $F(k)$, and $\Gamma(k)$.

12.5 Nonlinear Estimation for Dynamical Systems

Scribe: Shaina Johl

Problem statement

Dynamics model:

$$x(k+1) = f\left[k, x(k), u(k), v(k)\right] \tag{12.5}$$

$$E[v(k)] = 0, \quad E[v(k)v^T(j)] = \delta_{kj}Q(k) \tag{12.6}$$

Measurement model:

$$z(k) = h\left[k, x(k)\right] + w(k) \tag{12.7}$$

$$E\left[w(k)\right] = 0, \;\; E\left[w(k)w(j)\right] = \delta_{jk}R(k) \tag{12.8}$$

Q: How to optimally estimate $x(k)$?
A: See Bar Shalom 10.2 ■

Our strategy is a sub-optimal strategy. Approximate the optimal estimates by applying linear MMSE estimator to the non-linear problem.

- Approximate

$$\overline{x}(k+1) = E\left[x(k+1)|Z^k\right] \tag{12.9}$$
$$\overline{z}(k+1) = E\left[z(k+1)|Z^k\right] \tag{12.10}$$

- Approximate their covariances

$$\overline{P}(k+1),\ \ \overline{P}_{kz}(k+1),\ \ \overline{P}_{zz}(k+1) \tag{12.11}$$

- If we can assume that the approximations are valid then we can use our old update equations for the measurement update of the Kalman filter

$$\hat{x}(k+1) = \overline{x}(k+1) + \overline{P}_{xz}(k+1)\bar{P}_{zz}(k+1)\left[z(k+1) - \overline{z}(k+1)\right] \tag{12.12}$$
$$P(k+1) = \overline{P}(k+1) - \overline{P}_{xz}(k+1)\bar{P}_{zz}^{-1}(k+1)\overline{P}_{xz}^{T}(k+1) \tag{12.13}$$

12.5.1 Standard EKF Methods for Approximation

EKF Philosophy: Linearize the non-linear equations about the current best state estimate.

Prediction:

$$\overline{x}(k+1) = E\left[f\left[k, x(k), u(k), v(k)\right]|Z^k\right] \tag{12.14}$$

Expand in a Taylor series expansion about $x(k) = \hat{x}(k), v(k) = \hat{v}(k) = 0$:

$$\bar{x}(k+1) = E\left[f\left[k, \hat{x}(k), u(k), 0\right] + \underbrace{\left[\frac{\partial f}{\partial x}\Big|_{k,\hat{x}(k),u(k),0}\right]}_{F(k)}\left[x(k) - \hat{x}(k)\right]\cdots\right.$$
$$\left. + \left[\underbrace{\left[\frac{\partial f}{\partial v(k)}\Big|_{k,\hat{x}(k),u(k),0}\right]}_{\Gamma(k)} v(k)\right] + \text{Higher Order Terms}|Z^k\right] \tag{12.15}$$

Neglect higher order terms, hoping that the linearization is valid over the likely values of $x(k)$! This is surprisingly effective.

Now take the expectations:

$$\overline{x}(k+1) = f[k, \hat{x}(k), u(k), 0] + F(k)\underbrace{E[x(k) - \hat{x}(k)|Z^k]}_{\text{approximately}=0} + \Gamma(k)\underbrace{E\left[v(k)|Z^k\right]}_{0} \tag{12.16}$$

Note: Compute $\overline{x}(k+1)$ via our numerical procedure $f[k, x(k), u(k), v(k)]$.

Watch out: if we had retained the second order derivatives $\left[\frac{\partial^2 f}{\partial x^2}\right]$ and $\left[\frac{\partial^2 f}{\partial v^2}\right]$ then $P(k)$ and $Q(k)$ would have affected $\overline{x}(k+1)$. This is a property of non-linear filters in general.

$$\overline{P}(k+1) = E\left[(x(k+1) - \overline{x}(k+1))(\cdots)^T\right] \tag{12.17}$$

But,

$$x(k+1) = \overline{x}(k+1) + F(k)\left[x(k) - \hat{x}(k)\right] + \Gamma(k)v(k) + \underbrace{\text{HOT}}_{\text{neglect}} \tag{12.18}$$

Substituting and taking expectations yields:

$$\overline{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k) \tag{12.19}$$

Note: This is the same as for the linear Kalman Filter. The only difference is that $F(k)$ and $\Gamma(k)$ are computed by numerical integration.

Measurement update:

$$\begin{aligned}
\overline{z}(k+1) &= E\left[h\left[k+1, x(k+1)\right] + w(k+1)|Z^k\right] \\
&= E\left[h\left[k+1, \overline{x}(k+1)\right] + \underbrace{\left[\frac{\partial h}{\partial x}|_{k+1,\overline{x}(k+1)}\right]}_{H(k+1)}\left[x(k+1) - \overline{x}(k+1)\right] + \text{HOT} + w(k+1)|Z^k\right] \\
&= h\left[k+1, \overline{x}(k+1)\right] + H(k+1)\underbrace{E\left[x(k+1) - \overline{x}(k+1)|Z^k\right]}_{\text{approximately}=0} + \underbrace{E\left[w(k+1)|Z^k\right]}_{0} \\
&= \boxed{h\left[k+1, \overline{x}(k+1)\right]}
\end{aligned} \tag{12.20}$$

Note: If HOTs are retained, then $\overline{P}$ affects $\overline{z}$.

$$\overline{P}_{xz}(k+1) = E\left[(x(k+1) - \overline{x}(k+1))(z(k+1) - \overline{z}(k+1))^T|Z^k\right] \tag{12.21}$$

Note that

$$z(k+1) \approx \bar{z}(k+1) + H(k+1)\left[x(k+1) - \overline{x}(k+1)\right] + w(k+1) \tag{12.22}$$

Therefore

$$\overline{P}_{xz}(k+1) = E\left[(x(k+1) - \overline{x}(k+1))(H(k+1)\left[x(k+1) - \overline{x}(k+1) + w(k+1)\right])^T|Z^k\right] \tag{12.23}$$

$$= \overline{P}(k+1)H^T(k+1) \tag{12.24}$$

Similarly,

$$\overline{P}_{zz}(k+1) = H(k+1)\overline{P}(k+1)H^T(k+1) + R(k+1) \tag{12.25}$$

12.5.2 EKF as an algorithm

1. Start with $\hat{x}(0), P(0)$
2. Set $k = 0$

3. Compute

$$\overline{x}(k+1) = f\left[k, \hat{x}(k), u(k), 0\right] \tag{12.26}$$

$$F(k) = \left.\frac{\partial f}{\partial x}\right|_{k,\hat{x}(k),u(k),0} \tag{12.27}$$

$$\Gamma(k) = \left.\frac{\partial f}{\partial v(k)}\right|_{k,\hat{x}(k),u(k),0} \tag{12.28}$$

$$\overline{P}(k+1) = F(k)P(k)F^T(k) + \Gamma(k)Q(k)\Gamma^T(k) \tag{12.29}$$

$$\tag{12.30}$$

4. Compute the Measurement Update

$$\bar{z}(k+1) = h[k+1, \bar{x}(k+1)] \tag{12.31}$$

$$H(k+1) = \left.\frac{\partial h}{\partial x}\right|_{k+1,\overline{x}(k+1)} \tag{12.32}$$

$$\nu(k+1) = z(k+1) - \overline{z}(k+1) \tag{12.33}$$

$$S(k+1) = H(k+1)\bar{P}(k+1)H^T(k+1) + R(k+1) = \overline{P}_{zz}(k+1) \tag{12.34}$$

$$W(k+1) = \overline{P}(k+1)H^T(k+1)S^{-1}(k+1) \tag{12.35}$$

$$\hat{x}(k+1) = \overline{x}(k+1) + W(k+1)\nu(k+1) \tag{12.36}$$

$$P(k+1) = \overline{P}(k+1) - W(k+1)S(k+1)W^T(k+1) \tag{12.37}$$

Note that we can use alternate formulas instead of W and P for the Kalman filter.

5. Filter:

$$P(k+1) = [\bar{P}^{-1}(k+1) + H^T(k+1)R^{-1}(k+1)H(k+1)]^{-1} \tag{12.38}$$

Joseph form (guarantees that $P(k+1) > 0$:

$$P(k+1) = [I - W(k+1)H(k+1)]\bar{P}(k+1)[I - W(k+1)H(k+1)]^T \tag{12.39}$$

$$+ W(k+1)R(k+1)W^T(k+1) \tag{12.40}$$

$$W(k+1) = P(k+1)H^T(k+1)R^{-1}(k+1) \tag{12.41}$$

6. Increment k by 1 and go to Step 3.

12.6 Re-interpretation of EKF as a Gauss Newton Step

It can be thought of as a step towards the solution of the nonlinear MAP problem.

Consider the MAP cost function:

$$\begin{aligned} J_a\left[k, v\left(k\right), x\left(k\right), x(k+1)\right] &= \frac{1}{2}[x\left(k\right) - \hat{x}\left(k\right)]^T P^{-1}\left(k\right)[...] \\ &+ \frac{1}{2}v^T\left(k\right)Q^{-1}\left(k\right)v\left(k\right) \\ &+ \frac{1}{2}\{z\left(k+1\right) - h\left[k+1, x\left(k+1\right)\right]\}^T R^{-1}\left(k+1\right)\{...\} \end{aligned} \tag{12.42}$$

which comes from minimizing

$p\left[x\left(k\right),v\left(k\right),x\left(k+1\right)Z^{k}\right]=Cexp(-J_{a})$

Minimizing of J_a must be subject to the nonlinear dynamics. Define a nonlinear inverted dynamics function:

$x\left(k\right)=f^{-1}[k,x\left(k+1\right),u\left(k\right),v(k)]$

Here, the function is defined such that:

$x\left(k+1\right)=f[k,f^{-1}\left\{k,x\left(k+1\right),u\left(k\right),v\left(k\right)\right\},u\left(k\right),v(k)]$

The above can be better visualized by taking the example of our original nonlinear problem

$$f=F(k)x(k)+G(k)u(k)+\Gamma(k)v(k) \tag{12.43}$$

$$f^{-1}=F^{-1}(k)\left[x(k+1)-G(k)u(k)-\Gamma(k)\right] \tag{12.44}$$

We can obtain f^{-1} by numerically integrating backward f in time.

Use f^{-1} to eliminate $x(k)$ from the MAP cost function and thereby define:

$$\begin{aligned} J_{b}\left[k,v\left(k\right),x(k+1)\right] &= \frac{1}{2}\left[f^{-1}[k,x(k+1,u(k),v(k))]-\hat{x}\left(k\right)\right]^{T}P^{-1}\left(k\right)[...] \\ &+ \frac{1}{2}v^{T}\left(k\right)Q^{-1}\left(k\right)v\left(k\right)+\frac{1}{2}\left\{z\left(k+1\right)-h\left[k+1,x\left(k+1\right)\right]\right\}^{T}R^{-1}\left(k+1\right)\left\{\right. \end{aligned} \tag{12.45}$$

This is just a weighted least squares cost function for errors in the three equations given below:

1. $0=f^{-1}[k,x(k+1),u(k),v(k)]-\hat{x(k)}$ with weighting P^{-1}
2. $0=v(k)$ with weighting Q^{-1}
3. $0=z(k+1)-h[k+1,x(k+1)]$ with weighting $R^{-1}(k+1)$

Strategy: We use the Gauss-Newton method to linearize and solve. We start by first linearizing about the first guess at $x(k+1)=\bar{x}(k+1)$ and $v(k)=0$. Here $\bar{x}(k+1)$ is obtained by the equation below using our initial $\hat{x}(k)$

$$\bar{x}(k+1)=f[k,\hat{x}(k),u(k),v(j)] \tag{12.46}$$

The next step is to solve the linearized least squares problem for $\hat{x}(k)$ and $\hat{v}(k)$, then re-linearize and then iterate. We do this by linearizing each of the three equations above.

For 1) we have:

$$\begin{aligned} 0 &\cong f^{-1}\left[k,\bar{x}\left(k+1\right),u\left(k\right),v\left(k\right)\right]+\left[\frac{\partial f^{-1}}{\partial x\left(k+1\right)}\right]_{k,\bar{x}(k+1)}\left[x\left(k+1\right)-\bar{x}\left(k+1\right)\right] \\ &+ \left[\frac{\partial f^{-1}}{\partial v\left(k\right)}\right]_{k,\bar{x}(k+1)}\left[v\left(k\right)-0\right]-\hat{x}\left(k\right) \end{aligned} \tag{12.47}$$

In the above expression $\hat{x}\left(k\right)=f^{-1}\left[k,\bar{x}\left(k+1\right),u\left(k\right),v\left(k\right)\right]$ from the definition of inverse. Thus the $\hat{x}\left(k\right)'s$ cancel.

It can be shown that:

$$\left[\frac{\partial f^{-1}}{\partial x\left(k+1\right)}\right]_{k,\bar{x}(k+1),u(k),0} = F^{-1}\left(k\right) = \left[\frac{\partial f}{\partial x(k)}\right]^{-1}_{k,\hat{x}(k),u(k),0}$$

And

$$\left[\frac{\partial f^{-1}}{\partial v\left(k\right)}\right]_{k,\bar{x}(k+1),u(k),0} = -F^{-1}\left(k\right)\Gamma\left(k\right) = -F^{-1}(k)\left[\frac{\partial f}{\partial v(k)}\right]^{-1}_{k,\hat{x}(k),u(k),0}$$

The linearized equation then becomes:

$$0 = F^{-1}\left(k\right)\left[x\left(k+1\right) - \bar{x}\left(k+1\right)\right] - \left[F^{-1}\left(k\right)\Gamma\left(k\right)v(k)\right]$$

Similarly, we linearize 3) as shown below:

$$0 \cong z\left(k+1\right) - h\left[k+1, \bar{x}\left(k+1\right)\right] - \left[\frac{\partial h}{\partial x\left(k+1\right)}\right]_{k,\bar{x}(k+1)}\left[x\left(k+1\right) - \bar{x}\left(k+1\right)\right]$$

We define, $H\left(k+1\right) = \left[\frac{\partial h}{\partial x(k+1)}\right]_{k,\bar{x}(k+1)}$

Also, we know, $\bar{z}\left(k+1\right) = h\left[k+1, \bar{x}\left(k+1\right)\right]$

The linearized equation for 3) reduces to:

$$0 = z\left(k+1\right) - \bar{z}\left(k+1\right) - H(k+1)[x\left(k+1\right) - \bar{x}\left(k+1\right)]$$

Summarize: To summarize the three linearized equations are:

1. $$0 = F^{-1}\left(k\right)\left[x\left(k+1\right) - \bar{x}\left(k+1\right)\right] - \left[F^{-1}\left(k\right)\Gamma\left(k\right)v(k)\right]$$
2. $$0 = v\left(k\right)$$
3. $$0 = z\left(k+1\right) - \bar{z}\left(k+1\right) - H\left(k+1\right)\left[x\left(k+1\right) - \bar{x}\left(k+1\right)\right]$$

New cost function obtained by substituting the linearized equations back into the cost function.J_b

$$\begin{aligned} J_b\left[k, v\left(k\right), x(k+1)\right] &= \frac{1}{2}[x\left(k+1\right) - \bar{x}\left(k+1\right) - \Gamma\left(k\right)v(k)]^T F^{-T}\left(k\right)P^{-1}\left(k\right)F^{-1}\left(k\right)[...] \\ &+ \frac{1}{2}v^T\left(k\right)Q^{-1}\left(k\right)v\left(k\right) \\ &+ \frac{1}{2}\{z\left(k+1\right) - \bar{z}\left(k+1\right) - H\left(k+1\right)\left[x\left(k+1\right) - \bar{x}\left(k+1\right)\right]\}^T R^{-1}\left(k+1\right) \end{aligned} \quad (12.48)$$

The new cost function is minimized w.r.t $x\left(k+1\right)$and $v\left(k\right)$. If the linearization is good, this is very close to maximizing the aposteriori likelihood function and can be viewed as the justification for EKF. Also, there are analogies to represent Extended Kalman filter as a square root information filter.

12.6.1 Iterated Extended Kalman Filter

The traditional EKF takes only one Gauss-Newton step; the iterated EKF takes multiple Gauss-Newton steps, re-linearizing only the measurement equation at each step. One could also imagine re-linearizing the f^{-1} dynamics using $\hat{x}(k+1)$ *and* $\hat{v}(k)$. In fact, one can imagine re-linearizing over the past N steps. This is the idea behind the moving horizon estimator, also known as the backward-smoothing EKF (BSEKF) [4, 5].

Consider

$$\begin{aligned} J_c[k+1, x(k+1)] &= J_b[k, v_{\text{OPT}}(k), x(k+1)] \\ &\approx \tfrac{1}{2}[x(k+1) - \bar{x}(k+1)]^T \bar{P}^{-1}(k)[\dots] \\ &\quad + \tfrac{1}{2}[z(k+1) - h[k+1, x(k+1)]]^T R^{-1}(k+1)[\dots] \end{aligned}$$

where v_{OPT} is the value of $v(k)$ that minimizes J_b. The approximation in the above comes from our having linearized the dynamics about $\bar{x}(k+1)$. The measurement equation remains nonlinear.

Strategy: Start with a guess of $x(k+1)$; solve the linearized LS problem for a new guess; iterate.

Let $\hat{x}^i(k+1)$ be the estimated $x(k+1)$ after the i^{th} Gauss-Newton step; note that $\hat{x}^0(k+1) = \bar{x}(k+1)$. Also let

$$H^i(k+1) = \left[\left.\frac{\partial h}{\partial x}\right|_{k+1, \hat{x}^i(k+1)}\right]$$

The linearized measurement equation after the i^{th} step becomes

$$0 = z(k+1) - h\left[k+1,\ \hat{x}^i(k+1)\right] - H^i(k+1)\left[x(k+1) - \hat{x}^i(k+1)\right]$$

These iterations can be related back to the standard KF equations as follows:

$$\begin{aligned} 0 &= \left[\frac{\partial J_c^i}{\partial x(k+1)}\right]^T \\ &= \bar{P}^{-1}(k+1)[x(k+1) - \bar{x}(k+1)] \\ &\quad - {H^i}^T(k+1) R^{-1}(k+1)\left\{z(k+1) - h\left[k+1,\ \hat{x}^i(k+1)\right] - H^i(k+1)\left[x(k+1) - \hat{x}^i(k+1)\right]\right\} \end{aligned}$$

Note that this equation is linear in $x(k+1)$.

Let

$$P^{i+1}(k+1) = \left[\bar{P}^{-1}(k+1) + {H^i}^T(k+1) R^{-1}(k+1) H^i(k+1)\right]^{-1}$$

Then solving for $x(k+1)$ yields

$$\begin{aligned} \hat{x}^{i+1}(k+1) = \hat{x}^i(k+1) &+ P^i(k+1) {H^i}^T(k+1) \bar{P}^{-1}(k+1)\left\{z(k+1) - h\left[k+1,\ \hat{x}^i(k+1)\right]\right\} \\ &+ P^i(k+1) \bar{P}^{-1}(k+1)[\bar{x}(k+1) - \hat{x}^i(k+1)] \end{aligned}$$

with

$$\hat{P}^0(k+1) = \bar{P}(k+1) \quad \text{and} \quad \hat{x}^0(k+1) = \bar{x}(k+1)$$

Note that $\hat{x}^1(k+1)$ is the traditional (non-iterated) EKF estimate. Stop iterating when

$$\left\|\hat{x}^{i+1}(k+1) - \hat{x}^i(k+1)\right\|$$

gets very small. Use step size adjustment as before there is reason to worry about divergence.

Chapter 13

Multiple Model (MM) Filtering

Let θ be a vector of unknown parameters—it might affect *any* of $F, G, \Gamma, H, Q, R, \hat{x}(0), P(0)$.

A Bayesian approach to multiple model filtering seeks the following:

$$p[x(k), \theta | Z^k] = p[x(k)|\theta, Z^k]\; p[\theta | Z^k]$$

For convenience, let θ take on values in $\{\theta_1, \theta_2, \ldots, \theta_M\}$.

Then:

- $p[x(k)|\theta_j, Z^k]$ is the posterior density of $x(k)$ under the jth model.
- $p[\theta = \theta_j | Z^k] \triangleq \mu_j(k)$ is the probability that the jth model is correct given Z^k ($\sum_{j=1}^{M} \mu_j = 1$).

13.1 Static Case

Consider the static case, where $\theta(k) = \theta = \text{const}$.

13.1.1 Strategy

1. Determine how to propagate $\mu_j(k)$ to $\mu_j(k+1)$
2. Find $p[x(k)|Z^k] = \sum_{j=1}^{M} \mu_j(k)\; p[x(k)|\theta_j, Z^k]$ and use this to choose an optimal $\hat{x}(k)$ per MAP or MMSE. Also, calculate $P(k)$.

13.1.2 Steps

1. Propagate:

$$\begin{aligned} \mu_j(k) &= p[\theta_j | Z^k] \\ &= p[\theta_j | z(k), Z^{k-1}] \\ &= \frac{p[z(k)|\theta_j, Z^{k-1}]\; p[\theta_j | Z^{k-1}]}{p[z(k)|Z^{k-1}]} \\ \implies \mu_j(k) &= \frac{p[z(k)|\theta_j, Z^{k-1}]\; \mu_j(k-1)}{\sum_{l=1}^{M} p[z(k)|\theta_l, Z^{k-1}]\; \mu_l(k-1)} \end{aligned}$$

The factor $p[z(k)|\theta_j, Z^{k-1}]$ is the likelihood function of model j at time k.
In the linear Gaussian case:

$$\begin{aligned} p[z(k)|\theta_j, Z^{k-1}] &= \mathcal{N}(z(k); H_j(k)\,\bar{x}_j(k), S_j(k)) \\ &= \mathcal{N}(\nu(k); 0, S_j(k)) = p(\nu_j(k)) \end{aligned}$$

where $S_j(k)$ is the innovation covariance matrix for model j and $\nu_j(k) = z(k) - H_j(k)\,\bar{x}_j(k)$

2. Estimate:

$$\begin{aligned} \hat{x}_{MAP}(k) &= \arg\max_{x(k)}\ p[x(k)|Z^k] \\ \hat{x}_{MMSE}(k) &= E[x(k)|Z^k] = \sum_{j=1}^{M} \hat{x}_j(k)\,\mu_j(k) \\ P_{MMSE}(k) &= \sum_{j=1}^{M} \mu_j(k) \left(P_j(k) + \overbrace{[\hat{x}_j(k) - \hat{x}_{MMSE}(k)]\,[\hat{x}_j(k) - \hat{x}_{MMSE}(k)]^T}^{\text{spread of the means}} \right) \end{aligned}$$

Fig. 13.1 shows the schematic for the MMSE case.

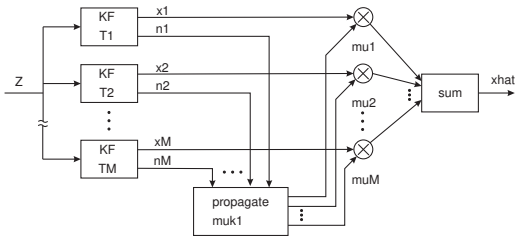


Figure 13.1: Multiple model filter schematic for MMSE case

13.2 Remarks

- Choosing $\{\theta_1, \theta_2, \ldots, \theta_M\}$ is the subject of ongoing research.
  - hard to distinguish similar θ_j
  - large Q makes models hard to distinguish
- Approximate multiple model filter for nonlinear systems can be designed—just replace KF with EKF.
- If the correct θ is among the θ_j, then corresponding $\mu_j(k)$ will approach 1 as $k \to \infty$.

Q: What can be done for time varying $\theta(k)$
A1: Ad hoc modification: impose an artificial lower bound on the $\mu_j(k)$
A2: Dynamic multiple model filter

Let $\theta(k) \in \{\theta_1, \theta_2, \ldots, \theta_M\}$ and assume model switching is a Markov process with transition probabilities given by

$$p_{ij} = p[\theta(k) = \theta_j | \theta(k-1) = \theta_i]$$

Then for a general non-linear system

$$\left.\begin{array}{rcl} x(k+1) & = & f[k, x(k), \theta(k)] \\ z(k) & = & h[k, x(k), \theta(k)] \end{array}\right\} \text{Hidden Markov Model}$$

There exist M^k possible sequences at time k.

Q: How can we deal practically with such exponential growth?
A:

Method	Complexity (# of filters)
(1) retain only most probable M sequences	M
(2) combine sequences that differ only before last N steps	M^N

Bar Shalom introduces:

- Generalized Pseudo-Bayesian estimator of first order (GPB1) ($N = 1$)
- Generalized Pseudo-Bayesian estimator of second order (GPB2) ($N = 2$)
- Interacting Multiple Model Estimator (IMM) (like $N = 2$ but only M filters used)

For additional material, please refer to Mayback section 10.8 and Bar Shalom Section 11.6.

Chapter 14

Particle Filtering

14.0.1 Motivation

A more perfect linear estimator. All MMSE estimators, including approximate techniques such as the Extended Kalman Filter (EKF) and Sigma Point Filter (SPF), reduce to taking the approximate conditional mean and covariance:

$$\hat{\boldsymbol{x}}(k) = \mathbb{E}\left[x(k)|z^k\right]$$
$$P(k) = \mathbb{E}\left[[x(k) - \hat{\boldsymbol{x}}(k)][x(k) - \hat{\boldsymbol{x}}(k)]^T\right]$$

Remarks

- EKF-produced approximations for $\hat{\boldsymbol{x}}(k)$ and $P(k)$ are poor
- SPF approximations are generally better
- But both EKF and SPF are considering only the first two moments of the posterior pdf $p[x(k)|z^k]$

Q: What if the posterior pdf is multi-modal?
Q: What would the estimation look like without approximation?
A: (Assume no control input for simplicity)

$$x(k+1) = f[k, x(k), v(k)]$$
$$z(k+1) = h[k+1, x(k+1)] + w(k+1)$$

■

Propagation

Prior at $k+1$ (Also known as the Chapman-Komorgorov Equation):

$$p[x(k+1)|z^k] = \int p[x(k+1)|x(k)]p[x(k)|z^k]dx(k)$$

Update

$$p[x(k+1)|z^{k+1}] = \frac{p[z(k+1)|x(k+1)]p[x(k+1)|z^k]}{p[z(k+1)|z^k]}$$

This recursion allows us access to the entire posterior pdf. We can then choose to optimize our estimate against any criterion we wish.

Q: Why should we settle for anything less than this optimal estimate?
A: Think about the one-dimensional problem: We can approach optimality by numerical integration. But grid size must be small and the grid must capture the tails of the distribution.
Now think about the multi-dimensional problem: If we need 100 cells per dimension and we have n_x dimensions, the total number of cells is 100^{n_x}. Another problem: The grid-based method is not readily parallelizable. ■

The Particle Filter suffers from the same drawbacks as the grid-based method (massive memory and computation is required even for modest n_x. Not readily parallelizable), but we'll study it anyway.
Key Idea: Estimate the posterior using weighted "particles":

$$p[x(k)|z^k] \approx \sum_{i=1}^{N_s} w_i \delta[x(k) - \chi_i(k)]$$

$$\sum_i w_i = 1$$

χ_i are called particles or support points
To choose the support points χ_i and the weights w_i we approximate the process of drawing a sample from a distribution.
Problem: For an arbitrary distribution, it is computationally expensive to generate random samples.

Importance Sampling

Suppose we can't draw from $p(x)$ economically, but we can from another distribution $q(x)$, called the importance density.

1. $q(x)$ is non-zero everywhere $p(x)$ is non-zero
2. $q(x)$ and $p(x)$ have similar mean and covariance

Then $p(x)$ can be approximated as: $p(x) \approx \sum_{i=1}^{N_s} w_i \delta[x - \chi_i]$ where we draw $\{\chi_i\}_{i=1}^{N_s}$ from $q(x)$ and w_i's are given by

$$w_i = c \cdot \frac{p(\chi_i)}{q(\chi_i)}$$

where c is a normalizing constant.
The approximation becomes exact as $N_s \to \infty$
Open Questions:

- **Q:** How fast does the approximation approach the truth?
- **Q:** What should N_s be for a given problem?

14.0.2 Particle Filter Algorithm

Basic Idea: Like the SPF except that

1. We use more samples
2. We draw them randomly
3. We don't attempt to shoehorn transformed samples into a Gaussian distribution

Generic Steps (Later we'll focus specifically on the Bootstrap part):

1. Choose an importance density $q[x(k)|z^k]$
2. Draw samples $\chi_i(k), i = 1, 2, ..., N_s$ from $q[x(k)|z^k]$
3. Compute importance weights: $w_i(k) = \frac{p[\chi_i(k)|z^k]}{q[\chi_i(k)|z^k]}[\sum_i \frac{p[\chi_i(k)|z^k]}{q[\chi_i(k)|z^k]}]^{-1}$
4. If necessary, approximate $p[x(k)|z^k] \approx \sum_{i=1}^{N_s} w_i(k)\delta[x(k) - \chi_i(k)]$
   And compute basic estimation quantities:

$$\hat{\boldsymbol{x}}(k) = \mathbb{E}\left[x(k)|z^k\right] \approx \sum_i w_i(k)\chi_i(k)$$

$$P(k) = \mathbb{E}\left[[x(k) - \hat{\boldsymbol{x}}(k)][x(k) - \hat{\boldsymbol{x}}(k)]^T|z^k\right]$$
$$\approx \sum_i w_i(k)[\chi_i(k) - \hat{\boldsymbol{x}}(k)][\chi_i(k) - \hat{\boldsymbol{x}}(k)]^T$$

   This step is not necessary unless you actually have to provide a single estimate.

How to choose $q[x(k)|z^k]$?

Choose q to factor in a convenient way:

$$q[\chi_i(k)|\chi_i(0), \cdots, \chi_i(k-1)] \cdot q[\chi_i(0), \cdots, \chi_i(k-1)|z^k]$$

Consider the true density

$$p[\chi i(0), \cdots, \chi_i(k)|z^k]$$
$$= p[\chi_i(0), \cdots, \chi_i(k)|z^{k-1}, z(k)]$$
$$= \frac{p[z(k)|\chi_i(0), \cdots, \chi_i(k), z^{k-1}]p[\chi_i(0), \cdots, \chi_i(k)|z^{k-1}]}{p[z(k)|z^{k-1}]}$$

Assume: $p[z(k)|\chi_i(0), \cdots, \chi_i(k), z^{k-1}] = p[z(k)|\chi_i(k)]$ (zeroeth-order Markov assumption)
Then $p[\chi_i(0), ..., \chi_i(k)|z^k] = \frac{p[z(k)|\chi_i(k)]p[\chi_i(0), \cdots, \chi_i(k)|z^{k-1}]}{p[z(k)|z^{k-1}]}$
Use Bayes' rule on the second factor in the numerator to get:

$$p[\chi_i(0), ..., \chi_i(k)|z^{k-1}] = p[\chi_i(k)|\chi_i(0), \cdots, \chi_i(k-1), z^{k-1}]p[\chi_i(0), ..., \chi_i(k-1)|z^{k-1}]$$

Assume: $p[\chi_i(k)|\chi_i(0), \cdots, \chi_i(k-1), z^{k-1}] = p[\chi_i(k)|\chi_i(k-1)]$ (first-order Markov assumption)

$$p[\chi_i(0), ..., \chi_i(k)|z^k] = \frac{p[z(k)|\chi_i(k)]p[\chi_i(k)|\chi_i(k-1)]p[\chi_i(0), \cdots, \chi_i(k-1)|z^{k-1}]}{p[z(k)|z^{k-1}]}$$

Now consider the weights:

$$w_i(k) = \frac{p[\chi_i(0), ..., \chi_i(k)|z^k]}{q[\chi_i(0), ..., \chi_i(k)|z^k]}$$

Then, making use of the above, we can write:

$$w_i(k) = c \cdot \frac{p[z(k)|\chi_i(k)]p[\chi_i(k)|\chi_i(k-1)]}{q[\chi_i(k)|\chi_i(0), \cdots, \chi_i(k-1)]} \cdot \frac{p[\chi_i(0), \cdots, \chi_i(k-1)|z^{k-1}]}{q[\chi_i(0), \cdots, \chi_i(k-1)|z^k]}$$

where the second fraction is the same as $w_i(k-1)$

How to pick $q[\chi_i(k)|\chi_i(0), \cdots, \chi_i(k-1)]$?

Bootstrap method: $q = p[\chi_i(k)|\chi_i(k-1)]$
Then the $w_i(k)$ become:

$$w_i = c \cdot p[z(k)|\chi_i(k)] \cdot w_i(k-1)$$

(This is similar to the Multiple-Model updates for the μ_i's)

14.0.3 Bootstrap Algorithm

1. Draw initial particles $\chi_i(0), i \in [1, N_s]$ from the known initial probability density $p[x(0)]$ and initialize weights one each particle equally:

$$\chi_i(0)\ p[x(0)], i \in [1, N_s]$$
$$w_i(0) = \frac{1}{N_s}, i \in [1, N_s]$$

2. Draw one sample of the process noise for each particle. Note this step is performed only because the bootstrap filter makes the particular choice of importance density $q[x(k)] = p[x(k)|x(k-1)]$, which is generally made because process noise is often assumed to be Gaussian.

$$v_i(k-1) \sim p[\bar{\boldsymbol{v}}(k-1)], i \in [1, N_s]$$

3. Propagate each particle forward according to the dynamics function. This is analogous to the prediction step in teh EKF or SPF, as it predicts the particle set forward in time without any measurement updates. Notice the particle weights don't change during this step.

$$\chi_i(k) = f[\chi_i(k-1), v_i(k-1), u(k-1)]$$

4. Repeat steps 2 and 3 until the next measurement update. Note that times between measurements can be subdivided into multiple predictions if necessary for accureate modeling or computational savings.

5. At the time of the measurement, recalculate the weights on the particles according to the bootstrap weight update equation. Note that the use of the primed $w_i'(k)$ to indicate that it is not yet normalized.

$$w_i'(k) = p[z(k)|\chi_i(k)] \cdot w_i(k-1)$$

Very small likelihoods $p[z(k)|\chi_i(k)]$ may cause the numerical underflow problems in the particle filter, i.e. a weight $w_i'(k)$ might get set to zero because it is too small to represent in double precision. To be safe, the particles may be updated according to log-likelihoods:

$$log[w_i'(k)] = log[p[z(k)|\chi_i(k)]] + log[w_i(k-1)]$$
$$w_i''(k) = exp[log(w_i'(k)) - max_i[log(w_i'(k))]]$$

Where $w_i''(k)$ indicates that the doubly-primed weights are also not normalized and are different from $w_i'(k)$. In performing this log likelihood update, the value of the largest weight $max_i[log(w_i'(k))]$ has been subtracted from each weight before taking teh exponent. This scales all weights prior to the exponentiation for added numerical robustness. In the particular (and typical) case of zero-mean additive Gaussian white measurement noise, the weight update is particularly simple. That is, if $z(k) = h[x(k)] + w(k)$ with $w(k) \sim N(0, R)$, then:

$$log[w_i'(k)] = -\frac{1}{2}[z(k) - h(\chi_i(k))]^T R^{-1}(k)[z(k) - h(\chi_i(k))] + log[w_i(k-1)]$$
$$w_i''(k) = exp(log[w_i'(k)] - max_i(log[w_i'(k)]))$$

Notice in taking the log of the Gaussian likelihood, we drop the normalization constant. That constant is the same for all weights, so it gets cancelled when the weights are re-normalized later on.

6. Re-normalize the weights so they sum to unity. This is necessary in order to preserve the fact that the set of particles actually represent a discrete approximation to the posterior probability density of $x(k)$.

$$w_i(k) = \frac{w_i''(k)}{\sum_{i=1}^{N_s} w_i''(k)}$$

7. Evaluate the effective number of particles $\hat{\boldsymbol{N}}_s$:

$$\hat{\boldsymbol{N}}_s = \frac{1}{\sum_{i=1}^{N_s}(w_i(k))^2}$$

8. Resample the particles if the effective number $\hat{\boldsymbol{N}}_s$ is too low. A decent heuristic is to resample if $\hat{\boldsymbol{N}}_s < N_s/2$, but resampling more or less than that may also be justified. Here is a common resampling algorithm to be used when the particle filter needs to be resampled:
   (a) Choose a random number η on $[0, 1]$ uniformly
   (b) Find m such that $\sum_{j=1}^{m-1} w_j(k) \leq \eta < \sum_{j=1}^{m} w_j(k)$
   (c) Set $\chi_i^{new}(k) = \chi_m(k)$ and $w_i^{new}(k) = \frac{1}{N_s}$
   (d) Repeat steps a through c until $\chi_i^{new}(k), i = [1, N_s]$ are chosen
   (e) Delete the old set of particles and use the new set and new weights

   Note that some old particles might appear more than once in the new set, whereas others might disappear altogether.

9. Compute basic estimation statistics when desired (but don't throw out the particle set!)

$$\hat{\boldsymbol{x}}(k) \approx \sum_{i=1}^{N_s} w_i(k)\chi_i(k)$$
$$P(k) \approx \sum_{i=1}^{N_s} w_i(k)[\chi_i(k) - \hat{\boldsymbol{x}}(k)][\chi_i(k) - \hat{\boldsymbol{x}}(k)]^T$$

10. Return to step 2 and continue

Note:

Q: What is $p[z(k)|\chi_i(k)]$?
A: Suppose $z(k) = h[k, x(k)] + w(k), w(k) \sim N(0, R)$
Then $p[z(k)|\chi_i(k)] = N(z(k); h[\chi_i(k)], R)$ ■

Bibliography

[1] H. W. Sorenson, "Least-squares estimation: from Gauss to Kalman," *Spectrum, IEEE*, vol. 7, no. 7, pp. 63–68, 1970.

[2] S. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry and engineering.* Perseus books Group, 2001.

[3] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing.* New Jersey: Prentice Hall, 2000.

[4] M. Psiaki, "Backward-smoothing extended Kalman filter," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 5, pp. 885–894, 2005.

[5] M. Psiaki, "The blind tricyclist problem and a comparative study of nonlinear filters: A challenging benchmark for evaluating nonlinear estimation methods," *Control Systems, IEEE*, vol. 33, no. 3, pp. 40–54, 2013.