OPTIMAL STATE ESTIMATION

Kalman, H_∞, and Nonlinear Approaches
Solution Manual

Dan Simon
Cleveland State University

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Optimal State Estimation Solution Manual / Dan J. Simon.
p. cm.—
"Wiley-Interscience."
Includes bibliographical references and index.
ISBN xxxxxxxx
1. Kalman filtering. 2. H_∞ filtering. 3. Nonlinear filtering. I. Simon, Dan J.

HA31.2.S873 2004
001.4'33—dc22 2004044064
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CONTENTS

INTRODUCTION

This solution manual is a companion to the text *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*, by Dan Simon (John Wiley & Sons, 2006). The MATLAB[1] source code for the computer exercise solutions is given at the end of this solution manual. The references in this solution manual refer to the references section in the text *Optimal State Estimation*. The equation numbers in this solution manual refer to the equations in the book *Optimal State Estimation*.

Although the MATLAB code for the solutions is not available on the Internet, MATLAB-based source code for the examples in the text is available at the author's Web site.[2] The author's e-mail address is also available on the Web site, and I eagerly invite feedback, comments, suggestions for improvements, and corrections.

A note on notation

Three dots between delimiters (parenthesis, brackets, or braces) means that the quantity between the delimiters is the same as the quantity between the previous set of identical delimiters in the same equation. For example,

$$
\begin{aligned}
(A + BCD) + (\cdots)^T &= (A + BCD) + (A + BCD)^T \\
A + [B(C + D)]^{-1}E[\cdots] &= A + [B(C + D)]^{-1}E[B(C + D)]
\end{aligned}
$$

[1] MATLAB is a registered trademark of The MathWorks, Inc.

[2] http://academic.csuohio.edu/simond/estimation – if the Web site address changes, it should be easy to find with an Internet search.

CHAPTER 1

Linear systems theory

Problems

Written exercises

1.1 Find the rank of the matrix $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

Solution

The rank of a matrix A can be defined as the dimension of the largest submatrix consisting of rows and columns of A whose determinant is nonzero. With this definition we see that the rank of the zero matrix is zero.

1.2 Find two 2×2 matrices A and B such that $A \neq B$, neither A nor B are diagonal, $A \neq cB$ for any scalar c, and $AB = BA$. Find the eigenvectors of A and B. Note that they share an eigenvector. Interestingly, every pair of commuting matrices shares at least one eigenvector [Hor85, p. 51].

Solution

Suppose A and B are given as

$$\begin{aligned} A &= \begin{bmatrix} a_1 & a_2 \\ a_2 & a_3 \end{bmatrix} \\ B &= \begin{bmatrix} b_1 & b_2 \\ b_2 & b_3 \end{bmatrix} \end{aligned}$$

Then we see that

$$\begin{aligned} AB &= \begin{bmatrix} a_1b_1 + a_2b_2 & a_1b_2 + a_2b_3 \\ a_2b_1 + a_3b_2 & a_2b_2 + a_3b_3 \end{bmatrix} \\ BA &= \begin{bmatrix} a_1b_1 + a_2b_2 & a_2b_1 + a_3b_2 \\ a_1b_2 + a_2b_3 & a_2b_2 + a_3b_3 \end{bmatrix} \end{aligned}$$

We see that $AB = BA$ if $a_1b_2 + a_2b_3 = a_2b_1 + a_3b_2$. This will be true, for example, if $a_1 = 1$, $a_2 = 2$, $a_3 = 1$, $b_1 = 1$, $b_2 = 3$, and $b_3 = 1$. This gives

$$\begin{aligned} A &= \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \\ B &= \begin{bmatrix} 1 & 3 \\ 3 & 1 \end{bmatrix} \end{aligned}$$

For these matrices A has the eigenvalues -1 and 3, B has the eigenvalues -2 and 4, and both A and B have the eigenvectors $\begin{bmatrix} -1 & 1 \end{bmatrix}^T$ and $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$.

1.3 Prove the three identities of Equation (1.26).

Solution

a). Suppose A is an $n \times m$ matrix, and B is an $m \times p$ matrix. Then

$$\begin{aligned} (AB)^T &= \left(\begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nm} \end{bmatrix} \begin{bmatrix} B_{11} & \cdots & B_{1p} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mp} \end{bmatrix} \right)^T \\ &= \begin{bmatrix} \sum A_{1j}B_{j1} & \cdots & \sum A_{1j}B_{jp} \\ \vdots & \ddots & \vdots \\ \sum A_{nj}B_{j1} & \cdots & \sum A_{nj}B_{jp} \end{bmatrix}^T \\ &= \begin{bmatrix} \sum A_{1j}B_{j1} & \cdots & \sum A_{nj}B_{j1} \\ \vdots & \ddots & \vdots \\ \sum A_{1j}B_{jp} & \cdots & \sum A_{nj}B_{jp} \end{bmatrix} \\ B^TA^T &= \begin{bmatrix} B_{11} & \cdots & B_{m1} \\ \vdots & \ddots & \vdots \\ B_{1p} & \cdots & B_{mp} \end{bmatrix} \begin{bmatrix} A_{11} & \cdots & A_{n1} \\ \vdots & \ddots & \vdots \\ A_{1m} & \cdots & A_{nm} \end{bmatrix} \\ &= \begin{bmatrix} \sum B_{j1}A_{1j} & \cdots & \sum B_{j1}A_{nj} \\ \vdots & \ddots & \vdots \\ \sum B_{jp}A_{1j} & \cdots & \sum B_{jp}A_{nj} \end{bmatrix} \end{aligned}$$

QED

b). Suppose that $(AB)^{-1} = C$. Then $CAB = I$. Postmultiplying both sides of this equation by B^{-1} gives $CA = B^{-1}$. Postmultiplying both sides of this equation by A^{-1} gives $C = B^{-1}A^{-1}$. Hence we see that $(AB)^{-1} = B^{-1}A^{-1}$. QED

c). Suppose A is an $n \times m$ matrix, and B is an $m \times n$ matrix. Then

$$\begin{aligned}
\mathrm{Tr}(AB) &= \mathrm{Tr}\left(\begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nm} \end{bmatrix}\begin{bmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mn} \end{bmatrix}\right) \\
&= \mathrm{Tr}\left(\begin{bmatrix} \sum A_{1j}B_{j1} & \cdots & \sum A_{1j}B_{jn} \\ \vdots & \ddots & \vdots \\ \sum A_{nj}B_{j1} & \cdots & \sum A_{nj}B_{jn} \end{bmatrix}\right) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ji} \\
\mathrm{Tr}(BA) &= \mathrm{Tr}\left(\begin{bmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mn} \end{bmatrix}\begin{bmatrix} A_{11} & \cdots & A_{1m} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nm} \end{bmatrix}\right) \\
&= \mathrm{Tr}\left(\begin{bmatrix} \sum B_{1j}A_{j1} & \cdots & \sum B_{1j}A_{jm} \\ \vdots & \ddots & \vdots \\ \sum B_{mj}A_{j1} & \cdots & \sum B_{mj}A_{jm} \end{bmatrix}\right) \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n} B_{ij}A_{ji}
\end{aligned}$$

QED

1.4 Find the partial derivative of the trace of AB with respect to A.

Solution

Suppose A is an $n \times m$ matrix, and B is an $m \times n$ matrix. Then

$$\begin{aligned}
\frac{\partial \mathrm{Tr}(AB)}{\partial A} &= \frac{\partial}{\partial A}\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ji} \\
&= \begin{bmatrix} \frac{\partial}{\partial A_{11}}\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ji} & \cdots & \frac{\partial}{\partial A_{1m}}\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ji} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial A_{n1}}\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ji} & \cdots & \frac{\partial}{\partial A_{nm}}\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}B_{ji} \end{bmatrix} \\
&= \begin{bmatrix} B_{11} & \cdots & B_{m1} \\ \vdots & \ddots & \vdots \\ B_{1n} & \cdots & B_{mn} \end{bmatrix} \\
&= B^T
\end{aligned}$$

1.5 Consider the matrix

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

Recall that the eigenvalues of A are found by find the roots of the polynomial $P(\lambda) = |\lambda I - A|$. Show that $P(A) = 0$. (This is an illustration of the Cayley–Hamilton theorem [Bay99, Che99, Kai00].)

Solution

$$\begin{aligned}
P(\lambda) &= |\lambda I - A| \\
&= \begin{vmatrix} \lambda - a & -b \\ -b & \lambda - c \end{vmatrix} \\
&= \lambda^2 - (a+c)\lambda + ac - b^2 \\
P(A) &= A^2 - (a+c)A + (ac - b^2)I \\
&= \begin{bmatrix} a^2 + b^2 & ab + bc \\ ab + bc & b^2 + c^2 \end{bmatrix} - (a+c)\begin{bmatrix} a & b \\ b & c \end{bmatrix} + \begin{bmatrix} ac - b^2 & 0 \\ 0 & ac - b^2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

1.6 Suppose that A is invertible and

$$\begin{bmatrix} A & A \\ B & A \end{bmatrix} \begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Find B and C in terms of A [Lie67].

Solution

Multiplying out the matrix equation gives the following two equations.

$$\begin{aligned}
A^2 + AC &= 0 \\
BA + AC &= I
\end{aligned}$$

Solving for B and C in terms of A gives

$$\begin{aligned}
B &= A + A^{-1} \\
C &= -A
\end{aligned}$$

1.7 Show that AB may not be symmetric even though both A and B are symmetric.

Solution

Suppose the symmetric matrices A and B are given as

$$\begin{aligned}
A &= \begin{bmatrix} a_1 & a_2 \\ a_2 & a_3 \end{bmatrix} \\
B &= \begin{bmatrix} b_1 & b_2 \\ b_2 & b_3 \end{bmatrix}
\end{aligned}$$

Then we see that

$$AB = \begin{bmatrix} a_1b_1 + a_2b_2 & a_1b_2 + a_2b_3 \\ a_2b_1 + a_3b_2 & a_2b_2 + a_3b_3 \end{bmatrix}$$

AB is not symmetric if $a_1b_2 + a_2b_3 \neq a_2b_1 + a_3b_2$.

1.8 Consider the matrix

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

where a, b, and c are real, and a and c are nonnegative.

a) Compute the solutions of the characteristic polynomial of A to prove that the eigenvalues of A are real.

b) For what values of b is A positive semidefinite?

Solution

a). The characteristic polynomial of A is

$$\begin{aligned} P(\lambda) &= |\lambda I - A| \\ &= \begin{vmatrix} \lambda - a & -b \\ -b & \lambda - c \end{vmatrix} \\ &= \lambda^2 - (a+c)\lambda + ac - b^2 \end{aligned}$$

Finding the roots of this gives

$$\lambda = \frac{1}{2}\left[a + c \pm \sqrt{(a-c)^2 + 4b^2}\right]$$

The discriminant is non-negative so λ is real.
QED

b). In order for A to be positive semidefinite, its eigenvalues must be positive. The eigenvalues are

$$\lambda = \begin{cases} \frac{1}{2}\left[a + c + \sqrt{(a-c)^2 + 4b^2}\right] \\ \frac{1}{2}\left[a + c - \sqrt{(a-c)^2 + 4b^2}\right] \end{cases}$$

The first eigenvalue is always non-negative. The second eigenvalue is non-negative if $a + c \geq \sqrt{(a-c)^2 + 4b^2}$. Solving this equation gives $|b| \leq \sqrt{ac}$ as the condition of positive semidefiniteness.

1.9 Derive the properties of the state transition matrix given in Equation (1.72).

Solution

$$\frac{d}{dt}e^{At} = \frac{d}{dt}\sum_{j=0}^{\infty}\frac{(At)^j}{j!}$$

$$\begin{aligned}
&= \frac{d}{dt}\left[I + At + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots\right] \\
&= A + A^2t + \frac{A^3t^2}{2!} + \cdots \\
&= A\left[I + At + \frac{(At)^2}{2!} + \cdots\right] \\
&= Ae^{At}
\end{aligned}$$

This proves the first equality. After writing the third expression of the above sequence of equations, we can bring the common factor A out to the right to obtain

$$\begin{aligned}
\frac{d}{dt}e^{At} &= \left[I + At + \frac{(At)^2}{2!} + \cdots\right] A \\
&= e^{At}A
\end{aligned}$$

QED

1.10 Suppose that the matrix A has eigenvalues λ_i and eigenvectors v_i ($i = 1, \cdots, n$). What are the eigenvalues and eigenvectors of $-A$?

Solution

$Av_i = \lambda_i v_i$, therefore $-Av_i = -\lambda_i v_i$. From this we see that $-A$ has eigenvalues $-\lambda_i$ and eigenvectors v_i.

1.11 Show that $|e^{At}| = e^{|A|t}$ for any square matrix A.

Solution:

$$\begin{aligned}
e^{At} &= I + At + \frac{A^2t^2}{2} + \cdots \\
|e^{At}| &= |I| + |At| + \frac{|A^2t^2|}{2} + \cdots \\
&= |I| + |A|t + \frac{|A|^2t^2}{2} + \cdots \\
&= e^{|A|t}
\end{aligned}$$

QED

1.12 Show that if $\dot{A} = BA$, then

$$\frac{d|A|}{dt} = \mathrm{Tr}(B)|A|$$

Solution:

The equation $\dot{A} = BA$ can be solved as $A = e^{Bt}A(0)$. Taking the determinant of this equation gives

$$|A| = |e^{Bt}A(0)|$$

$$\begin{aligned} &= |e^{Bt}||A(0)| \\ &= e^{|B|t}|A(0)| \end{aligned}$$

From this we see that

$$\begin{aligned} \frac{d|A|}{dt} &= |B|e^{|B|t}|A(0)| \\ &= \mathrm{Tr}(B)|A| \end{aligned}$$

QED

1.13 The linear position p of an object under constant acceleration is

$$p = p_0 + \dot{p}t + \frac{1}{2}\ddot{p}t^2$$

where p_0 is the initial position of the object.

a) Define a state vector as $x = \begin{bmatrix} p & \dot{p} & \ddot{p} \end{bmatrix}^T$ and write the state space equation $\dot{x} = Ax$ for this system.

b) Use all three expressions in Equation (1.71) to find the state transition matrix e^{At} for the system.

c) Prove for the state transition matrix found above that $e^{A0} = I$.

Solution

a).

$$\frac{d}{dt}\begin{bmatrix} p \\ \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} p \\ \dot{p} \\ \ddot{p} \end{bmatrix}$$

b). From the first expression in Equation (1.72) we obtain

$$\begin{aligned} e^{At} &= \sum_{j=0}^{\infty}\frac{(At)^j}{j!} \\ &= \frac{(At)^0}{0!} + \frac{At}{1!} + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots \\ &= I + \begin{bmatrix} 0 & t & 0 \\ 0 & 0 & t \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & t^2/2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 0 + \cdots \\ &= \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

From the second expression in Equation (1.72) we obtain

$$\begin{aligned} e^{At} &= \mathcal{L}^{-1}[(sI - A)^{-1}] \\ &= \mathcal{L}^{-1}\begin{bmatrix} s & -1 & 0 \\ 0 & s & -1 \\ 0 & 0 & s \end{bmatrix}^{-1} \end{aligned}$$

$$= \mathcal{L}^{-1}\begin{bmatrix} 1/s & 1/s^2 & 1/s^3 \\ 0 & 1/s & 1/s^2 \\ 0 & 0 & 1/s \end{bmatrix}$$
$$= \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

From the third expression in Equation (1.72) we obtain

$$e^{At} = Qe^{\hat{A}t}Q^{-1}$$

The eigendata of A are found to be

$$\lambda = \{0, 0, 0\}$$
$$v = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

Actually we can note that A is already in Jordan form, which means that its eigenvalues are on the diagonal, and its eigenvectors form the identity matrix when augmented together. Recall for a third order Jordan block that

$$e^{At} = \begin{bmatrix} e^{\lambda t} & te^{\lambda t} & \frac{t^2}{2}e^{\lambda t} \\ 0 & e^{\lambda t} & te^{\lambda t} \\ 0 & 0 & e^{\lambda t} \end{bmatrix}$$

In our case $\lambda = 0$ so

$$e^{At} = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$$

c). From the above expression for e^{At}, if we substitute $t = 0$ we see that $e^{At} = I$. QED

1.14 Consider the following system matrix.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Show that the matrix

$$S(t) = \begin{bmatrix} e^t & 0 \\ 0 & 2e^{-t} \end{bmatrix}$$

satisfies the relation $\dot{S}(t) = AS(t)$, but $S(t)$ is not the state transition matrix of the system.

Solution

$$\dot{S}(t) = \begin{bmatrix} e^t & 0 \\ 0 & -2e^{-t} \end{bmatrix}$$

$$
\begin{aligned}
AS(t) &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} e^t & 0 \\ 0 & 2e^{-t} \end{bmatrix} \\
&= \begin{bmatrix} e^t & 0 \\ 0 & -2e^{-t} \end{bmatrix}
\end{aligned}
$$

We see that $\dot{S}(t) = AS(t)$. However, the state transition matrix is found to be

$$
e^{At} = \begin{bmatrix} e^t & 0 \\ 0 & e^{-t} \end{bmatrix}
$$

QED

1.15 Give an example of a discrete-time system that is marginally stable but not asymptotically stable.

Solution

The system $x_{k+1} = x_k$ is marginally stable, because the state is bounded for any initial bounded state, but it is not asymptotically stable, because it is not true that the state approaches zero for all initial states.

1.16 Show (H, F) is an observable matrix pair if and only if (H, F^{-1}) is observable (assuming that F is nonsingular).

Solution

If (H, F) is observable, then $Qx \neq 0$ for all nonzero x, where

$$
Q = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix}
$$

Since F is nonsingular $F^{-(n-1)}x$ spans the entire n-dimensional space. (That is, any n-element vector can be written as $F^{-(n-1)}x$ for some n-element vector x.) So the observability of (H, F) is equivalent to $QF^{-(n-1)}x \neq 0$ for all nonzero x. This is equivalent to $Q'x \neq 0$ for all nonzero x, where

$$
Q' = \begin{bmatrix} HF^{-(n-1)} \\ HF^{-(n-2)} \\ \vdots \\ H \end{bmatrix}
$$

which is the observability matrix of (H, F^{-1}).
QED

Computer exercises

1.17 The dynamics of a DC motor can be described as

$$
J\ddot{\theta} + F\dot{\theta} = T
$$

where θ is the angular position of the motor, J is the moment of inertia, F is the coefficient of viscous friction, and T is the torque applied to the motor.

a) Generate a two-state linear system equation for this motor in the form

$$\dot{x} = Ax + Bu$$

b) Simulate the system for 5 s and plot the angular position and velocity. Use $J = 10$ kg m^2, $F = 100$ kg m^2/s, $x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$, and $T = 10$ N m. Use rectangular integration with a step size of 0.05 s. Do the output plots look correct? What happens when you change the step size Δt to 0.2 s? What happens when you change the step size to 0.5 s? What are the eigenvalues of the A matrix, and how can you relate their magnitudes to the step size that is required for a correct simulation?

Solution

a). Let $x_1 = \theta$ and $x_2 = \dot{\theta}$. Then

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -F/J \end{bmatrix} + \begin{bmatrix} 0 \\ 1/J \end{bmatrix} T$$

b). Output plots for various simulation step sizes are shown in Figures 1.1–1.3. With $\Delta t = 0.05$ the simulation works fine. With $\Delta t = 0.2$ the simulation results are obviously incorrect, although the simulation is still stable. With $\Delta t = 0.5$ the simulation blows up. The eigenvalues of A are 0 and -10. The simulation step size should be appreciably smaller than $1/|\lambda|_{max}$, which implies that the step size should be smaller than $1/10$, which is consistent with our experimental results.

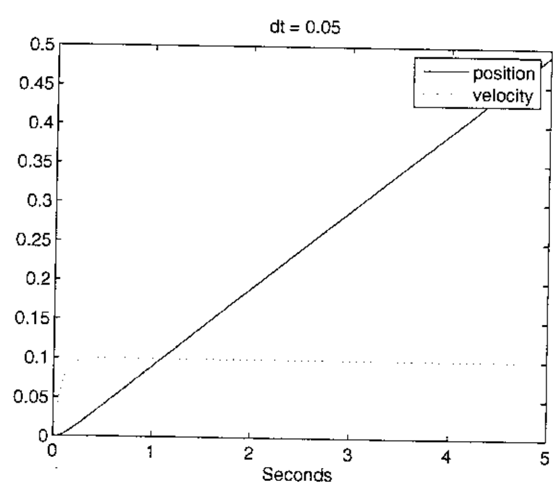

Figure 1.1 Problem 1.17 simulation with $\Delta t = 0.05$. Good simulation

1.18 The dynamic equations for a series RLC circuit can be written as

$$u = IR + L\dot{I} + V_c$$

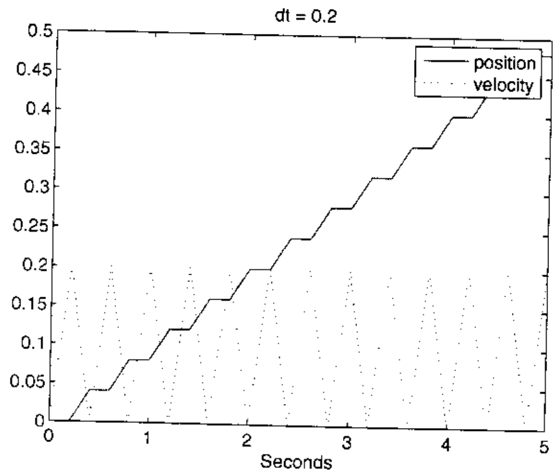

Figure 1.2 Problem 1.17 simulation with $\Delta t = 0.2$. Poor simulation

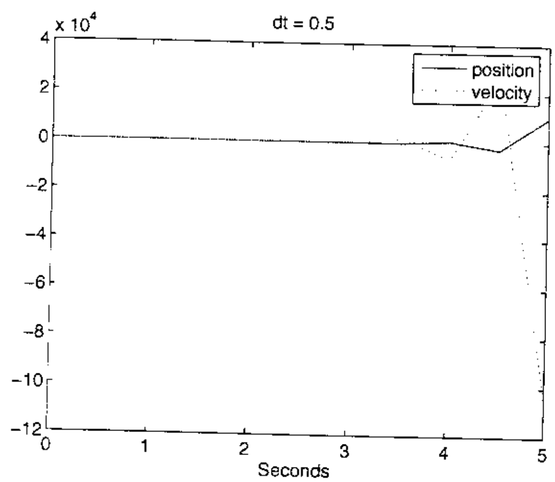

Figure 1.3 Problem 1.17 simulation with $\Delta t = 0.5$. Unstable simulation

$$I = C\dot{V}_c$$

where u is the applied voltage, I is the current through the circuit, and V_c is the voltage across the capacitor.

a) Write a state-space equation in matrix form for this system with x_1 as the capacitor voltage and x_2 as the current.

b) Suppose that $R = 3$, $L = 1$, and $C = 0.5$. Find an analytical expression for the capacitor voltage for $t \geq 0$, assuming that the initial state is zero, and the input voltage is $u(t) = e^{-2t}$.

c) Simulate the system using rectangular, trapezoidal, and fourth-order Runge–Kutta integration to obtain a numerical solution for the capacitor voltage.

Simulate from $t = 0$ to $t = 5$ using step sizes of 0.1 and 0.2. Tabulate the RMS value of the error between the numerical and analytical solutions for the capacitor voltage for each of your six simulations.

Solution

a).

$$\begin{aligned} u &= x_2 R + L\dot{x}_2 + x_1 \\ x_2 &= C\dot{x}_1 \end{aligned}$$

Putting this in matrix form gives

$$\dot{x} = \begin{bmatrix} 0 & 1/C \\ -1/L & -R/L \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/L \end{bmatrix} u$$

b).

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)} Bu(\tau)\, d\tau$$

Plugging in the values of R, L, and C into the A matrix and computing e^{At} gives

$$e^{At} = e^{-t} \begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix} + e^{-2t} \begin{bmatrix} -1 & -2 \\ 1 & 2 \end{bmatrix}$$

Substituting everything into the expression for $x(t)$ and computing the first element of $x(t)$ gives

$$\begin{aligned} x_1(t) &= 2\int_0^t \left[e^{-(t-\tau)} - e^{-2(t-\tau)} \right] e^{-2\tau}\, d\tau \\ &= 2(e^{-t} - e^{-2t} - te^{-2t}) \end{aligned}$$

c). Table 1.1 shows the RMS error of the numerical integration methods.

Table 1.1 Solution to Problem 1.18. RMS errors when numerically integrating the series RLC circuit, for various integration algorithms, and for various time step sizes T.

	$T = 0.1$	$T = 0.2$
Rectangular	0.016	0.035
Trapezoidal	0.012	0.024
Fourth order Runge Kutta	0.0018	0.0035

1.19 The vertical dimension of a hovering rocket can be modeled as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{Ku - gx_2}{x_3} - \frac{GM}{(R + x_1)^2} \\ \dot{x}_3 &= -u \end{aligned}$$

where x_1 is the vertical position of the rocket, x_2 is the vertical velocity, x_3 is the mass of the rocket, u is the control input (the flow rate of rocket propulsion), $K = 1000$ is the thrust constant of proportionality, $g = 50$ is the drag constant, $G = 6.673E-11$ m^3/kg/s^2 is the universal gravitational constant, $M = 5.98E24$ kg is the mass of the earth, and $R = 6.37E6$ m is the radius of the earth radius.

a) Find $u(t) = u_0(t)$ such that the system is in equilibrium at $x_1(t) = 0$ and $x_2(t) = 0$.

b) Find $x_3(t)$ when $x_1(t) = 0$ and $x_2(t) = 0$.

c) Linearize the system around the state trajectory found above.

d) Simulate the nonlinear system for five seconds and the linearized system for five seconds with $u(t) = u_0(t) + \Delta u \cos(t)$. Plot the altitude of the rocket for the nonlinear simulation and the linear simulation (on the same plot) when $\Delta u = 10$. Repeat for $\Delta u = 100$ and $\Delta u = 300$. Hand in your source code and your three plots. What do you conclude about the accuracy of your linearization?

Solution

a).

$$\begin{aligned}
\dot{x}_2 &= \frac{Ku - gx_2}{x_3} - \frac{GM}{(R+x_1)^2} \\
&= 0 \\
x_1 &= 0
\end{aligned}$$

Solving the above for $u(t)$ gives

$$u(t) = \frac{GMx_3}{KR^2}$$

b). From the third state equation and the equilibrium point obtained above we get

$$\dot{x}_3 = \frac{-GMx_3}{KR^2}$$

Solving for $x_3(t)$ gives

$$x_3(t) = x_3(0)\exp\left(\frac{-GMt}{KR^2}\right)$$

c). Use the notation $x_{10}(t) = 0$, $x_{20}(t) = 0$, $x_{30}(t) = x_3(0)\exp\left(\frac{-GMt}{KR^2}\right)$, and $u_0(t) = \frac{GMx_3}{KR^2}$ to denote the nominal trajectory.

$$\begin{aligned}
\Delta\dot{x}_1 &= \left.\frac{\partial \dot{x}_1}{\partial x_1}\Delta x_1 + \frac{\partial \dot{x}_1}{\partial x_2}\Delta x_2 + \frac{\partial \dot{x}_1}{\partial x_3}\Delta x_3 + \frac{\partial \dot{x}_1}{\partial u}\Delta u\right|_{x_{10}(t),x_{20}(t),x_{30}(t),u_0(t)} \\
&= \Delta x_2 \\
\Delta\dot{x}_2 &= \left.\frac{\partial \dot{x}_2}{\partial x_1}\Delta x_1 + \frac{\partial \dot{x}_2}{\partial x_2}\Delta x_2 + \frac{\partial \dot{x}_2}{\partial x_3}\Delta x_3 + \frac{\partial \dot{x}_2}{\partial u}\Delta u\right|_{x_{10}(t),x_{20}(t),x_{30}(t),u_0(t)} \\
&= \frac{2GM}{R^3}\Delta x_1 - \frac{g}{x_{30}(t)}\Delta x_2 - \frac{GM}{R^2 x_{30}(t)}\Delta x_3 + \frac{K}{x_{30}(t)}\Delta u
\end{aligned}$$

$$
\begin{aligned}
\Delta\dot{x}_3 &= \left.\frac{\partial \dot{x}_3}{\partial x_1}\Delta x_1 + \frac{\partial \dot{x}_3}{\partial x_2}\Delta x_2 + \frac{\partial \dot{x}_3}{\partial x_3}\Delta x_3 + \frac{\partial \dot{x}_3}{\partial u}\Delta u\right|_{x_{10}(t),x_{20}(t).x_{30}(t),u_0(t)} \\
&= -\Delta u
\end{aligned}
$$

d). Figure 1.4 shows simulation results for various values of Δu. As Δu increases the linearized simulation becomes less accurate (i.e., the linearized simulation does not track the nonlinear simulation as accurately).

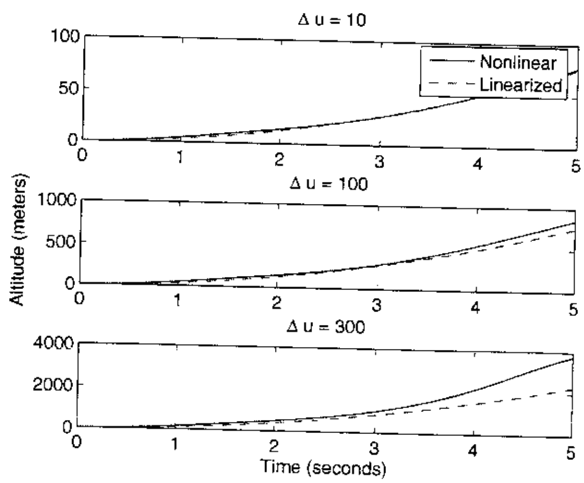

Figure 1.4 Rocket simulations for Problem 1.19

CHAPTER 2

Probability theory

Problems

Written exercises

2.1 What is the 0th moment of an RV? What is the 0th central moment of an RV?

Solution:

$$\begin{aligned}
i\text{th moment of } x &= E(x^i) \\
0\text{th moment of } x &= E(x^0) \\
&= E(1) \\
&= 1 \\
i\text{th central moment of } x &= E[(x-\bar{x})^i] \\
0\text{th central moment of } x &= E[(x-\bar{x})^0] \\
&= E(1) \\
&= 1
\end{aligned}$$

Optimal State Estimation Solution Manual, First Edition. By Dan J. Simon

2.2 Suppose a deck of 52 cards is randomly divided into four piles of 13 cards each. Find the probability that each pile contains exactly one ace [Gre01].

Solution:

Consider the first pile. There are a total of 52-choose-13 possible first piles. There are a total of 48-choose-12 different ways of selecting 12 non-Aces from the remaining 48 non-Ace cards. The odds that the first pile has exactly one Ace is therefore 4(48-choose-12)/(52-choose-13). If this event occurred we see that there are 39 cards remaining to be dealt, including three Aces. Therefore, the odds that the second pile has exactly one Ace is 3(36-choose-12)/(39-choose-13). If both of the previous events occurred we see that there are 26 cards remaining to be dealt, including two Aces. Therefore, the odds that the third pile has exactly one Ace is 2(24-choose-12)/(26-choose-13). Given that all three of the previous events occurred, the odds that the fourth pile has exactly one Ace is 1. Multiplying these odds together gives the total probability of 10.55%.

2.3 Determine the value of a in the function

$$f_X(x) = \begin{cases} ax(1-x) & x \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

so that $f_X(x)$ is a valid probability density function [Lie67].

Solution:

In order for $f_X(x)$ to be a valid pdf its integral from $-\infty$ to $+\infty$ must be equal to 1.

$$\begin{aligned} \int_{-\infty}^{\infty} f_X(x)\,dx &= \int_{-\infty}^{\infty} ax(1-x)\,dx \\ &= \frac{a}{6} \end{aligned}$$

Therefore $a = 6$.

2.4 Determine the value of a in the function

$$f_X(x) = \frac{a}{e^x + e^{-x}}$$

so that $f_X(x)$ is a valid probability density function. What is the probability that $|X| \leq 1$?

Solution:

In order for $f_X(x)$ to be a valid pdf its integral from $-\infty$ to $+\infty$ must be equal to 1.

$$\begin{aligned} \int_{-\infty}^{\infty} f_X(x)\,dx &= \int_{-\infty}^{\infty} \frac{a}{e^x + e^{-x}}\,dx \\ &= a\tan^{-1}(e^x)\Big|_{-\infty}^{\infty} \\ &= a\pi/2 \end{aligned}$$

Therefore $a = 2/\pi$. The probability that $|X| \leq 1$ is computed as

$$\begin{aligned} P(|X| \leq 1) &= a \tan^{-1}(e^x)\big|_{-1}^{1} \\ &= \frac{2}{\pi}\left(\tan^{-1} e - \tan^{-1} e^{-1}\right) \\ &\approx 0.55 \end{aligned}$$

2.5 The probability density function of an exponentially distributed random variable is defined as follows.

$$f_X(x) = \begin{cases} ae^{-ax} & x > 0 \\ 0 & x \leq 0 \end{cases}$$

where $a \geq 0$.

- **a)** Find the probability distribution function of an exponentially distributed random variable.
- **b)** Find the mean of an exponentially distributed random variable.
- **c)** Find the second moment of an exponentially distributed random variable.
- **d)** Find the variance of an exponentially distributed random variable.
- **e)** What is the probability that an exponentially distributed random variable takes on a value within one standard deviation of its mean?

Solution:

a).

$$\begin{aligned} P(x) &= \int_0^x ae^{-az}\, dz \\ &= \begin{cases} 1 - e^{-ax} & x > 0 \\ 0 & x \leq 0 \end{cases} \end{aligned}$$

b).

$$\bar{x} = \int_0^\infty xae^{-ax}\, dx$$

Using integration by parts we obtain

$$\begin{aligned} \bar{x} &= xe^{-ax}\big|_0^\infty + \int_0^\infty e^{-ax}\, dx \\ &= \frac{1}{a} \end{aligned}$$

c).

$$E(x^2) = \int_0^\infty x^2 ae^{-ax}\, dx$$

Using integration by parts we obtain

$$\begin{aligned} E(x^2) &= x^2 e^{-ax}\big|_0^\infty + \int_0^\infty 2xe^{-ax}\,dx \\ &= \frac{2}{a}\int_0^\infty axe^{-ax}\,dx \\ &= \frac{2}{a}\bar{x} \\ &= \frac{2}{a^2} \end{aligned}$$

d).

$$\begin{aligned} \sigma^2 &= E(x^2) - \bar{x}^2 \\ &= \frac{1}{a^2} \end{aligned}$$

e).

$$\begin{aligned} P(\bar{x} - \sigma \le x \le \bar{x} + \sigma) &= \int_0^{2/a} ae^{-ax}\,dx \\ &= 1 - e^{-2} \\ &\approx 0.86 \end{aligned}$$

2.6 Derive an expression for the skew of a random variable as a function of its first, second, and third moments.

Solution:

$$\begin{aligned} \text{skew} &= E[(x-\bar{x})^3] \\ &= E[x^3 - 3x^2\bar{x} + 3x\bar{x}^2 - \bar{x}^3] \\ &= E(x^3) - 3\bar{x}E(x^2) + 2\bar{x}^3 \end{aligned}$$

2.7 Consider the following probability density function:

$$f_X(x) = \frac{ab}{b^2 + x^2}, \qquad b > 0$$

a) Determine the value of a in the so that $f_X(x)$ is a valid probability density function. (The correct value of a makes $f_X(x)$ a Cauchy pdf.)

b) Find the mean of a Cauchy random variable.

Solution:

a). In order for $f_X(x)$ to be a valid pdf its integral from $-\infty$ to $+\infty$ must be equal to 1.

$$\int_{-\infty}^{\infty} \frac{ab}{b^2 + x^2}\,dx = a\tan^{-1}(x)\big|_{-\infty}^{\infty}$$

$$
\begin{aligned}
&= a\left(\frac{\pi}{2}+\frac{\pi}{2}\right) \\
&= a\pi
\end{aligned}
$$

Therefore $a = 1/\pi$.

b).

$$
\begin{aligned}
E(X) &= \int_{-\infty}^{\infty} x f_X(x)\,dx \\
&= \frac{b}{\pi}\int_{-\infty}^{\infty} \frac{x}{b^2+x^2}\,dx \\
&= \frac{b}{2\pi}\ln\left|b^2+x^2\right|_{-\infty}^{\infty} \\
&= \infty
\end{aligned}
$$

This indicates that the mean is ∞, but what it really says is that the integral does not converge to a real number, so the mean of a Cauchy random variable does not exist.

2.8 Consider two zero-mean uncorrelated random variables W and V with standard deviations σ_w and σ_v, respectively. What is the standard deviation of the random variable $X = W + V$?

Solution:

$$
\begin{aligned}
\sigma_x^2 &= E[(X-\bar{x})^2 \\
&= E[(W+V-\bar{w}-\bar{v})^2 \\
&= E[(W+V)^2] \\
&= E(W^2)+E(V^2)+2E(WV) \\
&= \sigma_w^2+\sigma_v^2+0 \\
\sigma_x &= \sqrt{\sigma_w^2+\sigma_v^2}
\end{aligned}
$$

2.9 Consider two scalar RVs X and Y.

a) Prove that if X and Y are independent, then their correlation coefficient $\rho = 0$.

b) Find an example of two RVs that are not independent but that have a correlation coefficient of zero.

c) Prove that if Y is a linear function of X then $\rho = \pm 1$.

Solution:

a). If X and Y are independent then $E[(X-\bar{X})(Y-\bar{Y})] = E(X-\bar{X})E(Y-\bar{Y})$. Therefore

$$
\rho = \frac{C_{XY}}{\sigma_x\sigma_y}
$$

$$
\begin{aligned}
&= \frac{E(X-\bar{X})E(Y-\bar{Y})}{\sigma_x\sigma_y} \\
&= \frac{[\bar{X}-\bar{X}][\bar{Y}-\bar{Y}]}{\sigma_x\sigma_y} \\
&= 0
\end{aligned}
$$

b). Suppose the discrete RV X has a 25% probability of being -1, a 25% probability of being $+1$, and a 50% probability of being 0. This means that $\bar{X} = 0$. Suppose that $Y = |X|$ so that X and Y are clearly dependent, and $E(XY) = E(X|X|) = 0$. Then

$$
\begin{aligned}
C_{XY} &= E(XY) - \bar{X}\bar{Y} \\
&= 0
\end{aligned}
$$

which means that $\rho = 0$.

c). If $Y = aX$ for some constant a then

$$
\begin{aligned}
\rho &= \frac{E[(X-\bar{X})(Y-\bar{Y}]}{\sigma_x\sigma_y} \\
&= \frac{aE[(X-\bar{X})^2]}{\sqrt{E[(X-\bar{X})^2]}\sqrt{E[(aX-a\bar{X})^2]}} \\
&= \frac{a}{|a|} \\
&= \pm 1
\end{aligned}
$$

If a is positive (negative) then $\rho = 1$ (-1).

2.10 Consider the following function [Lie67].

$$
f_{XY}(x,y) = \begin{cases} ae^{-2x}e^{-3y} & x > 0, y > 0 \\ 0 & \text{otherwise} \end{cases}
$$

a) Find the value of a so that $f_{XY}(x,y)$ is a valid joint probability density function.

b) Calculate $\bar{x}$ and $\bar{y}$.

c) Calculate $E(X^2)$, $E(Y^2)$, and $E(XY)$.

d) Calculate the autocorrelation matrix of the random vector $\begin{bmatrix} X & Y \end{bmatrix}^T$.

e) Calculate the variance σ_x^2, the variance σ_y^2, and the covariance C_{XY}.

f) Calculate the autocovariance matrix of the random vector $\begin{bmatrix} X & Y \end{bmatrix}^T$.

g) Calculate the correlation coefficient between X and Y.

Solution:

a). In order for $f_{XY}(x,y)$ to be a valid pdf its double integral from $-\infty$ to $+\infty$ must be equal to 1.

$$
\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\,dx\,dy = \int_0^{\infty}\int_0^{\infty} ae^{-2x}e^{-3y}\,dx\,dy
$$

$$\begin{aligned} &= a \int_0^\infty e^{-2x}\, dx \int_0^\infty e^{-3y}\, dy \\ &= \frac{a}{6} \end{aligned}$$

Therefore $a = 6$.

b).

$$\begin{aligned} \bar{x} &= \int_{-\infty}^{\infty} x f_X(x)\, dx \\ &= \int_{-\infty}^{\infty} x \left[\int_{-\infty}^{\infty} f_{XY}(x,y)\, dy \right] dx \\ &= \int_0^\infty x \left[\int_0^\infty 6e^{-2x} e^{-3y}\, dy \right] dx \\ &= 6 \int_0^\infty x e^{-2x} \left[\int_0^\infty e^{-3y}\, dy \right] dx \\ &= 2 \int_0^\infty x e^{-2x}\, dx \\ &= -e^{-2x} \left(x + \frac{1}{2} \right) \Big|_0^\infty \\ &= \frac{1}{2} \end{aligned}$$

Similarly, we can obtain

$$\bar{y} = \frac{1}{3}$$

c).

$$\begin{aligned} E(X^2) &= \int_{-\infty}^{\infty} x^2 f_X(x)\, dx \\ &= \int_{-\infty}^{\infty} x^2 \left[\int_{-\infty}^{\infty} f_{XY}(x,y)\, dy \right] dx \\ &= \int_0^\infty 2x^2 e^{-2x}\, dx \\ &= -e^{-2x} \left(x^2 + x + \frac{1}{2} \right) \Big|_0^\infty \\ &= \frac{1}{2} \end{aligned}$$

Similarly, we can obtain

$$\begin{aligned} E(Y^2) &= \frac{2}{9} \\ E(XY) &= \frac{1}{6} \end{aligned}$$

d).

$$\begin{aligned} R_{XY} &= \begin{bmatrix} E(X^2) & E(XY) \\ E(XY) & E(Y^2) \end{bmatrix} \\ &= \begin{bmatrix} 1/2 & 1/6 \\ 1/6 & 2/9 \end{bmatrix} \end{aligned}$$

e).

$$\begin{aligned} \sigma_X^2 &= E(X^2) - \bar{x}^2 \\ &= \frac{1}{4} \\ \sigma_Y^2 &= E(Y^2) - \bar{y}^2 \\ &= \frac{1}{9} \\ C_{XY} &= E(XY) - \bar{x}\bar{y} \\ &= 0 \end{aligned}$$

f).

$$\begin{aligned} C &= \begin{bmatrix} E((X-\bar{x})^2) & E((X-\bar{x})(Y-\bar{y})) \\ E((X-\bar{x})(T-\bar{y})) & E((T-\bar{y})^2) \end{bmatrix} \\ &= \begin{bmatrix} 1/4 & 0 \\ 0 & 1/9 \end{bmatrix} \end{aligned}$$

g).

$$\begin{aligned} \rho &= \frac{C_{XY}}{\sigma_x \sigma_y} \\ &= 0 \end{aligned}$$

2.11 A stochastic process has the autocorrelation $R_X(\tau) = Ae^{-k|\tau|}$, where A and k are positive constants.

a) What is the power spectrum of the stochastic process?

b) What is the total power of the stochastic process?

c) What value of k results in half of the total power residing in frequencies less than 1 Hz?

Solution:

a).

$$\begin{aligned} S_X(\omega) &= \int_{-\infty}^{\infty} R_X(\tau) e^{-j\omega\tau}\, d\tau \\ &= \int_{-\infty}^{0} Ae^{k\tau} e^{-j\omega\tau}\, d\tau + \int_{0}^{\infty} Ae^{-k\tau} e^{-j\omega\tau}\, d\tau \end{aligned}$$

$$
\begin{aligned}
&= \int_{-\infty}^{0} Ae^{(k-j\omega)\tau}\, d\tau + \int_{0}^{\infty} Ae^{-(k+j\omega)\tau}\, d\tau \\
&= \frac{2Ak}{k^2+\omega^2}
\end{aligned}
$$

b).

$$
\begin{aligned}
P_X &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(\omega)\, d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{2Ak}{k^2+\omega^2}\, d\omega \\
&= \frac{A}{\pi}\tan^{-1}\left(\frac{\omega}{k}\right)\Big|_{-\infty}^{\infty} \\
&= A
\end{aligned}
$$

c). The power at frequencies less than 1 Hz (which is 2π rad/sec) can be written as

$$
\begin{aligned}
P_1 &= \frac{A}{\pi}\tan^{-1}\left(\frac{\omega}{k}\right)\Big|_{-2\pi}^{2\pi} \\
&= \frac{2A}{\pi}\tan^{-1}\left(\frac{2\pi}{k}\right)
\end{aligned}
$$

Setting this equal to $A/2$ gives

$$
\begin{aligned}
\tan^{-1}\left(\frac{2\pi}{k}\right) &= \frac{\pi}{4} \\
\frac{2\pi}{k} &= \tan\left(\frac{\pi}{4}\right) \\
&= 1
\end{aligned}
$$

Therefore $k = 2\pi$.

2.12 Suppose X is a random variable, and $Y(t) = X\cos t$ is a stochastic process.

a) Find the expected value of $Y(t)$.

b) Find $A[Y(t)]$, the time average of $Y(t)$.

c) Under what condition is $\bar{y}(t) = A[Y(t)]$?

Solution:

a).

$$
\begin{aligned}
\bar{Y}(t) &= E[X\cos t] \\
&= \bar{x}\cos t
\end{aligned}
$$

b).

$$
A[Y(t)] = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} y(t)\, dt
$$

$$
\begin{aligned}
&= \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x \cos t \, dt \\
&= 0
\end{aligned}
$$

c). $\bar{y}(t) = A[Y(t)]$ if and only if $\bar{x} = 0$.

2.13 Consider the equation $Z = X + V$. The pdf's of X and B are given in Figure 2.1.

a) Plot the pdf of $(Z|X)$ as a function of X for $Z = 0.5$.

b) Given $Z = 0.5$, what is conditional expectation of X? What is the most probable value of X? What is the median value of X?

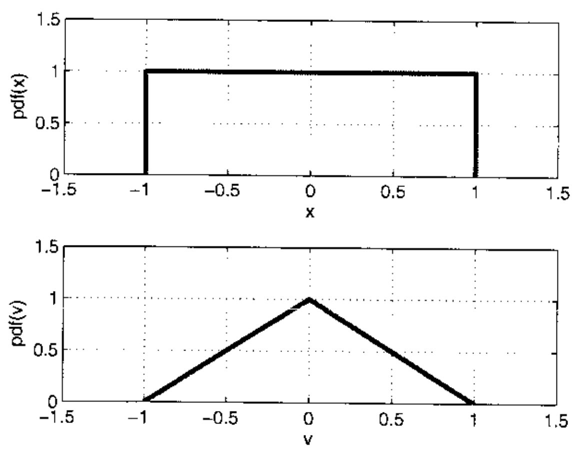

Figure 2.1 pdf's for Problem 2.13 [Sch73].

Solution:

a).

$$
\begin{aligned}
P(Z = 0.5|X = x_1) &= P(X + V = 0.5|X = x_1) \\
&= P(V = 0.5 - x_1)
\end{aligned}
$$

pdf$(Z = 0.5|x)$ is therefore equal to $f_V(v)$ shifted to the right by 0.5, except that the pdf must be truncated at $x = 1$ since x cannot be greater than 1. The truncation results in a scaling of the pdf by a factor of $8/7$ in order to give the pdf an area of 1. This is shown in Figure 2.2

b). From pdf$(Z = 0.5|x)$ we see that the conditional expectation of X is

$$
\begin{aligned}
\hat{x} &= \int_{-\infty}^{\infty} x \, \text{pdf}(Z = 0.5|x) \, dx \\
&= \int_{-1/2}^{1/2} x \left(\frac{8}{7}x + \frac{4}{7} \right) dx + \int_{1/2}^{1} x \left(\frac{-8}{7}x + \frac{12}{7} \right) dx \\
&= 17/42
\end{aligned}
$$

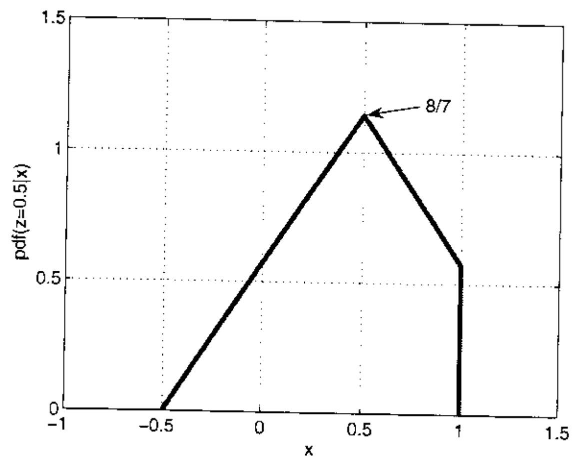

Figure 2.2 Solution to Problem 2.13.

The most probable value of X is the peak of the pdf($Z = 0.5|x$) curve, which is

$$\hat{x} = 1/2$$

The median value of X is the midpoint on the x-axis of the pdf($Z = 0.5|x$) curve, which is

$$\hat{x} = 1/4$$

2.14 The temperature at noon in London is a stochastic process. Is it ergodic?

Solution:

A stochastic process is ergodic only if its time average is equal to the mean of the time-varying RV at each moment in time. Since the temperature in London, considered as an RV, has a different mean at different times of the year, it is not ergodic.

Computer exercises

2.15 Generate $N = 50$ independent random numbers, each uniformly distributed between 0 and 1. Plot a histogram of the random numbers using 10 bins. What is the sample mean and standard deviation of the numbers that you generated? What would you expect to see for the mean and standard deviation (i.e., what are the theoretical mean and standard deviation)? Repeat for $N = 500$ and $N = 5,000$ random numbers. What changes in the histogram do you see as N increases?

Solution:

The solution is given in Table 2.1 and Figure 2.3.

Table 2.1 Solution to Problem 2.15. Your results may vary depending on the particular random number sequence that you generated. In general, the sample mean and standard deviation get closer to the theoretical values as N increases.

	Theoretical	$N = 50$	$N = 500$	$N = 5000$
Mean	0.500	0.565	0.476	0.504
Standard Deviation	$\sqrt{1/12} \approx 0.289$	0.299	0.297	0.288

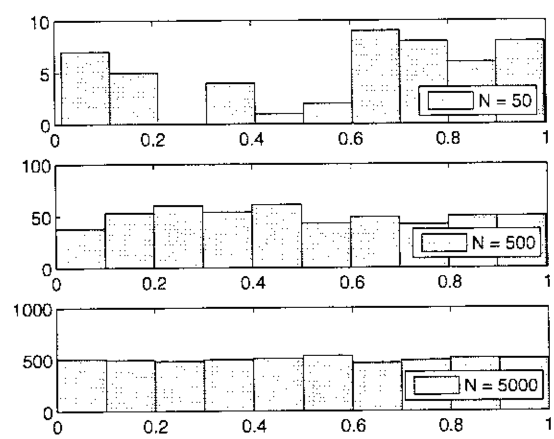

Figure 2.3 Solution to Problem 2.15. As N increases, the histogram gets closer to a truly uniform distribution. That is, the sample pdf approaches the theoretical pdf.

2.16 Generate 10,000 samples of $(x_1 + x_2)/2$, where each x_i is a random number uniformly distributed on $[-1/2, +1/2]$. Plot the 50-bin histogram. Repeat for $(x_1 + x_2 + x_3 + x_4)/4$. Describe the difference between the two histograms.

Solution:

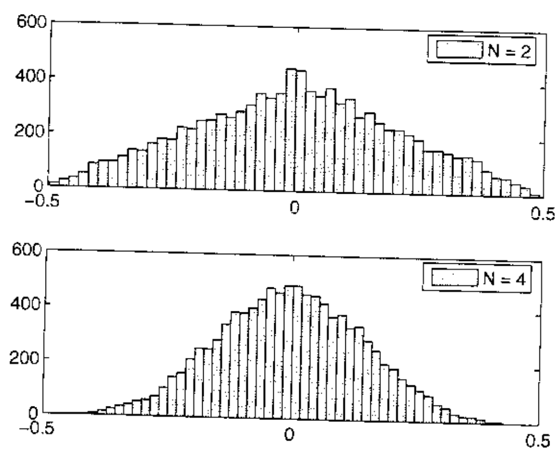

Figure 2.4 For $N = 2$ the histogram looks like a triangular pdf. For $N = 4$ the histogram looks more like a Gaussian pdf.

CHAPTER 3

Least squares estimation

Problems

Written exercises

3.1 In Equation (3.6) we computed the partial derivative of our cost function with respect to our estimate and set the result equal to 0 to solve for the optimal estimate. However, the solution minimizes the cost function only if the second derivative of the cost function with respect to the estimate is positive semidefinite. Find the second derivative of the cost function and show that it is positive semidefinite.

Solution:

$$\frac{\partial^2 J}{\partial^2 \hat{x}} = 2H^T H$$

This is positive semidefinite because for any vector y we have

$$\begin{aligned} y^T(2H^T H)y &= 2(Hy)^T(Hy) \\ &= 2||Hy||_2^2 \\ &\geq 0 \end{aligned}$$

QED

3.2 Prove that the matrix P_k that is computed from Equation (3.25) will always be positive definite if P_{k-1} and R_k are positive definite.

Solution

$$P_k = (I - K_k H_k)P_{k-1}(I - K_k H_k)^T + K_k R_k K_k^T$$

Let a be an arbitrary vector of correct dimensions so that $aP_k a^T$ is a scalar. Then

$$\begin{aligned} aP_k a^T &= a(I - K_k H_k)P_{k-1}(I - K_k H_k)^T a^T + aK_k R_k K_k^T a^T \\ &= bP_{k-1}b^T + cR_k c^T \end{aligned}$$

where b and c are defined by the above equation. If P_{k-1} and R_k are both positive definite, then the above expression is positive for all b and c, which means it is positive for all a, which means that P_k is positive definite.
QED

3.3 Consider the recursive least squares estimator of Equations (3.28)-(3.30). If zero information about the initial state is available, then $P_0 = \infty I$. Suppose that you have a system like this with $H_k = 1$. What will be the values of K_1 and P_1?

Solution:

$$\begin{aligned} K_1 &= \lim_{P_0 \to \infty} \frac{P_0}{P_0 + R} \\ &= 1 \\ P_1 &= (1 - K_1)^2 P_0 + K_1^2 R \\ &= R \end{aligned}$$

3.4 Consider a battery with a completely unknown voltage ($P_0 = \infty$). Two independent measurements of the voltage are taken to estimate the voltage, the first with a variance of 1, and the second with a variance of 4.

a) Write the weighted least squares voltage estimate in terms of the two measurements y_1 and y_2.

b) If weighted least squares is used to estimate the voltage, what is the variance of voltage estimate after the first measurement? What is the variance of the voltage estimate after the second measurement?

c) If the voltage is estimated as $(y_1 + y_2)/2$, an unweighted average of the measurements, what is the variance of the voltage estimate?

Solution:

a). From Equations (3.28)-(3.30) we calculate

$$\hat{x} = \frac{4}{5}y_1 + \frac{1}{5}y_2$$

b). From Equations (3.28)-(3.30) we calculate

$$
\begin{aligned}
P_1 &= 1 \\
P_2 &= 4/5
\end{aligned}
$$

c).

$$
\begin{aligned}
E\left\{[(y_1 - y_2)/2 - V]^2\right\} &= E[(v_1/2)^2 + (v_2/2)^2] \\
&= 1/4 + 4/4 \\
&= 5/4
\end{aligned}
$$

3.5 Consider a battery whose voltage is a random variable with a variance of 1. Two independent measurements of the voltage are taken to estimate the voltage, the first with a variance of 1, and the second with a variance of 4.

- **a)** Write the weighted least squares voltage estimate in terms of the initial estimate $\hat{x}_0$ and the two measurements y_1 and y_2.
- **b)** If weighted least squares is used to estimate the voltage, what is the variance of voltage estimate after the first measurement? What is the variance of the voltage estimate after the second measurement?

Solution:

a). From Equations (3.28)-(3.30) we calculate

$$
\hat{x} = \frac{4}{9}\hat{x}_0 + \frac{4}{9}y_1 + \frac{1}{9}y_2
$$

b). From Equations (3.28)-(3.30) we calculate

$$
\begin{aligned}
P_1 &= 1/2 \\
P_2 &= 4/9
\end{aligned}
$$

3.6 Suppose that $\{x_1, x_2, \cdots, x_n\}$ is a set of random variables, each with mean $\bar{x}$ and variance σ^2. Further suppose that $E[(x_i - \bar{x})(x_j - \bar{x})] = 0$ for $i \neq j$. We estimate $\bar{x}$ and σ^2 as follows.

$$
\begin{aligned}
\hat{\bar{x}} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\
\hat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{\bar{x}})^2
\end{aligned}
$$

- **a)** Is $\hat{\bar{x}}$ an unbiased estimate of $\bar{x}$? That is, is $E(\hat{\bar{x}}) = \bar{x}$?
- **b)** Find $E(x_i x_j)$ in terms of $\bar{x}$ and σ^2 for both $i = j$ and $i \neq j$.
- **c)** Is $\hat{\sigma}^2$ an unbiased estimate of σ^2? That is, is $E(\hat{\sigma}^2) = \sigma^2$? If not, how should we change $\hat{\sigma}^2$ to make it an unbiased estimate of σ^2?

Solution

a).

$$
\begin{aligned}
E(\hat{\bar{x}}) &= \frac{1}{n}\sum_{i=1}^{n} E(x_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} \bar{x} \\
&= \bar{x}
\end{aligned}
$$

So yes, $\hat{\bar{x}}$ an unbiased estimate of $\bar{x}$.

b). For $i = j$ we obtain

$$
\begin{aligned}
\sigma^2 &= E[(x_i - \bar{x})^2] \\
&= E(x_i^2) - \bar{x}^2 \\
E(x_i^2) &= \sigma^2 + \bar{x}^2
\end{aligned}
$$

For $i \neq j$ the problem statement tells us that $E[(x_i - \bar{x})(x_j - \bar{x})] = 0$. Therefore

$$
\begin{aligned}
E(x_i x_j) - \bar{x}^2 &= 0 \\
E(x_i x_j) &= \bar{x}^2
\end{aligned}
$$

c).

$$
\begin{aligned}
E(\hat{\sigma}^2) &= E\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\bar{x}})^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} E\left[x_i^2 - 2\hat{\bar{x}}x_i + \hat{\bar{x}}^2\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} E(x_i^2) - \frac{2}{n}\sum_{i,j=1}^{n} E(x_i x_j) + \frac{1}{n}\sum_{i,j,k=1}^{n} E(x_k x_j)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} E(x_i^2) - \frac{1}{n}\sum_{i,j=1}^{n} E(x_i x_j)\right]
\end{aligned}
$$

Now use the fact that $E(x_i^2) = \sigma^2 + \bar{x}^2$, and $E(x_i x_j) = \bar{x}^2$ for $i \neq j$, to simplify the above expression.

$$
\begin{aligned}
E(\hat{\sigma}^2) &= \frac{1}{n}\left[n(\sigma^2 + \bar{x}^2) - \frac{1}{n}[n(\sigma^2 + \bar{x}^2) + n(n-1)\bar{x}^2]\right] \\
&= \frac{(n-1)\sigma^2}{n}
\end{aligned}
$$

We see that $\hat{\sigma}^2$ an not an unbiased estimate of σ^2. Based on the above derivation it is apparent that

$$
\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\bar{x}})^2
$$

is an unbiased estimate of σ^2.

3.7 Suppose a scalar signal has the values 1, 2, and 3. Consider three different estimates of this time-varying signal. The first estimate is 3, 4, 1. The second estimate is 1, 2, 6. The third estimate is 5, 6, 7. Create a table showing the RMS value, average absolute error, and standard deviation of the error of each estimate. Which estimate results in the error with the smallest RMS value? Which estimate results in the error with the smallest infinity-norm? Which estimate gives the error with the smallest standard deviation? Which estimate do you think is best from an intuitive point of view? Which estimate do you think is worst from an intuitive point of view?

Solution:

For Estimate #1 the errors are 2, 2, and -2. We therefore find

$$\begin{aligned}
\text{RMS Error} &= \sqrt{(2^2 + 2^2 + (-2)^2)/3} = 2 \\
\text{Ave Abs. Error} &= (|2| + |2| + |-2|)/3 = 2 \\
\text{Average Error} &= (2 + 2 - 2)/3 = 2/3 \\
\text{Std. Dev.} &= \sqrt{[(2 - 2/3)^2 + (2 - 2/3)^2 + (-2 - 2/3)^2]/3} = 4\sqrt{2}/3
\end{aligned}$$

Note that your standard deviation measurement will be different if you normalize by $(n - 1)$ (which is 2 in this case) instead of n. The corresponding quantities for the other estimates are found similarly.

Table 3.1 shows that Estimate #2 gives the smallest RMS error, Estimate #1 gives the smallest average absolute error, and Estimate #3 gives the error with the smallest standard deviation. The error signals are $\{2, 2, -2\}$ for Estimate #1, $\{0, 0, 3\}$ for Estimate #2, and $\{4, 4, 4\}$ for Estimate #3. Intuitively it looks like Estimate #2 is best and Estimate #3 is worst; however, this is a matter of opinion because the term "best" is ambiguous.

Table 3.1 Solution to Problem 3.7.

	RMS error	Ave. abs. error	Std. dev. of error
Estimate #1	2	2	1.8856
Estimate #2	1.7321	3	1.4142
Estimate #3	4	4	0

3.8 Suppose a random variable x has the pdf $f(x)$ given in Figure 3.1.

a) x can be estimated by taking the median of its pdf. That is, $\hat{x}$ is the solution to the equation

$$\int_{-\infty}^{\hat{x}} f(x)\,dx = \int_{\hat{x}}^{\infty} f(x)\,dx$$

Find the median estimate of x.

b) x can be estimated by taking the mode of its pdf. That is,

$$\hat{x} = \arg\max f(x)$$

Find the mode estimate of x.

c) x can be estimated by computing its mean. That is,

$$\hat{x} = \int_{-\infty}^{\infty} x f(x)\, dx$$

Find the mean of x.

d) x can be estimated by computing the minimax value. That is.

$$\hat{x} = \min \max_{x} |x - \hat{x}|$$

Find the minimax estimate of x.

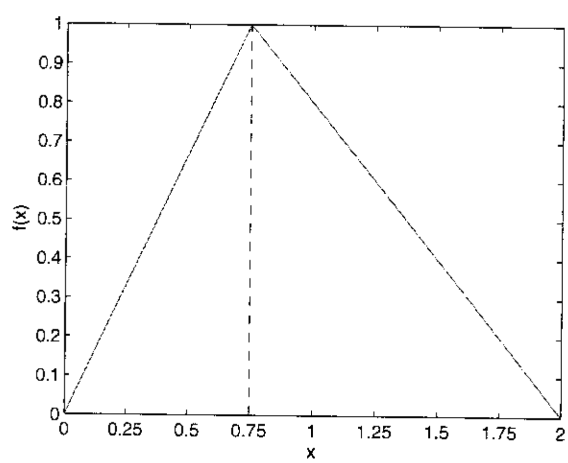

Figure 3.1 pdf for Problem 3.8.

Solution:

From Figure 3.1 we see that the pdf $f(x)$ is given as

$$f(x) = \begin{cases} 4x/3 & x \in [0, 3/4] \\ 8/5 - 4x/5 & x \in [3/4, 2] \\ 0 & \text{otherwise} \end{cases}$$

a). It should be clear from Figure 3.1 that the median estimate of x is greater than 3/4. Also, the total area of the pdf is 1. We can therefore solve for the median of the pdf as

$$\int_{\hat{x}}^{2} (8/5 - 4x/5)\, dx = 1/2$$

$$
\begin{aligned}
8/5 + 2\hat{x}^2/5 - 8\hat{x}/5 &= 1/2 \\
\hat{x} &= 2 - \sqrt{5}/2 \\
&\approx 0.8820
\end{aligned}
$$

b). The mode of $f(x)$ is that value of x where $f(x)$ takes its maximum value. From Figure 3.1 we see that this is

$$\hat{x} = 3/4$$

c). The mean of x is computed as

$$
\begin{aligned}
\hat{x} &= \int_{-\infty}^{\infty} x f(x)\, dx \\
&= \int_0^{3/4} 4x^2/3\, dx + \int_{3/4}^{2} x\,(8/5 - 4x/5)\, dx \\
&= 99/108 \\
&\approx 0.9167
\end{aligned}
$$

d). The minimax estimate of x is that value of x that minimizes the maximum possible estimation error. From Figure 3.1 we see that this is given as

$$\hat{x} = 1$$

3.9 Suppose you are responsible for increasing the tracking accuracy of a radar system. You presently have a radar that has a measurement variance of 10. For equal cost you could either: (a) optimally combine the present radar system with a new radar system that has a measurement variance of 6; or, (b) optimally combine the present radar system with two new radar systems that both have the same performance as the original system [May79]. Which would you propose to do? Why?

Solution:

Assume that the initial position uncertainty of the target is infinite – that is, $P_0 = \infty$. For option (a) we have $R_1 = 10$ and $R_2 = 6$. Equation (3.30) gives

$$
\begin{aligned}
K_1 &= \frac{P_0}{P_0 + R_1} \\
&= 1 \\
P_1 &= (1 - K_1)^2 P_0 + K_1^2 R_1 \\
&= R_1 \\
K_2 &= \frac{P_1}{P_1 + R_2} \\
&= 5/8 \\
P_2 &= (1 - K_2)^2 P_1 + K_2^2 R_2 \\
&= 3.75
\end{aligned}
$$

For option (b) we have $R_1 = R_2 = R_3 = 10$. Equation (3.30) gives

$$\begin{aligned}
K_1 &= \frac{P_0}{P_0 + R_1} \\
&= 1 \\
P_1 &= (1 - K_1)^2 P_0 + K_1^2 R_1 \\
&= R_1 \\
K_2 &= \frac{P_1}{P_1 + R_2} \\
&= 1/2 \\
P_2 &= (1 - K_2)^2 P_1 + K_2^2 R_2 \\
&= 5 \\
K_3 &= \frac{P_2}{P_2 + R_3} \\
&= 1/3 \\
P_3 &= (1 - K_3)^2 P_2 + K_3^2 R_3 \\
&= 10/3 \\
&\approx 3.333
\end{aligned}$$

So we should choose option (b) because it gives a more accurate position estimate than option (a).

3.10 Consider the differential equation

$$\dot{x} + 3x = u$$

If the input $u(t)$ is an impulse, there are two solutions $x(t)$ that satisfy the differential equation. One solution is causal and stable, the other solution is anticausal and unstable. Find the two solutions.

Solution:

The causal and stable solution is

$$x(t) = e^{-3t}\mathcal{U}(t)$$

where $\mathcal{U}(t)$ is the unit step function. The anticausal and unstable solution is

$$x(t) = -e^{-3t}\mathcal{U}(-t)$$

3.11 Suppose a signal $x(t)$ with power spectral density

$$S_x(s) = \frac{1 - s^2}{s^4 - 5s^2 + 4}$$

is corrupted with additive white noise $v(t)$ with a power spectral density $S_v(s) = 1$.

a) Find the optimal noncausal Wiener filter to extract the signal from the noise corrupted signal.

b) Find the optimal causal Wiener filter to extract the signal from the noise corrupted signal.

Solution:

a). The optimal noncausal Wiener filter is given as

$$\begin{aligned} G(s) &= \frac{S_x(s)}{S_{xv}(s)} \\ &= \frac{S_x(s)}{S_x(s)+S_v(s)} \\ &= \frac{s^2-5}{s^2-4} \end{aligned}$$

b). The optimal causal Wiener filter is obtained as follows.

$$\begin{aligned} S_{xv}(s) &= \underbrace{\frac{s+\sqrt{5}}{s+2}}_{S_{xv}^{+}(s)} \underbrace{\frac{s-\sqrt{5}}{s-2}}_{S_{xv}^{-}(s)} \\ \frac{S_x(s)}{S_{xv}^{-}(s)} &= \frac{-1}{(s+2)(s-2)} \frac{s-2}{s-\sqrt{5}} \\ &= \frac{-1}{(s+2)(s-\sqrt{5})} \\ &= \underbrace{\frac{1}{(2+\sqrt{5})(s+2)}}_{\text{causal part}} - \frac{1}{(2+\sqrt{5})(s-\sqrt{5})} \\ G(s) &= \frac{1}{S_{xv}^{+}(s)} \left(\text{causal part of } \frac{S_x(s)}{S_{xv}^{-}(s)} \right) \\ &= \left(\frac{s+2}{s+\sqrt{5}} \right) \frac{1}{(2+\sqrt{5})(s+2)} \\ &= \frac{1}{(2+\sqrt{5})(s+\sqrt{5})} \end{aligned}$$

3.12 A system has the transfer function

$$G(s) = \frac{1}{s-3}$$

If the input is an impulse, there are two solutions for the output $x(t)$ that satisfy the transfer function. One solution is causal and unstable, the other solution is anticausal and stable. Find the two solutions.

Solution:

The causal and unstable solution is

$$x(t) = e^{3t}\mathcal{U}(t)$$

The anticausal and stable solution is

$$x(t) = -e^{3t}\mathcal{U}(-t)$$

Computer exercises

3.13 The production of steel in the United States between 1946 and 1956 was 66.6, 84.9, 88.6, 78.0, 96.8, 105.2, 93.2, 111.6, 88.3, 117.0, and 115.2 million tons [Sor80]. Find the least squares fit to these data using (a) linear curve fit; (b) quadratic curve fit; (c) cubic curve fit; (d) quartic curve fit. For each case give the following: (1) a plot of the original data along with the least squares curve; (2) the RMS error of the least squares curve; (3) the prediction of steel production in 1957.

Solution:

Figure 3.2 shows the four regression curves. The RMS errors of the regression curves are 8.782 (linear curve fit), 8.667 (quadratic curve fit), 8.289 (cubic curve fit), and 8.282 (quartic curve fit). The predicted steel production in 1957 is 118.7 (linear curve fit), 114.5 (quadratic curve fit), 126.2 (cubic curve fit), and 128.9 (quartic curve fit).

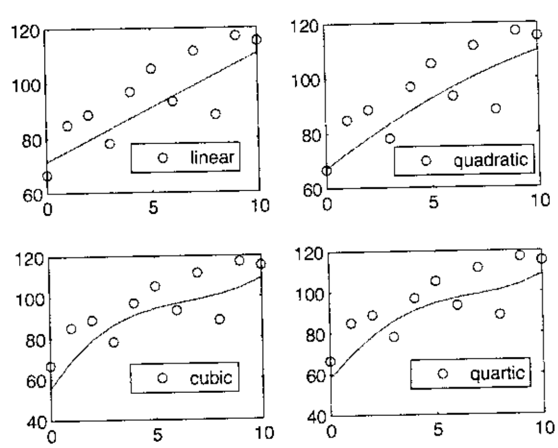

Figure 3.2 Solution to Problem 3.13

3.14 Implement the Wiener filters for the three examples given in Section 3.4 and verify the results shown in Section 3.4.5. Hint: Example 8.6 shows that if $\dot{x} = -x + w$ where $w(t)$ is white noise with a variance of $Q_c = 2$, then

$$S_x(\omega) = \frac{2}{\omega^2 + 1}$$

From Sections 1.4 and 8.1 we see that this system can be simulated as

$$\begin{aligned} x(t + \Delta t) &= e^{-\Delta t} x(t) + w(t)\sqrt{Q_c \Delta t} \\ y(t) &= x(t) + v(t)\sqrt{R_c / \Delta t} \end{aligned}$$

where $w(t)$ and $v(t)$ are independent zero-mean, unity variance random variables.

Solution:

I ran 20 simulations with a simulation time of 100 s and a step size of 0.1 s. The average $E(e^2)$ that I obtained was 0.95 for the parametric Wiener filter, 0.82 for the causal Wiener filter, and 0.78 for the noncausal Wiener filter.

CHAPTER 4

Propagation of states and covariances

Problems

Written exercises

4.1 Prove that

$$\frac{d}{dt}\left(E[x]\right) = E\left[\frac{dx}{dt}\right]$$

Solution:

The derivative and expectation operators are both linear, therefore they can be interchanged.

4.2 Suppose that a dynamic scalar system is given as $x_{k+1} = fx_k + w_k$, where w_k is zero-mean white noise with variance q. Show that if the variance of x_k is σ^2 for all k, then it must be true that $f^2 = (\sigma^2 - q)/\sigma^2$.

Solution:

From the system equation we see that

$$\sigma_{k+1}^2 = f^2\sigma_k^2 + q$$

If $\sigma_{k+1}^2 = \sigma_k^2 = \sigma^2$ for all k, then

$$\begin{aligned} \sigma^2 &= f^2\sigma^2 + q \\ f^2 &= (\sigma^2 - q)/\sigma^2 \end{aligned}$$

QED

4.3 Consider the system

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & 1 \\ 0 & 1/2 \end{bmatrix} x_{k-1} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_{k-1} \\ w_k &\sim (0,1) \end{aligned}$$

where w_k is white noise.

a) Find all possible steady-state values of the mean of x_k.

b) Find all possible steady-state values of the covariance of x_k.

Solution:

a). The mean of x_k satisfies the equation

$$\bar{x}_k = F\bar{x}_{k-1}$$

The steady-state version of this equation is

$$\begin{aligned} (I - F)\bar{x} &= 0 \\ \begin{bmatrix} 0 & -1 \\ 0 & 1/2 \end{bmatrix} \bar{x} &= 0 \\ \bar{x} &= \begin{bmatrix} \text{anything} \\ 0 \end{bmatrix} \end{aligned}$$

Actually $\bar{x}_k(1)$ will be equal to its initial value $\bar{x}_0(1)$.

b). The steady-state solution to Equation (4.4) can be written as

$$\begin{aligned} P &= FPF^T + Q \\ \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1/2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Equating individual elements on the left and right side of the above equation gives

$$\begin{aligned} P_{11} &= P_{11} + 2P_{12} + P_{22} \\ P_{12} &= (P_{12} + P_{22})/2 \\ P_{22} &= P_{22}/4 + 1 \end{aligned}$$

Solving these equations gives

$$\begin{aligned} P_{12} &= 4/3 \\ P_{22} &= 4/3 \end{aligned}$$

P_{11} does not have a steady-state solution.

4.4 Consider the system of Example 1.2.

a) Discretize the system to find the single step state transition matrix F_k, the discrete-time input matrix G_k, and the multiple-step state transition matrix $F_{k,i}$.

b) Suppose the covariance of the initial state is $P_0 = \text{diag}(1, 0)$, and zero-mean discrete-time white noise with a covariance of $Q = \text{diag}(1, 0)$ is input to the discrete-time system. Find a closed-form solution for P_k.

Solution:

a). Denote the discretization step size as T. From Section 1.4 we see that

$$\begin{aligned} F_k &= e^{AT} \\ &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \\ G_k &= F_k \int_0^T e^{-A\tau}\, d\tau B \\ &= \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \end{aligned}$$

$F_{k,i}$ is equal to zero for $k < i$, and it is equal to the identity matrix for $k = i$. For $k > i$ it is equal to

$$\begin{aligned} F_{k,i} &= F_{k-1}F_{k-2}\cdots F_i \\ &= F_k^{k-i} \\ &= \begin{bmatrix} 1 & (k-i)T \\ 0 & 1 \end{bmatrix} \end{aligned}$$

b).

$$\begin{aligned} P_k &= F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \\ \begin{bmatrix} P_{11,k} & P_{12,k} \\ P_{12,k} & P_{22,k} \end{bmatrix} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{11,k-1} & P_{12,k-1} \\ P_{12,k-1} & P_{22,k-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ T & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned}$$

Multiplying this out gives

$$\begin{aligned} P_{11,k} &= P_{11,k-1} + 2TP_{12,k-1} + T^2P_{22,k-1} + 1 \\ P_{12,k} &= P_{12,k-1} + TP_{22,k-1} \\ P_{22,k} &= P_{22,k-1} \end{aligned}$$

Since $P_{22,0} = 0$ we see that $P_{22,k} = 0$ for all k. Combine this with the fact that $P_{12,0} = 0$ to see that $P_{12,k} = 0$ for all k. We can use these facts to see that $P_{11,k} = k + 1$. Therefore

$$P_k = \begin{bmatrix} k+1 & 0 \\ 0 & 0 \end{bmatrix}$$

4.5 Two chemical mixtures are poured into a tank. One has concentration c_1 and is poured at rate F_1, and the other has concentration c_2 and is poured at rate F_2. The tank has volume V, and its outflow is at concentration c and rate F. This is typical of many process control systems [Kwa72]. The linearized equation for this system can be written as

$$\dot{x} = \begin{bmatrix} -\frac{F_0}{2V_0} & 0 \\ 0 & -\frac{F_0}{V_0} \end{bmatrix} x + \begin{bmatrix} 1 & 1 \\ \frac{c_1 - c_0}{V_0} & \frac{c_2 - c_0}{V_0} \end{bmatrix} w$$

where F_0, V_0, and c_0 are the linearization points of F, V, and c. The state x consists of deviations from the steady-state values of V and c, and the noise input w consists of the deviations from the steady-state values of F_1 and F_2. Suppose that $F_0 = 2V_0$, $c_1 - c_0 = V_0$, and $c_2 - c_0 = 2V_0$. Suppose the noise input w has an identity covariance matrix.

a) Use Equation (4.27) to calculate Q_{k-1}.
b) Use Equation (4.28) to approximate Q_{k-1}.
c) Evaluate your answer to part (a) for small $(t_k - t_{k-1})$ to verify that it matches your answer to part (b).

Solution:

a). First compute Q_c as follows.

$$\begin{aligned} Q_c &= BE(ww^T)B^T \\ &= \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix} \end{aligned}$$

Now compute Q_{k-1} as follows.

$$\begin{aligned} Q_{k-1} &= \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)} Q_c e^{A^T(t_k-\tau)}\, d\tau \\ &= e^{At_k} \int_{t_{k-1}}^{t_k} e^{-A\tau} Q_c e^{-A^T\tau}\, d\tau\, e^{A^T t_k} \\ &= e^{At_k} \begin{bmatrix} e^{2t_k} - e^{2t_{k-1}} & e^{3t_k} - e^{3t_{k-1}} \\ e^{3t_k} - e^{3t_{k-1}} & \frac{5}{4}(e^{4t_k} - e^{4t_{k-1}}) \end{bmatrix} e^{A^T t_k} \\ &= \begin{bmatrix} 1 - e^{-2T} & 1 - e^{-3T} \\ 1 - e^{-3T} & \frac{5}{4}(1 - e^{4T}) \end{bmatrix} \end{aligned}$$

where $T = t_k - t_{k-1}$ is the discretization step size.

b).

$$\begin{aligned} Q_{k-1} &\approx Q_c T \\ &= \begin{bmatrix} 2T & 3T \\ 3T & 5T \end{bmatrix} \end{aligned}$$

c). The answer to part (a) can be written as

$$Q_{k-1} = \begin{bmatrix} 1 - (1 - 2T + \frac{(2T)^2}{2!} + \cdots) & 1 - (1 - 3T + \frac{(3T)^2}{2!} + \cdots) \\ 1 - (1 - 3T + \frac{(3T)^2}{2!} + \cdots) & \frac{5}{4}(1 - (1 - 4T + \frac{(4T)^2}{2!} + \cdots)) \end{bmatrix}$$

For small T the higher order powers of T can be neglected, which gives an approximate value for Q_{k-1} which is the same as part (b).

4.6 Suppose that a certain sampled data system has the following state-transition matrix and approximate Q_{k-1} matrix [as calculated by Equation (4.28)]:

$$\begin{aligned} F_{k-1} &= \begin{bmatrix} e^{-T} & 0 \\ 0 & e^{-2T} \end{bmatrix} \\ Q_{k-1} &= \begin{bmatrix} 2T & 3T \\ 3T & 5T \end{bmatrix} \end{aligned}$$

where $T = t_k - t_{k-1}$ is the discretization step size. Use Equation (4.26) to compute the steady-state covariance of the state as a function of T.

Solution:

In steady state Equation (4.26) can be written as

$$P = F_{k-1} P F_{k-1}^T + Q_{k-1}$$

Multiplying out and equating individual elements of the matrices on the right and left side of this equation gives

$$\begin{aligned} P_{11} &= e^{-2T} P_{11} + 2T \\ P_{12} &= e^{-3T} P_{12} + 3T \\ P_{22} &= e^{-4T} P_{22} + 5T \end{aligned}$$

Solving these equations gives

$$\begin{aligned} P_{11} &= \frac{2T}{1 - e^{-2T}} \\ P_{12} &= \frac{3T}{1 - e^{-3T}} \\ P_{22} &= \frac{5T}{1 - e^{-4T}} \end{aligned}$$

4.7 Consider the tank system described in Problem 4.5. Find closed-form solutions for the elements of the state covariance as functions of time.

Solution:

First compute Q_c as follows.

$$\begin{aligned} Q_c &= B E(w w^T) B^T \\ &= \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix} \end{aligned}$$

From Equation (4.49) we obtain

$$\begin{bmatrix} \dot{P}_{11} & \dot{P}_{12} \\ \dot{P}_{12} & \dot{P}_{22} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} + \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} + \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$$

Multiplying out and equating the matrix elements on the right and left side of this equation gives

$$\begin{aligned}
\dot{P}_{11} &= -2P_{11} + 2 \\
\dot{P}_{12} &= -3P_{12} + 3 \\
\dot{P}_{22} &= -4P_{22} + 5
\end{aligned}$$

Solving these equations gives

$$\begin{aligned}
P_{11} &= 1 - e^{-2t} \\
P_{12} &= 1 - e^{-3t} \\
P_{22} &= \frac{5}{4}(1 - e^{-4t})
\end{aligned}$$

4.8 Consider the system

$$\begin{aligned}
x_{k+1} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} x_k + w_k \\
w_k &\sim (0, Q) \\
Q &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}
\end{aligned}$$

Use Equation (4.5) to find the steady-state covariance of the state vector.

Solution:

$$\begin{aligned}
P &= \sum_{i=0}^{\infty} F^i Q (F^T)^i \\
&= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1/4 & 1/4 \\ 0 & 1/4 \end{bmatrix} + \begin{bmatrix} 1/16 & 1/16 \\ 0 & 1/16 \end{bmatrix} + \cdots \\
&= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \sum_0^{\infty} \left(\frac{1}{4}\right)^i \\
&= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \frac{4}{3} \\
&= \begin{bmatrix} 4/3 & 4/3 \\ 0 & 4/3 \end{bmatrix}
\end{aligned}$$

4.9 The third condition of Theorem 21 gives a sufficient condition for the discrete-time Lyapunov equation to have a unique, symmetric, positive semidefinite solution. Since the condition is sufficient but not necessary, there may be cases that do not meet the criteria of the third condition that still have a unique, symmetric, positive semidefinite solution. Give an example of one such case with a nonzero solution.

Solution:

There are many such examples. One is

$$F = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$
$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

This situation does not meet the third condition of Theorem 21 because F is not stable. However, substituting these matrices in the discrete-time Lyapunov equation gives

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Expanding this equation and solving gives the unique solution

$$P = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

which is symmetric and positive semidefinite.

4.10 Prove the sifting property of the continuous-time impulse function $\delta(t)$, which can be stated as

$$\int_{-\infty}^{\infty} f(t)\delta(t-\alpha)\, dt = f(\alpha)$$

Solution:

Note that $\delta(t-\alpha) = 0$ everywhere except at $t = \alpha$. In addition, we know that $\delta(t-\alpha)$ has an area of one. Therefore

$$\begin{aligned} \int_{-\infty}^{\infty} f(t)\delta(t-\alpha)\, dt &= \int_{-\infty}^{\infty} f(\alpha)\delta(t-\alpha)\, dt \\ &= f(\alpha)\int_{-\infty}^{\infty} \delta(t-\alpha)\, dt \\ &= f(\alpha) \end{aligned}$$

QED

Computer exercises

4.11 Write code for the propagation of the mean and variance of the state of Example 4.2. Use $m_0 = 1$, $P_0 = 2$, $f = -0.5$ and $q_c = 1$. Plot the mean and variance of x for 5 seconds. Repeat for $P_0 = 0$. Based on the plots, what does the steady-state value of the variance appear to be? What is the analytically determined steady-state value of the variance?

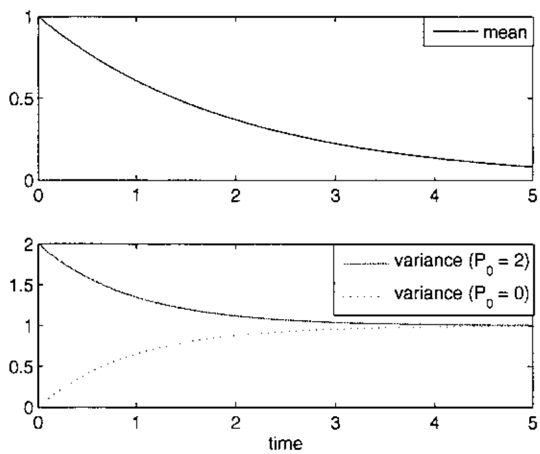

Figure 4.1 Solution to Problem 4.11

Solution:

Figure 4.1 shows the solution. The steady-state value of the variance appears to be 1. Analytically, the steady-state value of the variance is $-q_c/2f = 1$.

4.12 Consider the RLC circuit of Example 1.8 with $R = L = C = 1$. Suppose the applied voltage is continuous-time zero-mean white noise with a variance of 1. The initial capacitor voltage is a random variable with a mean of 1 and a variance of 1. The initial inductor current is a random variable (independent of the initial capacitor voltage) with a mean of 2 and a variance of 2. Write a program to propagate the mean and covariance of the state for five seconds. Plot the two elements of the mean of the state, and the three unique elements of the covariance. Based on the plots, what does the steady-state value of the covariance appear to be? What is the analytically determined steady-state value of the covariance? (Hint: The MATLAB function LYAP can be used to solve for the continuous-time algebraic Lyapunov equation.)

Solution:

To solve this we first must see that $Q_c = Bq_cB^T$, where $q_c = 1$ is the variance of the applied voltage. Figure 4.2 shows the solution. The steady-state value of the variance appears to be $\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$. Analytically, the steady-state value of the variance is the solution of the Lyapunov equation $AP + PA^T + Q_c = 0$, which is $\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$.

4.13 Consider the RLC circuit of Problem 1.18 with $R = 3$, $L = 1$, and $C = 0.5$. Suppose the applied voltage is continuous-time zero-mean white noise with a variance of 1. We can find the steady-state covariance of the state a couple of different ways.

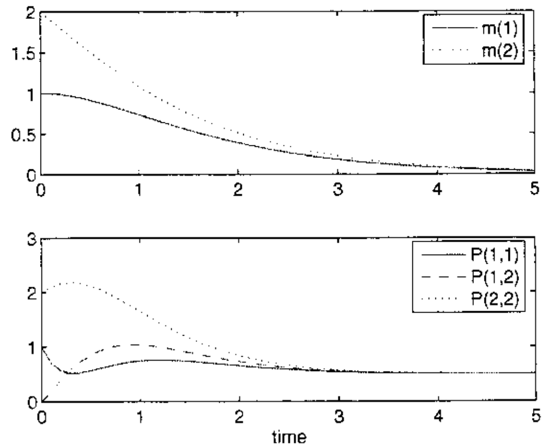

Figure 4.2 Solution to Problem 4.12

- Use Equation (4.49).
- Discretize the system and use Equation (4.4) along with the MATLAB function DLYAP. In this case, the discrete-time white noise covariance Q is related to the continuous-time white noise covariance Q_c by the equation $Q = TQ_c$, where T is the discretization step size (see Section 8.1.1).

a) Analytically compute the continuous-time, steady-state covariance of the state.

b) Analytically compute the discretized steady-state covariance of the state in the limit as $T \to \infty$.

c) One way of measuring the distance between two matrices is by using the MATLAB function NORM to take the Frobenius norm of the difference between the matrices. Generate a plot showing the Frobenius norm of the difference between the continuous-time, steady-state covariance of the state, and the discretized steady-state covariance of the state for T between 0.01 and 1.

Solution:

a). First we note that $Q_c = Bq_cB^T$, where $q_c = 1$ is the variance of the input, and B is found in Problem 1.18 as $B = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. The continuous-time steady-state Lyapunov equation is $AP + PA^T + Q_c = 0$. Analytically solving this equation gives

$$P = \begin{bmatrix} 1/3 & 0 \\ 0 & 1/6 \end{bmatrix}$$

b). The discrete-time system matrix F and the discrete-time noise covariance are found as

$$F = e^{AT}$$

$$
\begin{aligned}
&= e^{-T}\begin{bmatrix} 2 & 2 \\ -1 & -1 \end{bmatrix} + e^{-2T}\begin{bmatrix} -1 & -2 \\ 1 & 2 \end{bmatrix} \\
\lim_{T\to\infty} F &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
Q &= Q_c T \\
&= \begin{bmatrix} 0 & 0 \\ 0 & T \end{bmatrix}
\end{aligned}
$$

Substituting these values into the discretized steady-state Lyapunov equation $P = FPF^T + Q$ and solving for P gives

$$
\lim_{T\to\infty} P = \begin{bmatrix} 0 & 0 \\ 0 & \infty \end{bmatrix}
$$

c). The solution is shown in Figure 4.3.

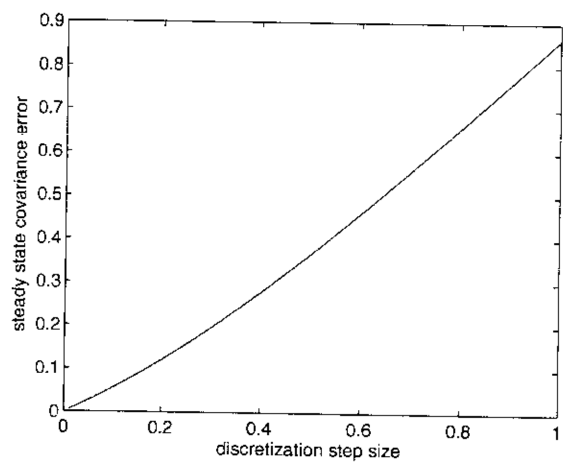

Figure 4.3 Solution to Problem 4.13

CHAPTER 5

The discrete-time Kalman filter

Problems

Written exercises

5.1 A radioactive mass has a half-life of τ seconds. At each time step the number of emitted particles x is half of what it was one time step ago, but there is some error w_k (zero-mean with variance Q) in the number of emitted particles due to background radiation. At each time step, the number of emitted particles is counted. The instrument used to count the number of emitted particles has a random error at time k of v_k, which is zero-mean with a variance of R. Assume that w_k and v_k are uncorrelated.

a) Write the linear system equations for this system.

b) Suppose we want to use a Kalman filter to find the optimal estimate of the number of emitted particles at each time step. Write the one-step *a posteriori* Kalman filter equations for this system.

c) Find the steady-state *a posteriori* estimation-error variance for the Kalman filter.

d) What is the steady-state Kalman gain when $Q = R$? What is the steady-state Kalman gain when $Q = 2R$? Give an intuitive explanation for why the steady-state gain changes the way it does when the ratio of Q to R changes.

Solution

a).

$$\begin{aligned} x_k &= \frac{1}{2}x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \end{aligned}$$

b).

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ &= \frac{1}{4}P_{k-1}^+ + Q \\ K_k &= P_k^- H_k^T(H_kP_k^-H_k^T + R_k)^{-1} \\ &= \frac{P_k^-}{P_k^- + R} \\ \hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ \\ &= \frac{1}{2}\hat{x}_{k-1}^+ \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \\ &= \frac{1}{2}\hat{x}_{k-1}^+ + K_k(y_k - \hat{x}_{k-1}^+/2) \\ P_k^+ &= (I - K_kH_k)P_k^- \\ &= \frac{R(P_{k-1}^+ + 4Q)}{P_{k-1}^+ + 4(Q+R)} \\ K_k &= P_k^+H_k^TR_k^{-1} \\ &= \frac{P_{k-1}^+ + 4Q}{P_{k-1}^+ + 4(Q+R)} \end{aligned}$$

The one-step *a posteriori* equations can be summarized as follows.

$$\begin{aligned} P_k^+ &= \frac{R(P_{k-1}^+ + 4Q)}{P_{k-1}^+ + 4(Q+R)} \\ K_k &= \frac{P_{k-1}^+ + 4Q}{P_{k-1}^+ + 4(Q+R)} \\ \hat{x}_k^+ &= \frac{1}{2}\hat{x}_{k-1}^+ + K_k(y_k - \hat{x}_{k-1}^+/2) \end{aligned}$$

c).

$$P^+ = \frac{R(P^+ + 4Q)}{P^+ + 4(Q+R)}$$

Solving for P^+ gives

$$P^+ = \frac{1}{2}\left[-(4Q+3R) \pm \sqrt{(4Q+3R)^2 + 16QR}\right]$$

We use the plus sign to get a positive solution for P^+.

d). Use the formula $K_k = P_k^+ H_k^T R_k^{-1}$ to get the steady-state gain

$$K = \frac{1}{2R}\left[-(4Q+3R) + \sqrt{(4Q+3R)^2 + 16QR}\right]$$

When $Q = R$ this gives $K \approx 0.53$. When $Q = 2R$ this gives $K \approx 0.68$. We see that as process noise increases relative to measurement noise, the gain increases. This is because the Kalman filter places more emphasis on the measurements when the system model is less certain.

5.2 This problem illustrates the robustness that is achieved by the use of the Joseph form of the covariance measurement update equation. Suppose you have a discrete-time Kalman filter for a scalar system.

a) Find $\partial P_k^+/\partial K_k$ for the third form of the covariance measurement update in Equation (5.19).

b) Find $\partial P_k^+/\partial K_k$ for the Joseph form (the first form) of the covariance measurement update in Equation (5.19). After you get your answer, substitute for K_k from the Kalman gain expression.

c) Use the above results to explain why the Joseph form of the covariance measurement-update equation is stable and robust.

Solution:

a). For a scalar system, the third form of the covariance measurement update is $P^+ = (1 - KH)P^-$. From this we obtain

$$\frac{\partial P^+}{\partial K} = -HP^-$$

b). For a scalar system, the Joseph form of the covariance measurement update is $P^+ = (1 - KH)^2 P^- + RK^2$. From this we obtain

$$\frac{\partial P^+}{\partial K} = -2H(1 - KH)P^- + 2KR$$

Substituting for K from Equation (5.19) gives

$$\begin{aligned}
\frac{\partial P^+}{\partial K} &= -2H\left(1 - \frac{P^-H^2}{P^-H^2 + R}\right)P^- + \frac{2P^-HR}{P^-H^2 + R} \\
&= \frac{-2H^3(P^-)^2 - 2HP^-R + 2H^3(P^-)^2 + 2HP^-R}{P^-H^2 + R} \\
&= 0
\end{aligned}$$

c). The above results show that the Joseph form of the covariance measurement update is insensitive to numerical errors in the Kalman gain.

5.3 Prove that $E[\hat{x}_k^+(\tilde{x}_k^+)^T] = 0$. Hint: Since $\hat{x}_0^+ = E[x_0]$ is a constant and $\tilde{x}_0^+ = x_0 - \hat{x}_0^+$ is zero-mean, we know that $E[\hat{x}_0^+(\tilde{x}_0^+)^T] = 0$. Given this information, prove that $E[\hat{x}_1^+(\tilde{x}_1^+)^T] = 0$. From this point, use induction to complete the proof.

Solution:

This solution is taken from [Gel74]. We will prove that if $E[\hat{x}_k\tilde{x}_k^T] = 0$, then $E[\hat{x}_{k+1}\tilde{x}_{k+1}^T] = 0$. Note that the + superscript has been dropped for ease of notation. We will assume that the input is zero, again for ease of notation.

$$
\begin{aligned}
E[\hat{x}_{k+1}\tilde{x}_{k+1}^T] &= E\Big\{ [F_k\hat{x}_k + K_{k+1}(-H_{k+1}F_k\tilde{x}_k + H_{k+1}w_k + v_{k+1})] \times \\
&\quad \left[\tilde{x}_k^T (F_k - K_{k+1}H_{k+1}F_k)^T + w_k^T(K_{k+1}H_{k+1} - I)^T + v_{k+1}^T K_{k+1}^T\right]\Big\} \\
&= -K_{k+1}H_{k+1}F_kP_k(F_k^T - F_k^T H_{k+1}^T K_{k+1}^T) + \\
&\quad K_{k+1}H_{k+1}Q_k(H_{k+1}^T K_{k+1}^T - I) + K_{k+1}R_{k+1}K_{k+1}^T \\
&= -K_{k+1}H_{k+1}(F_kP_kF_k^T + Q_k) + \\
&\quad K_{k+1}H_{k+1}(F_kP_kF_k^T + Q_k)H_{k+1}^T K_{k+1}^T + K_{k+1}R_{k+1}K_{k+1}^T \\
&= -K_{k+1}H_{k+1}P_{k+1}^-(I - K_{k+1}^T H_{k+1}^T) + K_{k+1}R_{k+1}K_{k+1}^T \\
&= K_{k+1}(-H_{k+1}P_{k+1}^T + R_{k+1}K_{k+1}^T) \\
&= 0
\end{aligned}
$$

Since it is true for $k = 0$, it is also true for $k = 1$; since it is true for $k = 1$, it is also true for $k = 2$; etc. Proof by induction.
QED

5.4 Suppose that you have a fish tank with x_p piranhas and x_g guppies [Bay99]. Once per week, you put guppy food into the tank (which the piranhas do not eat). Each week the piranhas eat some of the guppies. The birth rate of the piranhas is proportional to the guppy population, and the death rate of the piranhas is proportional to their own population (due to overcrowding). Therefore $x_p(k+1) = x_p(k) + k_1x_g(k) - k_2x_p(k) + w_p(k)$, where k_1 and k_2 are proportionality constants and $w_p(k)$ is white noise with a variance of one that accounts for mismodeling. The birth rate of the guppies is proportional to the food supply u, and the death rate of the guppies is proportional to the piranha population. Therefore, $x_g(k+1) = x_g(k) + u(k) - k_3x_p(k) + w_g(k)$, where k_3 is a proportionality constant and $w_g(k)$ is white noise with a variance of one that accounts for mismodeling. The step size for this model is one week. Every week, you count the piranhas and guppies. You can count the piranhas accurately because they are so large, but your guppy count has zero-mean noise with a variance of one. Assume that $k_1 = 1$ and $k_2 = k_3 = 1/2$.

a) Generate a linear state-space model for this system.

b) Suppose that at the initial time you have a perfect count for x_p and x_g. Using a Kalman filter to estimate the guppy population, what is the variance of your guppy population estimate after one week? What is the variance after two weeks?

c) What is the ratio of the piranha population to the guppy population when they reach steady state? Assume that the process noise is zero for this part of the problem.

Solution:

a). Let the first element of the state vector be the piranha population, and the second element of the state vector be the guppy population. The state space equations are given as

$$
\begin{aligned}
x_{k+1} &= \begin{bmatrix} 1-k_2 & k_1 \\ -k_3 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k + \begin{bmatrix} w_{p,k} \\ w_{g,k} \end{bmatrix} \\
&= \begin{bmatrix} 1/2 & 1 \\ -1/2 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k + w_k \\
y_k &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k \\
w_k &\sim (0, Q) \\
v_k &\sim (0, R) \\
Q &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
R &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}
$$

b). Since the initial population count is perfect we have

$$
P_0^+ = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}
$$

From the Kalman filter equations we obtain

$$
\begin{aligned}
P_1^- &= F P_0^+ F^T + Q \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
K_1 &= P_1^- H^T (H P_1^- H^T + R)^{-1} \\
&= \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} \\
P_1^+ &= (I - K_1 H) P_1^- \\
&= \begin{bmatrix} 0 & 0 \\ 0 & 1/2 \end{bmatrix} \\
P_2^- &= F P_1^+ F^T + Q \\
&= \begin{bmatrix} 3/2 & 1/2 \\ 1/2 & 3/2 \end{bmatrix} \\
K_2 &= P_2^- H^T (H P_2^- H^T + R)^{-1} \\
&= \begin{bmatrix} 1 & 0 \\ 1/7 & 4/7 \end{bmatrix} \\
P_2^+ &= (I - K_2 H) P_2^- \\
&= \begin{bmatrix} 0 & 0 \\ 0 & 4/7 \end{bmatrix}
\end{aligned}
$$

We see that the guppy population estimation variance is 1/2 after one week and 4/7 after two weeks.

c). In steady state $x_{k+1} = x_k$ so

$$\begin{aligned} x_k &= Fx_k + Gu_k \\ x_k &= (I - F)^{-1}Gu_k \\ &= \begin{bmatrix} 2 \\ 1 \end{bmatrix} u_k \end{aligned}$$

We see that the ratio of the piranha population to the guppy population in steady state is 2:1.

5.5 The measured output of a simple moving average process is $y_k = z_k + z_{k-1}$, where $\{z_j\}$ is zero-mean white noise with a variance of one.

a) Generate a state-space description for this system with the first element of x_k equal to z_{k-1} and second element equal to z_k.

b) Suppose that the initial estimation-error covariance is equal to the identity matrix. Show that the *a posteriori* estimation-error covariance is given by

$$P_k^+ = \frac{1}{k+1} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

c) Find $E\left[\|x_k - \hat{x}_k^+\|_2^2\right]$ as a function of k.

Solution:

a). The state space description is given as

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} z_{k+1} \\ &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_k + w_k \\ w_k &\sim (0, Q) \\ Q &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ y_k &= \begin{bmatrix} 1 & 1 \end{bmatrix} x_k + v_k \\ v_k &\sim (0, R) \\ R &= 0 \end{aligned}$$

b). Going through the Kalman filter equations we can derive

$$P_1^+ = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Now suppose that

$$P_k^+ = \frac{1}{k+1} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Then P_{k+1}^+ can be derived as follows.

$$P_{k+1}^- = FP_k^+F^T + Q$$

$$
\begin{aligned}
&= \frac{1}{k+1}\begin{bmatrix} 1 & 0 \\ 0 & k+1 \end{bmatrix} \\
K_{k+1} &= P_{k+1}^{-}H^T\left(HP_{k+1}^{-}H^T + R\right)^T \\
&= \frac{1}{k+2}\begin{bmatrix} 1 \\ k+1 \end{bmatrix} \\
P_{k+1}^{+} &= \left(I - K_{k+1}H\right)P_{k+1}^{-} \\
&= \frac{1}{k+2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}
\end{aligned}
$$

Since P_k^+ has the specified form for $k=1$, it also has the specified form for $k=2$. Proof by induction.
QED

c).

$$
\begin{aligned}
E\left[\|x_k - \hat{x}_k^+\|_2^2\right] &= E\left\{\mathrm{Tr}\left[(x_k - \hat{x}_k^+)(x_k - \hat{x}_k^+)^T\right]\right\} \\
&= \mathrm{Tr}\left\{E\left[(x_k - \hat{x}_k^+)(x_k - \hat{x}_k^+)^T\right]\right\} \\
&= \mathrm{Tr}\left(P_k^+\right) \\
&= \frac{2}{k+1}
\end{aligned}
$$

5.6 In this problem, we use the auxiliary variable $S_k = H_kP_k^-H_k^T + R_k$. Note that

$$
\begin{bmatrix} I & 0 \\ -P_k^-H_k^TS_k^{-1} & I \end{bmatrix}\begin{bmatrix} S_k & H_kP_k^- \\ P_k^-H_k^T & P_k^- \end{bmatrix} = \begin{bmatrix} S_k & H_kP_k^- \\ 0 & P_k^+ \end{bmatrix}
$$

Use the product rule for determinants to show that

$$
|P_k^+| = \frac{|P_k^-||R_k|}{|S_k|}
$$

Solution:

Using the given matrix equation and the product rule for determinants of Equation (1.48), we obtain

$$
\begin{aligned}
|P_k^-||S_k - H_kP_k^-H_k^T| &= |S_k||P_k^+| \\
|P_k^-||R_k| &= |S_k||P_k^+| \\
|P_k^+| &= \frac{|P_k^-||R_k|}{|S_k|}
\end{aligned}
$$

QED

5.7 In Section 4.1, we saw that Σ_k, the covariance of the state of a discrete-time system, is given as $\Sigma_{k+1} = F_k\Sigma_kF_k^T + Q_k$. Use this along with the one-step expression for the *a priori* estimation-error covariance of the Kalman filter to show that $\Sigma_k - P_k^- \geq 0$ for all k. Give an intuitive explanation for this expression [And79].

Solution:

We have

$$\begin{aligned}\Sigma_{k+1} &= F_k \Sigma_k F_k^T + Q_k \\ P_{k+1}^- &= F_k P_k^- F_k^T - F_k P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- F_k^T + Q_k\end{aligned}$$

Subtracting the second equation from the first gives

$$\begin{aligned}\Delta_{k+1} &= \Sigma_{k+1} - P_{k+1}^- \\ &= F_k \Delta_k F_k^T + F_k P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- F_k^T\end{aligned}$$

We note that if Δ_k is positive semidefinite and R_k is positive definite, then Δ_{k+1} is positive semidefinite. So since $\Delta_1 = 0$ (positive semidefinite) we see that $\Delta_2 \geq 0$, which in turn means that $\Delta_3 \geq 0$, etc. We conclude that $\Sigma_k - P_k^- \geq 0$ for all k. Intuitively this means that the covariance of the state is larger than the covariance of the state estimate, which means that the measurement always allows us to reduce our uncertainty in the state.

5.8 Consider the system of Problem 5.1.

a) Use the method of Section 5.4 to find a closed-form solution for P_k^-, assuming that $Q = 1$, $R = 5$, and $P_0 = 0$.

b) Use your result from above to find the steady-state value of P_k^-.

Solution:

a). From Equation (5.50) we obtain

$$\Psi = \begin{bmatrix} 9/10 & 2 \\ 2/5 & 2 \end{bmatrix}$$

The eigendata of Ψ are found to be

$$\begin{aligned}\Lambda &= \begin{bmatrix} 5/2 & 0 \\ 0 & 2/5 \end{bmatrix} \\ M &= \begin{bmatrix} 5/4 & -4 \\ 1 & 1 \end{bmatrix} \\ M^{-1} &= \frac{1}{21} \begin{bmatrix} 4 & 16 \\ -4 & 5 \end{bmatrix}\end{aligned}$$

From this we obtain

$$\begin{aligned}\Psi^k &= M \Lambda^k M^{-1} \\ &= \frac{1}{21} \left[\left(\frac{5}{2}\right)^k \begin{pmatrix} 5 & 20 \\ 4 & 16 \end{pmatrix} + \left(\frac{2}{5}\right)^k \begin{pmatrix} 16 & -20 \\ -4 & 5 \end{pmatrix} \right]\end{aligned}$$

This gives A_k and B_k as

$$\begin{bmatrix} A_k \\ B_k \end{bmatrix} = \Psi^k \begin{bmatrix} P_0 \\ 1 \end{bmatrix}$$

$$
\begin{aligned}
&= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
A_k &= \frac{1}{21}\left[20\left(\frac{5}{2}\right)^k - 20\left(\frac{2}{5}\right)^k\right] \\
B_k &= \frac{1}{21}\left[16\left(\frac{5}{2}\right)^k + 5\left(\frac{2}{5}\right)^k\right]
\end{aligned}
$$

From this we obtain

$$
\begin{aligned}
P_k^- &= A_k B_k^{-1} \\
&= \frac{20\left(\frac{5}{2}\right)^k - 20\left(\frac{2}{5}\right)^k}{16\left(\frac{5}{2}\right)^k + 5\left(\frac{2}{5}\right)^k}
\end{aligned}
$$

b). From the above we result we obtain

$$
\begin{aligned}
\lim_{k\to\infty} P_k^- &= \frac{20}{16} \\
&= \frac{5}{4}
\end{aligned}
$$

5.9 Suppose that a Kalman filter is designed for the system

$$
\begin{aligned}
x_{k+1} &= x_k \\
y_k &= x_k + v_k \\
v_k &\sim (0, R)
\end{aligned}
$$

a) Suppose that $E(x_0^2) = 1$. Design a Kalman filter for the system and find a closed-form expression for P_k^-. What is the limit of P_k^- as $k \to \infty$?

b) Now suppose that the true process equation is actually $x_{k+1} = x_k + w_k$, where $w_k \sim (0, Q)$. Find a difference equation for the variance of the *a priori* estimation error if the Kalman filter that you designed in part (a) is used to estimate the state. What is the limit of the estimation-error variance as $k \to \infty$?

Solution

a).

$$
\begin{aligned}
P_{k+1}^- &= P_k^- - K_k P_k^- \\
&= P_k^- - \frac{(P_k^-)^2}{P_k^- + R} \\
&= \frac{P_k^-}{P_k^- + R}
\end{aligned}
$$

Since $P_1^- = P_0^+ = 1$, we see that

$$
P_k^- = \left(\sum_{i=1}^{k-1} R_i\right)^{-1}
$$

We see that

$$\lim_{k \to \infty} P_k^- = 0$$

b). The *a priori* state estimate can be written as

$$\begin{aligned} \hat{x}_{k+1}^- &= \hat{x}_k^- + K_k(y_k - \hat{x}_k^-) \\ &= \hat{x}_k^- + \frac{1}{k+1}(x_k + v_k - \hat{x}_k^-) \\ &= \frac{k}{k+1}\hat{x}_k^- + \frac{1}{k+1}x_k + \frac{1}{k+1}v_k \end{aligned}$$

The state at time $(k+1)$ is given as

$$x_{k+1} = x_k + w_k$$

Subtracting $\hat{x}_{k+1}^-$ from x_{k+1} gives

$$\begin{aligned} \epsilon_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\ &= (x_k - w_k) - \left(\frac{k}{k+1}\hat{x}_k^- + \frac{1}{k+1}x_k + \frac{1}{k+1}v_k \right) \\ &= \frac{k}{k+1}\epsilon_k + w_k - \frac{1}{k+1}v_k \end{aligned}$$

Taking the variance of both sides gives

$$E(\epsilon_{k+1}^2) = \left(\frac{k}{k+1} \right)^2 E(\epsilon_k^2) + Q + \frac{R}{k+1}$$

Because of the Q on the right side of the above equation, we see that

$$\lim_{k \to \infty} E(\epsilon_k^2) = \infty$$

5.10 Suppose that a Kalman filter is designed for a discrete LTI system with an assumed measurement noise covariance of R, but the actual measurement noise covariance is $(R + \Delta R)$. The output of the Kalman filter will indicate that the *a priori* estimation-error covariance is P_k^-, but the actual *a priori* estimation-error covariance will be Σ_k^-. Find a difference equation for $\Delta_k = (\Sigma_k^- - P_k^-)$. Will Δ_k always be positive definite?

Solution:

From Equation (5.24) we obtain

$$\hat{x}_{k+1}^- = F(I - K_k H)\hat{x}_k^- + F K_k y_k$$

where we have assumed $u_k = 0$ for ease of notation. Given the system equations

$$\begin{aligned} x_{k+1} &= F x_k + w_k \\ y_k &= H x_k + v_k \end{aligned}$$

where $w_k \sim (0, Q)$ and $v_k \sim (0, R)$, we can obtain

$$\begin{aligned}
\epsilon_{k+1} &= x_{k+1} - \hat{x}_{k+1}^- \\
&= F(I - K_k H)\epsilon_k + w_k - F K_k v_k \\
P_{k+1}^- &= E[\epsilon_{k+1}\epsilon_{k+1}^T] \\
&= F(I - K_k H)P_k^-(I - K_k H)^T F^T + Q + F K_k R K_k^T F^T
\end{aligned}$$

This is the estimation-error covariance if the true measurement noise covariance is equal to R as assumed. If the measurement noise covariance is instead equal to $R + \Delta R$, then the estimation-error covariance is equal to Σ_k^- and is instead governed by the equation

$$\Sigma_{k+1}^- = F(I - K_k H)\Sigma_k^-(I - K_k H)^T F^T + Q + F K_k (R + \Delta R) K_k^T F^T$$

Subtracting the P_{k-1}^- equation from the Σ_{k+1}^- equation gives

$$\begin{aligned}
\Delta_{k+1} &= \Sigma_{k+1}^- - P_{k+1}^- \\
&= F(I - K_k H)\Delta_k(I - K_k H)^T F^T + F K_k \Delta R K_k^T F^T
\end{aligned}$$

If ΔR is not positive definite, then Δ_k might also not be positive definite.

Computer exercises

5.11 Let p_k denote the wombat population at time k, and f_k denote the size of the wombat's food supply at time k. From one time step to the next, half of the existing wombat population dies, but the number of new wombats is added to the population is equal to twice the food supply. The food supply is constant except for zero-mean random fluctuations with a variance of 10. At each time step the wombat population is counted with an error that has zero mean and a variance of 10. The initial state is

$$\begin{aligned}
p_0 &= 650 \\
f_0 &= 250
\end{aligned}$$

The initial state estimate and uncertainty is

$$\begin{aligned}
\hat{p}_0 &= 600 \\
E[(\hat{p}_0 - p_0)^2] &= 500 \\
\hat{f}_0 &= 200 \\
E[(\hat{f}_0 - f_0)^2] &= 200
\end{aligned}$$

Design a Kalman filter to estimate the population and food supply.

a) Simulate the system and the Kalman filter for 10 time steps. Hand in the following.

- Source code listing.
- A plot showing the true population and the estimated population as a function of time.

- A plot showing the true food supply and the estimated food supply as a function of time.
- A plot showing the standard deviation of the population and food supply estimation error as a function of time.
- A plot showing the elements of the Kalman gain matrix as a function of time.

b) Compare the standard deviation of the estimation error of your simulation with the steady-state theoretical standard deviation based on P_k^+. Why is there such a discrepancy?

c) Run the simulation again for 1000 time steps and compare the experimental estimation error standard deviation with the theoretical standard deviation.

Solution

a). The plots are shown in Figure 5.1.

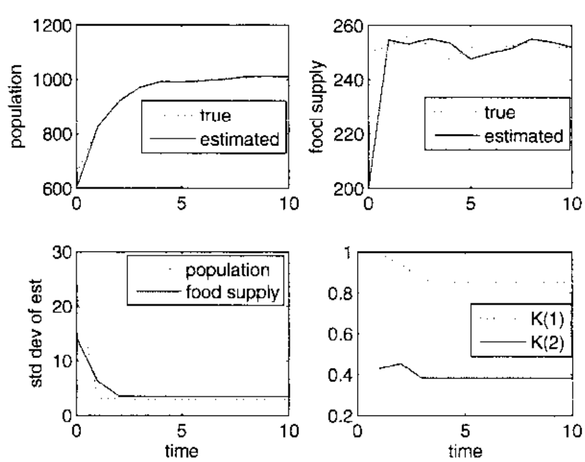

Figure 5.1 Solution to Problem 5.11

b). Depending on the simulated noise history, the standard deviations of the population and food supply estimation errors from the simulation are both around 15. The steady-state theoretical standard deviations that come out of the Kalman filter equations are 2.9 for population and 3.5 for food supply. The reason for the discrepancy is the large initial estimation error and the short simulation time.

c). When we run for 1000 time steps the standard deviations of the population and food supply estimation errors from the simulation are about 3.3 and 3.9 respectively. This is much closer to the theoretical standard deviation because the simulation time is stretched out long enough that the initial estimation errors do not swamp the errors at the later times.

5.12 Consider the RLC circuit described in Problem 1.18 with $R = 3$, $L = 1$, and $C = 0.5$. The input voltage is zero-mean, unity variance white noise. Suppose that the capacitor voltage is measured at 10 Hz with zero-mean, unity variance white noise. Design a Kalman filter to estimate the inductor current, with an initial covariance $P_0^+ = 0$. Generate a plot showing the *a priori* and *a posteriori* variances of the inductor current estimate for 20 time steps. Based on the plot, what is the steady-state value of P_k^-? Use the development of Section 5.4.1 to approximate the steady-state value of P_k^- using 1, 2, 3, and 4 successive squares of the Ψ matrix.

Solution:

Figure 5.2 shows the *a priori* variances of the inductor current estimate for 20 time steps. Based on the plot it appears that P_k^- converges to 1.9592. If we use Section 5.4.1 to approximate the steady-state value of P_k^-, we obtain the following:

$$\begin{aligned} p = 1 &: \quad P_\infty^- = 1.5367 \\ p = 2 &: \quad P_\infty^- = 1.8995 \\ p = 3 &: \quad P_\infty^- = 1.9336 \\ p = 4 &: \quad P_\infty^- = 1.9592 \end{aligned}$$

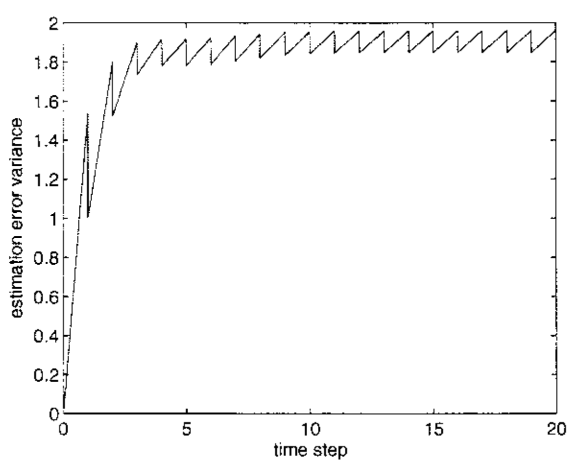

Figure 5.2 Solution to Problem 5.12

CHAPTER 6

Alternate Kalman filter formulations

Problems

Written exercises

6.1 In this chapter, we discussed alternatives to the standard Kalman filter formulation. Some of these alternatives include the sequential Kalman filter, the information filter, and the square root filter.

a) What is the advantage of the sequential Kalman filter over the batch Kalman filter? What is the advantage of the batch Kalman filter over the sequential Kalman filter?

b) What is the advantage of the information filter over the standard Kalman filter? What is the advantage of the standard Kalman filter over the information filter?

c) What is the advantage of the square root filter over the standard Kalman filter? What is an advantage of the standard Kalman filter over the square root Kalman filter?

Solution:

a). An advantage of the sequential filter is that it does not require any matrix inversions. An advantage of the batch filter is that it can handle time-varying nondiagonal measurement noise covariances.

b). An advantage of the information filter is that it requires less computational effort if the noise covariances are constant and the number of measurements is much greater than the number of states. Also, we can represent zero information exactly with the information filter, but only approximately with the standard Kalman filter. An advantage of the standard Kalman filter is that it requires less computational effort if the number of states is greater than the number of measurements. Also, we can represent zero uncertainty exactly with the standard Kalman filter, but only approximately with the information filter.

c). An advantage of the square root filter is that it results in twice as much numerical precision. An advantage of the standard Kalman filter is that it requires less computational effort.

6.2 Suppose that you have a system with the following measurement and measurement noise covariance matrices:

$$\begin{aligned} H &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ R &= \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \end{aligned}$$

You want to use a sequential Kalman filter to estimate the state of the system. Derive the normalized measurement, measurement matrix, and measurement noise covariance matrix that could be used in a sequential Kalman filter.

Solution:

First we compute the eigendata of R. The eigenvalue matrix $\hat{R}$ and eigenvector matrix S can be computed as

$$\begin{aligned} \hat{R} &= \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \\ S &= \frac{1}{2}\begin{bmatrix} -\sqrt{2} & \sqrt{2} \\ \sqrt{2} & \sqrt{2} \end{bmatrix} \end{aligned}$$

The normalized covariance matrix is $\hat{R}$, and the normalized measurement matrix is

$$\begin{aligned} \tilde{H} &= S^{-1}H \\ &= S \end{aligned}$$

The normalized measurement is

$$\begin{aligned} \tilde{y} &= S^{-1}y \\ &= Sy \end{aligned}$$

6.3 Consider the two alternative forms for the information matrix time-update equation. What advantages does Equation (6.28) have? What advantages does Equation (6.30) have?

Solution:

Equation (6.28) guarantees that $\mathcal{I}_k^-$ will be symmetric positive definite, assuming that $\mathcal{I}_{k-1}^+$ is symmetric positive definite. However, Equation (6.28) requires two $n \times n$ matrix inversions. If Q_k is constant, Equation (6.30) requires only one $n \times n$ matrix inversion.

6.4 A radioactive mass has a half-life of τ seconds. At each time step k the number of emitted particles x is half of what it was one time step ago, but there is some error w_k (zero-mean with variance Q_k) in the number of emitted particles due to background radiation. At each time step the number of emitted particles is counted with two separate and independent instruments. The instruments used to count the number of emitted particles both have a random error at each time step that is zero-mean with a unity variance. The initial uncertainty in the number of radioactive particles is a random variable with zero mean and unity variance.

a) The discrete-time equations that model this system have a one-dimensional state and a two-dimensional measurement. Use the information filter to compute the *a priori* and *a posteriori* information matrix at $k = 1$ and $k = 2$. Assume that $Q_0 = 1$ and $Q_1 = 5/4$.

b) Another way to solve this problem is to realize that the two measurements can be averaged to form a single measurement with a smaller variance than the two independent measurements. What is the variance of the averaged measurement at each time step? Use the standard Kalman filter equations to compute the *a priori* and *a posteriori* covariance matrix at $k = 1$ and $k = 2$, and verify that it is the inverse of the information matrix that you computed in part (a).

Solution:

The system model is given as

$$
\begin{aligned}
x_k &= \frac{1}{2}x_{k-1} + w_{k-1} \\
y_k &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_k + v_k
\end{aligned}
$$

where $w_k \sim (0, Q_k)$ and $v_k \sim (0, I)$.

a). The information matrix equations are given in Equation (6.33), from which we obtain

$$
\begin{aligned}
\mathcal{I}_1^- &= 1 - \left(\frac{1}{4}\right)\left(\frac{1}{1 + 1/4}\right) \\
&= \frac{4}{5} \\
\mathcal{I}_1^+ &= \frac{4}{5} + \begin{bmatrix} 1 & 1 \end{bmatrix} I \begin{bmatrix} 1 \\ 1 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{14}{5} \\
\mathcal{I}_2^- &= \frac{4}{5} - \left(\frac{1}{4}\right)\left(\frac{4}{5}\right)\left(\frac{1}{14/5+1/5}\right)\left(\frac{4}{5}\right) \\
&= \frac{56}{75} \\
\mathcal{I}_2^+ &= \frac{56}{75} + 2 \\
&= \frac{206}{75}
\end{aligned}
$$

b). Suppose we have two independent measurements of the scalar x, each with variance R. Then the variance of the averaged measurement is computed as

$$
\begin{aligned}
E\left\{\left[\frac{1}{2}(y_1+y_2)-x\right]^2\right\} &= E\left\{\left[\frac{1}{2}(y_1-x)+\frac{1}{2}(y_2-x)\right]^2\right\} \\
&= E\left\{[v_1/2+v_2/2]^2\right\} \\
&= R/4+R/4 \\
&= R/2
\end{aligned}
$$

In this problem each measurement has unity variance, so the variance of the averaged measurement is $1/2$. We can use Equation (5.19) to obtain

$$
\begin{aligned}
P_1^- &= \frac{1}{4}+1 \\
&= \frac{5}{4} \\
P_1^+ &= \left[\left(P_1^-\right)^{-1} + H^T R^{-1} H\right]^{-1} \\
&= (4/5+2)^{-1} \\
&= \frac{5}{14} \\
P_2^- &= \left(\frac{1}{4}\right)\left(\frac{5}{14}\right) + \frac{5}{4} \\
&= \frac{75}{56} \\
P_2^+ &= \left[\left(P_2^-\right)^{-1} + 2\right]^{-1} \\
&= \left(\frac{56}{75}+2\right)^{-1} \\
&= \frac{75}{206}
\end{aligned}
$$

Note that the P_i^- and P_i^+ quantities obtained here are the inverses of the $\mathcal{I}_i^-$ and $\mathcal{I}_i^+$ quantities obtained in part (a).

6.5 Prove that the singular values of a diagonal matrix are the magnitudes of the diagonal elements.

Solution:

Suppose that $P = \text{diag}(P_1, \cdots, P_n)$. Then $P^T P = \text{diag}(P_1^2, \cdots, P_n^2)$. The singular values of P are computed as

$$\begin{aligned} \sigma(P) &= \sqrt{\lambda(P^T P)} \\ &= \{|P_1|, \cdots, |P_n|\} \end{aligned}$$

QED

6.6 Prove that SS^T is symmetric positive semidefinite for any S matrix.

Solution:

For any compatibly dimensioned column vector x we have $x^T SS^T x = y^T y = \|y\|_2^2 \geq 0$, where $y = S^T x$. Since $x^T SS^T x \geq 0$ for all x, we see that SS^T is positive semidefinite. Also note that $(SS^T)^T = (S^T)^T S^T = SS^T$, so SS^T is also symmetric.
QED

6.7 Find an upper triangular matrix S (using only paper and pencil) such that

$$SS^T = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$$

Is your solution unique?

Solution:

$$\begin{aligned} \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \begin{bmatrix} S_{11} & 0 \\ S_{12} & S_{22} \end{bmatrix} &= \begin{bmatrix} S_{11}^2 + S_{12}^2 & S_{12}S_{22} \\ S_{12}S_{22} & S_{22}^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix} \end{aligned}$$

Equating the elements of the matrices on right sides of the above equations gives $S_{22} = 3$, $S_{12} = 1$, and $S_{11} = 0$.

$$S = \begin{bmatrix} 0 & 1 \\ 0 & 3 \end{bmatrix}$$

$-S$ will also solve the problem, so the above solution is not unique.

6.8 Find an upper triangular matrix S (using only paper and pencil) such that

$$SS^T = \begin{bmatrix} 5 & 2 & -2 \\ 2 & 2 & -1 \\ -2 & -1 & 1 \end{bmatrix}$$

How many solutions exist to this problem?

Solution:

$$
\begin{bmatrix} S_{11} & S_{12} & S_{13} \\ 0 & S_{22} & S_{23} \\ 0 & 0 & S_{33} \end{bmatrix} \begin{bmatrix} S_{11} & 0 & 0 \\ S_{12} & S_{22} & 0 \\ S_{13} & S_{23} & S_{33} \end{bmatrix} = \begin{bmatrix} S_{11}^2 + S_{12}^2 + S_{13}^2 & S_{12}S_{22} + S_{13}S_{23} & S_{13}S_{33} \\ S_{12}S_{22} + S_{13}S_{23} & S_{22}^2 + S_{23}^2 & S_{23}S_{33} \\ S_{13}S_{33} & S_{23}S_{33} & S_{33}^2 \end{bmatrix}
$$
$$
= \begin{bmatrix} 5 & 2 & -2 \\ 2 & 2 & -1 \\ -2 & -1 & 1 \end{bmatrix}
$$

Equating the elements of the matrices on the right sides of the above equations gives eight possible solutions.

$$
S = \begin{bmatrix} \pm 1 & 0 & -2 \\ 0 & \pm 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \text{ or } \begin{bmatrix} \pm 1 & 0 & 2 \\ 0 & \pm 1 & 1 \\ 0 & 0 & -1 \end{bmatrix}
$$

6.9 Verify Equation (6.70). Hint: Equate the two sides of the equation, take the trace, and solve for γ. Make sure to explain why taking the trace is valid.

Solution:

Equation (6.70) says

$$
I - a\phi\phi^T = (I - a\gamma\phi\phi^T)^2
$$

This gives

$$
\begin{aligned}
I - a\phi\phi^T &= I - 2a\gamma\phi\phi^T + a^2\gamma^2(\phi\phi^T)^2 \\
0 &= (a\phi\phi^T)^2\gamma^2 - 2a\gamma\phi\phi^T + a\phi\phi^T
\end{aligned}
$$

The matrix on the right in the above equation has a rank less than or equal to one (since ϕ is a column matrix). Therefore its eigenvalues are $\{0, \cdots, 0, \mu\}$, where μ might be nonzero. So if we adjust γ to make the trace of the matrix zero then all of the eigenvalues will be zero (since the trace is equal to the sum of the eigenvalues). This will make $\mu = 0$ which will reduce the rank of the matrix to zero, thereby making the matrix equal to 0. Taking the trace of the matrix on the right in the above equation gives

$$
\begin{aligned}
0 &= (a\phi^T\phi)^2\gamma^2 - 2a\phi^T\phi\gamma + a\phi^T\phi \\
&= a\phi^T\phi\gamma^2 - 2\gamma + 1 \\
&= (aR_{ik} - 1)\gamma^2 + 2\gamma - 1 \\
aR_{ik}\gamma^2 &= \gamma^2 - 2\gamma + 1 \\
\pm\gamma\sqrt{R_{ik}a} &= 1 - \gamma \\
\gamma \pm \gamma\sqrt{aR_{ik}} &= 1 \\
\gamma &= \frac{1}{1 \pm \sqrt{aR_{ik}}}
\end{aligned}
$$

QED

6.10 Suppose that an orthogonal matrix $\tilde{T}$ is desired to satisfy Equation (6.97), where Cholesky factorization is used to compute the matrix square roots on the left side of the equation. This equation can then be written as $U = \tilde{T}A$, where U is an upper triangular matrix. Show that such a transformation cannot be found unless the two-norm of the first column of A happens to be equal to $|U_{11}|$. [Note that this does not necessarily prevent the possibility of the transformation of Equation (6.97), because U could be nontriangular if nontriangular square root matrices are used to form the U matrix.]

Solution:

We have $U_1 = \tilde{T}A_1$, where U_1 and A_1 are the first columns of U and A. Since $\tilde{T}$ is orthogonal, $\tilde{T}^T\tilde{T} = I$. Since U is upper triangular, $U_1 = \begin{bmatrix} U_{11} & 0 & \cdots & 0 \end{bmatrix}^T$. This means that

$$\begin{aligned} U_1^T U_1 &= (\tilde{T}A_1)^T(\tilde{T}A_1) \\ U_{11}^2 &= A_1^T \tilde{T}^T \tilde{T} A_1 \\ &= A_1^T A_1 \\ |U_{11}| &= ||A_1||_2 \end{aligned}$$

QED

6.11 Use the Householder method (using only paper and pencil) to find an orthogonal T such that $TA = \begin{bmatrix} W \\ 0 \end{bmatrix}$ where W is a 2×2 matrix and

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 1 \\ 2 & 2 \end{bmatrix}$$

Solution:

Following the algorithm in Section 6.3.5.1 gives

$$\begin{aligned} \sigma_1 &= 3 \\ \beta_1 &= 1/12 \\ u^{(1)} &= \begin{bmatrix} 4 & 2 & 0 & 2 \end{bmatrix}^T \\ y^{(1)} &= \begin{bmatrix} 1 & 1 \end{bmatrix}^T \\ A^{(2)} &= \begin{bmatrix} -3 & -3 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\ \sigma_2 &= 1 \\ \beta_2 &= 1 \\ u^{(2)} &= \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T \\ y^{(2)} &= \begin{bmatrix} 0 & 1 \end{bmatrix}^T \end{aligned}$$

$$
\begin{aligned}
A^{(3)} &= \begin{bmatrix} -3 & -3 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\
W &= \begin{bmatrix} -3 & -3 \\ 0 & -1 \end{bmatrix} \\
T &= (I - \beta_2 u^{(2)} u^{(2)T})(I - \beta_1 u^{(1)} u^{(1)T}) \\
&= \begin{bmatrix} -1/3 & -2/3 & 0 & -2/3 \\ 0 & 0 & -1 & 0 \\ 2/3 & -2/3 & 0 & 1/3 \\ -2/3 & -1/3 & 0 & 2/3 \end{bmatrix}
\end{aligned}
$$

6.12 Use the modified Gram–Schmidt method (using only paper and pencil) to solve Problem 6.11.

Solution:

Following the algorithm in Section 6.3.5.2 gives

$$
\begin{aligned}
\sigma_1 &= 3 \\
W_1 &= \begin{bmatrix} 3 & 3 \end{bmatrix} \\
T_1 &= \begin{bmatrix} 1/3 & 2/3 & 0 & 2/3 \end{bmatrix} \\
A_2^{(2)} &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T \\
\sigma_2 &= 1 \\
W_2 &= \begin{bmatrix} 0 & 1 \end{bmatrix} \\
T_2 &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \\
T_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \\
T_4 &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \\
T_5 &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \\
T_6 &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Now we perform a standard Gram-Schmidt orthonormalization procedure on the last four rows of T (i.e., T_3 through T_6).

$$
\begin{aligned}
T_3 &= T_3 - (T_3 T_1^T)T_1 - (T_3 T_2^T)T_2 \\
&= \begin{bmatrix} 8/9 & -2/9 & 0 & -2/9 \end{bmatrix} \\
T_3 &= T_3/||T_3||_2 \\
&= \begin{bmatrix} 2\sqrt{2}/3 & -1/3\sqrt{2} & 0 & -1/3\sqrt{2} \end{bmatrix} \\
T_4 &= T_4 - (T_4 T_1^T)T_1 - (T_4 T_2^T)T_2 - (T_4 T_3^T)T_3 \\
&= \begin{bmatrix} 0 & 1/2 & 0 & 1/2 \end{bmatrix} \\
T_4 &= T_4/||T_4||_2 \\
&= \begin{bmatrix} 0 & \sqrt{2}/2 & 0 & -\sqrt{2}/2 \end{bmatrix}
\end{aligned}
$$

Since we have four nonzero rows of T the algorithm is complete.

$$T = \begin{bmatrix} 1/3 & 2/3 & 0 & 2/3 \\ 0 & 0 & 1 & 0 \\ 2\sqrt{2}/3 & -1/3\sqrt{2} & 0 & -1/3\sqrt{2} \\ 0 & \sqrt{2}/2 & 0 & -\sqrt{2}/2 \end{bmatrix}$$

6.13 Compute the U-D factorization (using only paper and pencil) for the matrix

$$P = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$$

Solution:

$$\begin{aligned} P &= UDU^T \\ &= \begin{bmatrix} 1 & u_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_{11} & 0 \\ 0 & d_{22} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ u_{12} & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix} &= \begin{bmatrix} d_{11} + u_{12}^2 d_{22} & u_{12} d_{22} \\ u_{12} d_{22} & d_{22} \end{bmatrix} \end{aligned}$$

Equating elements in the two matrices above gives $d_{22} = 9$, $u_{12} = 1/3$, and $d_{11} = 0$.

$$\begin{aligned} U &= \begin{bmatrix} 1 & 1/3 \\ 0 & 1 \end{bmatrix} \\ D &= \begin{bmatrix} 0 & 0 \\ 0 & 9 \end{bmatrix} \end{aligned}$$

Computer exercises

6.14 Consider the RLC circuit of Example 1.8 with $R = 100$ and $L = C = 1$. Suppose the applied voltage is continuous-time, zero-mean white noise with a standard deviation of 3. The initial capacitor voltage and inductor current are both zero. Discretize the system with a time step of 0.1. The discrete-time measurements consist of the capacitor voltage and the inductor current, both measurements containing zero-mean unity variance noise. Implement a sequential Kalman filter for the system. Simulate the system for 2 seconds. Let the initial state estimate be equal to the initial state, and the initial estimation covariance be equal to $0.1I$. Hint: Set the discrete-time process noise covariance $Q = Q_c \Delta t$, where Q_c is the covariance of the continuous-time process noise, and Δt is the discretization step size. Q will be nondiagonal, which means you need to use the algorithm in Section 2.7 to simulate the process noise.

a) Generate a plot showing the *a priori* variance of the capacitor voltage estimation error, and the two *a posteriori* variances of the capacitor voltage estimation error.

b) Generate a plot showing a typical trace of the true, *a posteriori* estimated, and measured capacitor voltage. What is the standard deviation of the capacitor voltage measurement error? What is the standard deviation of the capacitor voltage estimation error?

Solution:

a). Figure 6.1 shows the variance of the capacitor voltage estimation error.

b). Figure 6.2 shows the true, estimated, and measured capacitor voltage for a typical simulation. Your results may vary depending on the noise history that you realized. The measurement error has a standard deviation of 1. The estimation error has a standard deviation of about 0.3.

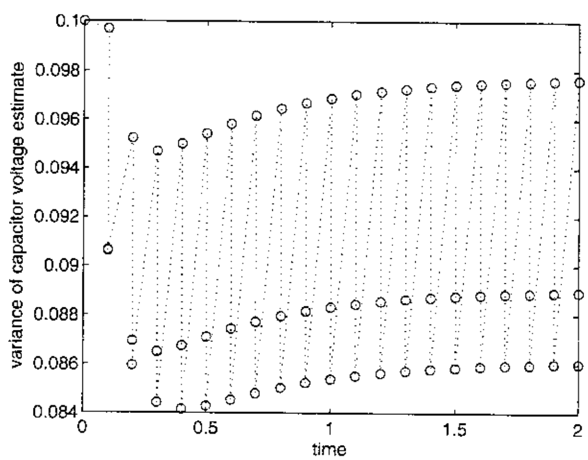

Figure 6.1 Solution to part (a) of Problem 6.14

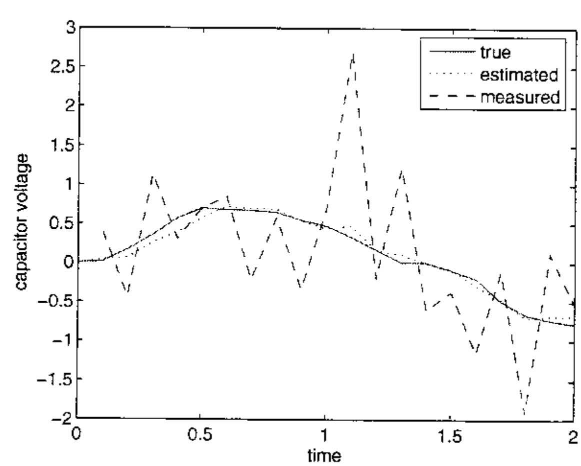

Figure 6.2 Solution to part (b) of Problem 6.14

6.15 The pitch motion of an aircraft flying at constant speed can be approximately described by the following equations [Ste94]:

$$
\begin{aligned}
\dot{x} &= \begin{bmatrix} -0.5680 & 17.9800 \\ 1.0000 & -1.2370 \end{bmatrix} x + \begin{bmatrix} 0.1750 & 0.1750 \\ -0.0010 & -0.0010 \end{bmatrix} u + \begin{bmatrix} 17.9800 \\ -1.2370 \end{bmatrix} w \\
y(t_k) &= x(t_k) + v_k
\end{aligned}
$$

where x_1 is the pitch rate, x_2 is the angle of attack, u consists of the elevator and flap angles, and w is disturbance due to wind. Suppose that the variance of the wind disturbance is 0.001, and the measurement variances are 0.3. Discretize the system with a step size of 0.01 and simulate the system and a square root Kalman filter for 100 time steps. Use an initial state of zero, an initial state estimate of zero, an initial estimation-error covariance of $0.01I$, and a control input of zero. Hint: Set the discrete-time process noise covariance $Q = Q_c\Delta t$, where Q_c is the covariance of the continuous-time process noise, and Δt is the discretization step size. Q will be nondiagonal, which means you need to use the algorithm in Section 2.7 to simulate the process noise.

a) Generate a plot showing the *a posteriori* variance of the estimation errors of the two states.

b) Generate a plot showing a typical trace of the true, *a posteriori* estimated, and measured pitch rate. What is the standard deviation of the pitch rate measurement error? What is the standard deviation of the pitch rate estimation error?

c) Generate a plot showing a typical trace of the true, *a posteriori* estimated, and measured angle of attack. What is the standard deviation of the angle of attack measurement error? What is the standard deviation of the angle of attack estimation error?

Solution:

a). Figure 6.3 shows the variance of the estimation error of the two states.

b). Figure 6.4 shows the true, estimated, and measured pitch rate for a typical simulation. Your results may vary depending on the noise history that you realized. The measurement error has a standard deviation of $\sqrt{0.3} \approx 0.55$. The estimation error has a standard deviation of about 0.19.

c). Figure 6.5 shows the true, estimated, and measured angle of attack for a typical simulation. Your results may vary depending on the noise history that you realized. The measurement error has a standard deviation of $\sqrt{0.3} \approx 0.55$. The estimation error has a standard deviation of about 0.15.

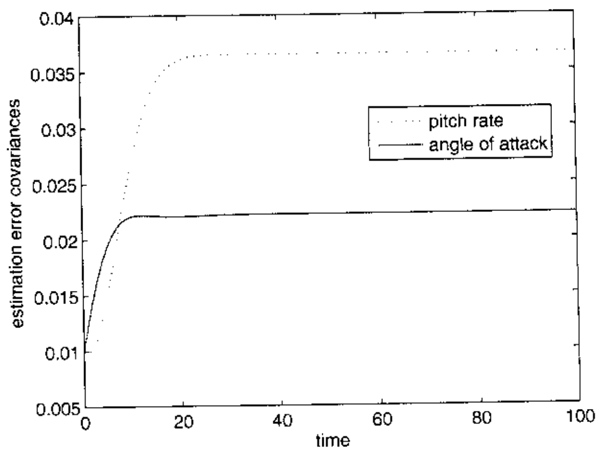

Figure 6.3 Solution to part (a) of Problem 6.15

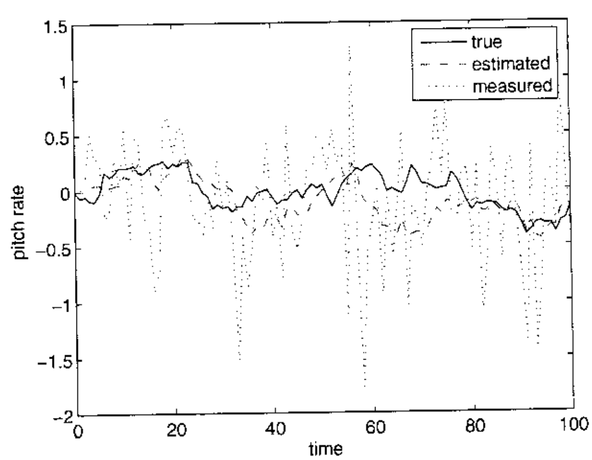

Figure 6.4 Solution to part (b) of Problem 6.15

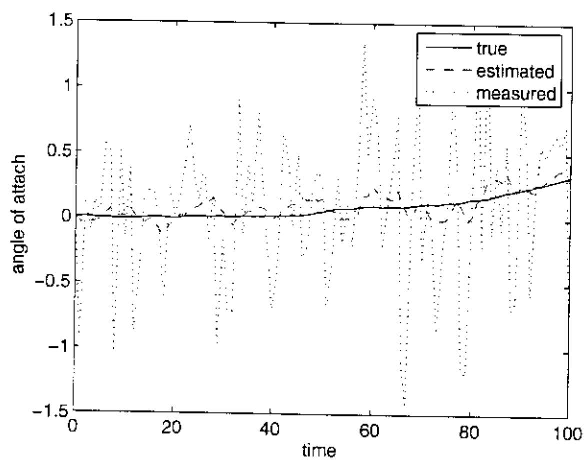

Figure 6.5 Solution to part (c) of Problem 6.15

CHAPTER 7

Kalman filter generalizations

Problems

Written exercises

7.1 Consider the scalar system

$$\begin{aligned} x_k &= \frac{1}{2}x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \\ v_k &= \frac{1}{2}v_{k-1} + \zeta_{k-1} \end{aligned}$$

where $w_k \sim (0, Q)$ and $\zeta_k \sim (0, Q_\zeta)$. Let $Q = Q_\zeta = 1$.

a) Design a Kalman filter in which the dynamics of the measurement noise v_k are ignored and it is assumed that v_k is white noise with a variance of Q_ζ. Based on the incorrect Kalman filter equations, what does the Kalman filter think that the steady-state *a posteriori* estimation covariance is?

b) Based on the incorrect Kalman filter equations, what is the true steady-state *a posteriori* estimation covariance $E(e_k^2)$? Hint: Find a recursive equation for $E(e_k^2)$ in terms of $E(e_{k-1}^2)$, $E(w_{k-1}^2)$, $E(v_k^2)$, and $E(e_{k-1}v_k)$, then solve for the steady-state value of $E(e_k^2)$.

c) Design a Kalman filter using the state augmentation approach in which the dynamics of the measurement noise are correctly taken into account. What is the steady-state estimation covariance? Hint: You may need to use MATLAB's DARE function to solve the steady-state Riccati equation that is associated with this question.

Solution:

a). If it is assumed that $v_k \sim (0, Q_\zeta)$, the Kalman filter equations are given as

$$\begin{aligned} P_k^- &= F P_{k-1}^+ F^T + Q \\ K_k &= P_k^- (P_k^- + Q_\zeta)^{-1} \\ &= P_k^+ Q_\zeta^{-1} \\ \hat{x}_k^- &= \frac{1}{2}\hat{x}_{k-1}^+ \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - \hat{x}_k^-) \\ P_k^+ &= (1 - K_k)P_k^- \end{aligned}$$

Solving this for steady state gives

$$\begin{aligned} \lim_{k\to\infty} P_k^+ &= \frac{\sqrt{65} - 7}{2} \\ &\approx 0.53 \end{aligned}$$

This is what the Kalman filter thinks the steady-state *a posteriori* estimation-error covariance is equal to.

b). Note that

$$\begin{aligned} e_k &= x_k - \hat{x}_k^+ \\ &= \frac{1}{2}x_{k-1} + w_{k-1} - \frac{1}{2}\hat{x}_{k-1}^+ - K_k(y_k - \frac{1}{2}\hat{x}_{k-1}^+) \\ &= \frac{1}{2}e_{k-1} + w_{k-1} - K_k(\frac{1}{2}e_{k-1} + v_k + w_{k-1}) \\ E(e_k^2) &= \frac{1}{4}(1 - K_k)^2 E(e_{k-1}^2) + (1 - K_k)^2 Q + K_k^2 E(v_k^2) - K_k(1 - K_k)E(e_{k-1}v_k) \end{aligned}$$

In order to solve this we first need to derive $E(v_k^2)$ as follows.

$$\begin{aligned} E(v_k^2) &= E\left[\left(\frac{1}{2}v_{k-1} + \zeta_{k-1}\right)^2\right] \\ &= \frac{1}{4}E(v_{k-1}^2) + E(\zeta_{k-1}^2) \\ &= \frac{1}{4}E(v_{k-1}^2) + 1 \end{aligned}$$

In steady state this has the solution $E(v_k^2) = 4/3$. Next we derive $E(e_{k-1}v_k)$ as follows.

$$E(e_{k-1}v_k) = E\left[\left(x_{k-1} - \hat{x}_{k-1}^+\right)\left(\frac{1}{2}v_{k-1} + \zeta_{k-1}\right)\right]$$

$$
\begin{aligned}
&= \frac{1}{2}E[(x_{k-1} - \hat{x}_{k-1}^+)v_{k-1}] \\
&= -\frac{1}{2}E[\hat{x}_{k-1}^+ v_{k-1}] \\
&= -\frac{1}{2}E\left[\left(\frac{1}{2}\hat{x}_{k-2}^+ + K_{k-1}(y_{k-1} - \frac{1}{2}\hat{x}_{k-2}^+)\right) v_{k-1}\right] \\
&= -\frac{1}{2}E(K_{k-1}y_{k-1}v_{k-1}) \\
&= -\frac{1}{2}K_{k-1}E[(x_{k-1} + v_{k-1})v_{k-1}] \\
&= -\frac{1}{2}K_{k-1}E(v_{k-1}^2) \\
&= -\frac{2K_{k-1}}{3}
\end{aligned}
$$

From part (a) we see that the steady-state value of K_k is $(\sqrt{65}-7)/2$. Therefore the steady-state value of $E(e_{k-1}v_k)$ is equal to $(\sqrt{65}-7)/3$. Now we substitute the values of $E(v_k^2)$ and $E(e_{k-1}v_k)$ into the earlier equation for $E(e_k^2)$ and assume steady state to obtain

$$E(e^2) = \frac{1}{4}(1-K)^2E(e^2) + (1-K)^2Q + \frac{4}{3}K^2 - \frac{\sqrt{65}-7}{3}K(1-K)$$

Solving for $E(e^2)$ gives

$$
\begin{aligned}
E(e^2) &= \frac{(1-K)^2 + 4K^2/3 + 2K^2(1-K)/3}{1-(1-K)^2/4} \\
&\approx 0.72
\end{aligned}
$$

c). The augmented system is given as

$$
\begin{aligned}
\begin{bmatrix} x_k \\ v_k \end{bmatrix} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} w_{k-1} \\ \zeta_{k-1} \end{bmatrix} \\
y_k &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix}
\end{aligned}
$$

The process noise covariance Q is a 2×2 identity matrix, and the measurement noise covariance R is zero. The steady-state *a priori* estimation-error covariance is given as

$$P^- = FP^-F^T - FP^-H^T(HP^-H^T)^{-1}HP^-F^T + Q$$

This can be solved using the MATLAB function DARE(F^T, H^T, Q, R). Alternatively, the matrix elements on the left and right side of the above equation can be equated to obtain the simultaneous equations

$$
\begin{aligned}
P_{11}^- &= \frac{P_{11}^-}{4} - \frac{(P_{11}^- + P_{12}^-)^2}{4(P_{11}^- + 2P_{12}^- + P_{22}^-)} + 1 \\
P_{12}^- &= \frac{P_{12}^-}{4} - \frac{(P_{11}^- + P_{12}^-)(P_{22}^- + P_{12}^-)}{4(P_{11}^- + 2P_{12}^- + P_{22}^-)} \\
P_{22}^- &= \frac{P_{22}^-}{4} - \frac{(P_{22}^- + P_{12}^-)^2}{4(P_{11}^- + 2P_{12}^- + P_{22}^-)} + 1
\end{aligned}
$$

Either way that the equation is solved gives the steady-state *a priori* estimation-error covariance

$$P^- = \begin{bmatrix} 7/6 & -1/6 \\ -1/6 & 7/6 \end{bmatrix}$$

Now we find the steady-state *a posteriori* estimation-error covariance as

$$\begin{aligned} P^+ &= P^- - KHP^- \\ &= P^- - P^-H^T(HP^-H^T)^{-1}HP^- \\ &= \begin{bmatrix} 2/3 & -2/3 \\ -2/3 & 2/3 \end{bmatrix} \end{aligned}$$

We see that the steady-state *a posteriori* estimation-error covariance for x_k is $2/3$.

7.2 Show that the Kalman filter for an LTI system with a noise-free scalar measurement that satisfies the equation $(HQH^T)Q = QH^THQ$ has a steady-state *a posteriori* covariance of zero.

Solution:

Recall the Kalman filter equations

$$\begin{aligned} P_k^- &= FP_{k-1}^+F^T + Q \\ K_k &= P_k^-H^T(HP_k^-H^T + R)^{-1} \\ P_k^+ &= (I - K_kH)P_k^- \end{aligned}$$

Set $R = 0$ and substitute for P_k^- and K_k into the P_k^+ equation to get the following equation for the steady-state value of the *a posteriori* covariance.

$$P = FPF^T + Q - (FPF^T + Q)H^T \left[H(FPF^T + Q)H^T\right]^{-1} H(FPF^T + Q)$$

Plug in $P = 0$ to see what happens.

$$0 = Q - QH^T(HQH^T)^{-1}HQ$$

If H is a row vector then this can be written as

$$(HQH^T)Q = QH^THQ$$

So if this equation is satisfied then $P = 0$ is a solution for the steady-state *a posteriori* covariance.
QED

7.3 Consider the scalar system

$$\begin{aligned} x_k &= x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \end{aligned}$$

where $w_k \sim (0, Q)$ and $v_k \sim (0, R)$ are white noise processes with $Q = R = 1$. Suppose that $E(w_k v_{k+1}) = M = 1$.

a) Design a Kalman filter in which the correlation between w_k and v_{k+1} is ignored. Based on the incorrect Kalman filter equations, what does it appear that the steady-state *a posteriori* estimation covariance is?

b) For the Kalman filter designed above, write a recursive equation for the *a posteriori* estimation error $e_k = x_k - \hat{x}_k^+$. Use this equation to find the steady-state solution to $E(e_k^2)$.

c) Design a Kalman filter in which the correlation between w_k and v_{k+1} is correctly taken into account. Show that the steady-state *a posteriori* estimation covariance is zero. Explain why the estimation covariance goes to zero in spite of the existence of process noise and measurement noise. (Hint: Use the correlation between w_k and v_{k+1} to write an equivalent two-state system, and then use the results of Problem 7.2.)

Solution:

a). The Kalman filter equations when it is assumed that $M = 0$ are given as

$$\begin{aligned}
P_k^- &= P_{k-1}^+ + Q \\
K_k &= P_k^-(P_k^- + R)^{-1} \\
&= P_k^+ R^{-1} \\
\hat{x}_k^- &= \hat{x}_{k-1}^+ \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - \hat{x}_k^-) \\
P_k^+ &= (1 - K_k)P_k^-
\end{aligned}$$

Solving this for steady state gives

$$\begin{aligned}
\lim_{k\to\infty} P_k^+ &= \frac{\sqrt{5}-1}{2} \\
&\approx 0.6180
\end{aligned}$$

This is what the Kalman filter thinks the steady-state *a posteriori* estimation-error covariance is equal to.

b). A recursive equation for the estimation error can be written as

$$\begin{aligned}
x_{k+1} - \hat{x}_{k+1}^+ &= (x_k + w_k) - (1 - K_k)\hat{x}_k^+ - K_k y_{k+1} \\
&= x_k + w_k - (1 - K_k)\hat{x}_k^+ - K_k x_k - K_k w_k - K_k v_{k+1} \\
e_{k-1} &= (1 - K_k)e_k + (1 - K_k)w_k - K_k v_{k+1}
\end{aligned}$$

Taking the expected value of the square of both sides gives

$$E(e_{k+1}^2) = (1 - K_k)^2 E(e_k^2) + (1 - K_k)^2 Q + K_k^2 R - 2K_k(1 - K_k)M$$

In steady state $E(e_{k+1}^2) = E(e_k^2)$, and $K_k = K$ (a constant). Solving for $E(e_k^2)$ gives

$$E(e^2) = \frac{(1-K)^2 Q + K^2 R - 2K(1-K)M}{1 - (1-K)^2}$$

Substituting $Q = R = M = 1$, and the steady-state value of $K \approx 0.6180$, gives

$$E(e^2) = 0.0652$$

(Also note that if we substitute $M = 0$ into the $E(e^2)$ equation we get $E(e^2) = 0.6180$, which is the same as the steady-state *a posteriori* covariance that calculated in part (a).)

c). The Kalman filter that takes M into account is given as

$$\begin{aligned}
P_k^- &= P_{k-1}^+ + Q \\
K_k &= (P_k^- + M)(P_k^- + 2M + R)^{-1} \\
\hat{x}_k^- &= \hat{x}_{k-1}^+ \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - \hat{x}_k^-) \\
P_k^+ &= (1 - K_k)P_k^- - K_k M
\end{aligned}$$

Solving this for steady state gives

$$\lim_{k\to\infty} P_k^+ = 0$$

How can this be? Since $E(w_k^2) = E(v_{k+1}^2) = E(w_k v_{k+1}) = 1$, we see that $v_{k+1} = w_k$. (This can be deduced from the results of Problem 2.9.) Therefore

$$\begin{aligned}
y_k &= x_k + v_k \\
&= x_k + w_{k-1} \\
&= x_k + (x_k - x_{k-1}) \\
&= 2x_k - x_{k-1}
\end{aligned}$$

The system can therefore be rewritten as

$$\begin{aligned}
\begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-2} \\ x_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ w_{k-1} \end{bmatrix} \\
y_k &= \begin{bmatrix} -1 & 2 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix}
\end{aligned}$$

We see that this equivalent system satisfies the conditions of Problem 7.2, which means that the Kalman filter for this system has a covariance that tends to zero as time tends to infinity.

7.4 Consider the system

$$\begin{aligned}
x_k &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} x_{k-1} + w_{k-1} \\
y_k &= \begin{bmatrix} 1 & 1 \end{bmatrix} x_k + v_k
\end{aligned}$$

where $w_k \sim (0, Q)$ and $Q = I$.

a) Find one matrix square root of Q.

b) Is (F, H) observable?

c) Is (F, H) detectable?

d) Is (F, G) controllable for all G such that $GG^T = Q$?

e) Is (F, G) stabilizable for all G such that $GG^T = Q$?

f) Use the above results to specify how many positive definite solutions exist to the DARE that is associated with the Kalman filter for this problem.

g) Use the above results to specify whether or not the steady-state Kalman filter for this system is stable.

Solution:

a). $G = I$ is one matrix square root of Q.

b). The observability matrix of (F, H) is given as

$$\begin{aligned} \mathcal{O} &= \begin{bmatrix} H \\ HF \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ 1/2 & 1/2 \end{bmatrix} \end{aligned}$$

which is not full rank, therefore the system is not observable.

c). The system is stable, which is a subset of detectability, therefore the system is detectable.

d). If G is a square root of Q then

$$\begin{aligned} GG^T &= Q \\ \begin{bmatrix} G_{11}^2 + G_{12}^2 & G_{11}G_{21} + G_{12}G_{22} \\ G_{11}G_{21} + G_{12}G_{22} & G_{21}^2 + G_{22}^2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

The controllability matrix of (F, G) is given as

$$\begin{aligned} \mathcal{C} &= \begin{bmatrix} G & FG \end{bmatrix} \\ &= \begin{bmatrix} G_{11} & G_{12} & G_{11}/2 & G_{12}/2 \\ G_{21} & G_{22} & G_{21}/2 & G_{22}/2 \end{bmatrix} \end{aligned}$$

In order for the system to be uncontrollable, the controllability matrix must have a rank of one. That means that $G_{12} = \alpha G_{11}$ and $G_{22} = \alpha G_{21}$ for some constant α. If that is true then

$$GG^T = \begin{bmatrix} G_{11}^2(1+\alpha^2) & G_{11}G_{21}(1+\alpha^2) \\ G_{11}G_{21}(1+\alpha^2) & G_{21}^2(1+\alpha^2) \end{bmatrix}$$

It is impossible to equate this with the identity matrix, so that means the controllability matrix is full rank. So the answer is: Yes, (F, G) is controllable for all G such that $GG^T = Q$.

e). The system is stable, which is a subset of stabilizability, therefore the system is stabilizable.

f). Combining the above results with Theorem 23, we see that the DARE that is associated with the Kalman filter for this problem has a unique positive semidefinite solution.

g). Combining the above results with Theorem 23, we see that the steady-state Kalman filter that is associated with this problem is stable.

7.5 Prove that the matrix $\mathcal{H}$ in Equation (7.85) is symplectic.

Solution:

We start with

$$\begin{aligned}
\mathcal{H} &= \begin{bmatrix} F^{-T} & F^{-T}H^TR^{-1}H \\ QF^{-T} & F+QF^{-T}H^TR^{-1}H \end{bmatrix} \\
J &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \\
J^{-1} &= \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}
\end{aligned}$$

From this we can compute

$$J^{-1}\mathcal{H}J = \begin{bmatrix} F^T + H^TR^{-1}HF^{-1}Q & -H^TR^{-1}HF^{-1} \\ -F^{-1}Q & F^{-1} \end{bmatrix}$$

Multiplying this matrix by $\mathcal{H}$ gives the identity matrix, which proves that $J^{-1}\mathcal{H}J = \mathcal{H}^{-1}$.
QED

7.6 In this problem, we will use the shorthand notation $P = P^+$ and $M = P^-$. Use the following procedure to find α as a function of β for the α-β filter [Bar01].

a) Use the time-update equation for M to solve for the three unique elements of P as a function of the three unique elements of M.

b) Use the measurement-update equation for P to solve for the three unique elements of P as a function of the three unique elements of M.

c) Equate the sets of equations from the two steps above to get expressions for $M_{11}K_1$, $M_{12}K_1$, and $M_{12}K_2$, that do not have any P_{ij} terms.

d) Use Equation (7.64) to solve for M_{11} and M_{12}.

e) Combine the five equations from the two previous steps to get a single equation with K_1 and K_2 that does not have any M_{ij} terms.

f) Replace K_1 and K_2 in the previously obtained equation with α and β from Equation (7.63), then solve for α as a function of β.

Solution:

a). From the time update expression for M from Equation (7.62) we get

$$\begin{aligned}
M &= FPF^T + Q \\
P &= F^{-1}(M-Q)F^{-T}
\end{aligned}$$

Substituting for F and Q from Equation (7.60) and simplifying gives

$$\begin{aligned}
P_{11} &= M_{11} - 2TM_{12} + T^2M_{22} - T^4\sigma^2/4 \\
P_{12} &= M_{12} + T^3\sigma^2/2 - M_{22}T \\
P_{22} &= M_{22} - T^2\sigma^2
\end{aligned}$$

b). We use the measurement update expression $P = M - KHM$ from Equation (7.62), substitute for H from Equation (7.60), substitute for K from

Equation (7.64), and then simplify to get

$$\begin{aligned} P_{11} &= M_{11} - M_{11}K_1 \\ P_{12} &= M_{12} - M_{12}K_1 \\ P_{22} &= M_{22} - M_{12}K_2 \end{aligned}$$

c). Equating the two sets of equations for P_{ij} from the two steps above and solving gives

$$\begin{aligned} M_{11}K_1 &= 2TM_{12} - T^2M_{22} + T^4\sigma^2/4 \\ M_{12}K_1 &= M_{22}T - T^3\sigma^2/2 \\ M_{12}K_2 &= T^2\sigma^2 \end{aligned}$$

d). From Equation (7.64) we obtain

$$\begin{aligned} M_{11} &= \frac{RK_1}{1-K_1} \\ M_{12} &= \frac{RK_2}{1-K_1} \end{aligned}$$

e). We use the $M_{11}K_1$ equation that we obtained in part (c), and substitute for M_{11} and M_{12} from part (d) to obtain an equation that we will call Equation (A). We then use the $M_{12}K_1$ and $M_{12}K_2$ equations from part (c) to solve for M_{22} in terms of M_{12}, K_1, and K_2, and substitute this expression for M_{22} in Equation (A). After some simplification this gives

$$K_1^2 - 2TK_2 + TK_1K_2 + T^2K_2^2/4 = 0$$

f). Noting from Equation (7.63) that $\alpha = K_1$ and $\beta = TK_2$, the equation obtained in part (e) can be written as

$$\alpha^2 - 2\beta + \alpha\beta + \beta^2/4 = 0$$

Solving for α gives

$$\alpha = \sqrt{2\beta} - \beta/2$$

(We took the positive solution for α to get a stable filter.)

7.7 Prove the properties of symplectic matrices that are listed immediately following Equation (7.86).

Solution:

a). Since $\mathcal{H}^{-1}$ is part of the definition of a symplectic matrix from Equation (7.86), $\mathcal{H}$ is nonsingular by definition and therefore does not have any zero eigenvalues.
QED

b). Since $\mathcal{H}^{-1} = J^{-1}\mathcal{H}^T J$, we obtain

$$\begin{aligned}
|\lambda I - \mathcal{H}^{-1}| &= |\lambda I - J^{-1}\mathcal{H}^T J| \\
&= |J^{-1}(\lambda I - \mathcal{H}^T)J| \\
&= |J^{-1}||\lambda I - \mathcal{H}^T||J| \\
&= |\lambda I - \mathcal{H}|
\end{aligned}$$

So if λ is eigenvalue of $\mathcal{H}$ then it is also an eigenvalue of $\mathcal{H}^{-1}$. But it is also true that if λ is an eigenvalue of $\mathcal{H}$ then λ^{-1} is an eigenvalue of $\mathcal{H}^{-1}$. The only way to reconcile these two facts is to realize that if λ is an eigenvalue of $\mathcal{H}$ then λ^{-1} is also an eigenvalue of $\mathcal{H}$.
QED

c). Since $\mathcal{H}^{-1} = J^{-1}\mathcal{H}^T J$, we obtain

$$\begin{aligned}
|\mathcal{H}^{-1}| &= |J^{-1}\mathcal{H}^T J| \\
&= |J^{-1}||\mathcal{H}^T||J| \\
&= |\mathcal{H}|
\end{aligned}$$

But it is also true that $|\mathcal{H}^{-1}| = |\mathcal{H}|^{-1}$. Combining this with the above equation, we see that $|\mathcal{H}| = |\mathcal{H}|^{-1}$, which means that $|\mathcal{H}| = \pm 1$.
QED

7.8 Recall that the steady-state, zero-input, one-step formulation of the *a posteriori* Kalman filter can be written as

$$\begin{aligned}
\hat{x}_k^+ &= (I - KH)F\hat{x}_{k-1}^+ + Ky_k \\
\hat{y}_k &= H\hat{x}_k^+
\end{aligned}$$

Prove that if (F, H) is observable and $(I - HK)$ is full rank, then the Kalman filter in the above equation is an observable system. Hint: $H(I - KH) = (I - HK)H$.

Solution

The observability of the original system is determined by the rank of

$$Q_s = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix}$$

The original system is observable if $Q_s x \neq 0$ for all $x \neq 0$. The observability of the Kalman filter is determined by the rank of

$$Q_f = \begin{bmatrix} H \\ H[(I - KH)F] \\ \vdots \\ H[(I - KH)F]^{n-1} \end{bmatrix}$$

The Kalman filter is observable if $Q_f x \neq 0$ for all $x \neq 0$. Suppose that $(I - HK)$ is nonsingular and that the Kalman filter is unobservable. Then there exists some nonzero x such that

$$
\begin{aligned}
Hx &= 0 \\
H[(I-KH)F]x &= 0 \\
H[(I-KH)F]^2 x &= 0 \\
&\cdots \\
H[(I-KH)F]^{n-1} x &= 0
\end{aligned}
$$

Consider the second equation above. Note that $H[(I-KH)F]x = (I-HK)HFx$, and $(I - HK)$ is nonsingular. So if $H[(I-KH)F]x = 0$ then $HFx = 0$. Now consider the third equation above. Note that

$$
\begin{aligned}
H[(I-KH)F]^2 x &= H(I-KH)F(I-KH)Fx \\
&= H(I-KH)F^2 x - H(I-KH)FKHFx \\
&= H(I-KH)F^2 x \\
&= (I-HK)HF^2 x
\end{aligned}
$$

where the third equality follows because we showed earlier that $HFx = 0$. If the above expression is equal to 0, then since $(I - HK)$ is nonsingular, it must be true that $HF^2 = 0$. Continuing this development we see that if $H[(I-KH)F]^k x = 0$ then $HF^k x = 0$ for any $k \in [0, n-1]$. This shows that if $Q_f x = 0$ then $Q_s x = 0$. This proves that if the filter is unobservable then the original system is unobservable. Therefore, if the original system is observable the filter is also observable.
QED

7.9 Suppose you have a two-state Newtonian system of the type described in Section 7.3.1. The sample time is 1 and the variance of the acceleration noise is 1. A requirement is given to estimate the position with an *a posteriori* steady-state variance of 1 or less. What is the largest measurement variance that will meet the requirement?

Solution:

From Equation (7.71) we can find the *a posteriori* steady-state variance P_{11}^+ as a function of R. Plotting P_{11}^+ as a function of R we see that $R \leq 1.46$ to meet the requirement.

Computer exercises

7.10 Consider the system described in Problem 7.1. Implement the Kalman filter that assumes white noise and the Kalman filter that assumes colored noise. Numerically calculate the RMS *a posteriori* estimation-error variance and verify that it matches the analytically calculated values from your answer to Problem 7.1.

Solution:

Running the simulations for 2000 time steps gives RMS estimation errors of about 0.75 for the white Kalman filter, and 0.68 for the colored Kalman filter. These numerical results are pretty close to the analytically determined values of 0.72 and 0.67. Your results may vary depending on your particular noise history, but they should be close to these numbers.

7.11 Plot the α and β parameters of the α-β filter as a function of λ. Use a log scale for λ with a range of 10^{-3} to 10^{3}. What are the limiting values of α and β as $\lambda \to 0$? Does this make intuitive sense? What are the limiting values of α and β as $\lambda \to \infty$?

Solution:

The equations are taken from Section 7.3.1, and Figure 7.1 shows the results. The limiting values of α and β are both 0 as $\lambda \to 0$. This makes sense because if $\lambda \to 0$ that means that $R \to \infty$, or else $\sigma_w \to 0$. In either case we ignore the measurements and base our state estimate entirely on the system model. As $\lambda \to \infty$ we see from Figure 7.1 that $\alpha \to 1$ and $\beta \to 2$.

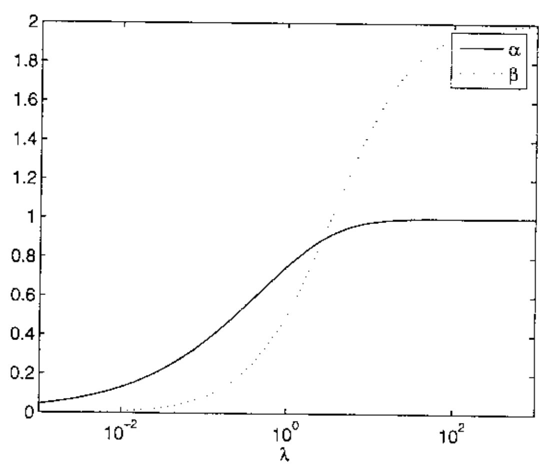

Figure 7.1 Solution to Problem 7.11

7.12 Plot the α, β, and γ parameters of the α-β-γ filter as a function of λ. Use a log scale for λ with a range of 10^{-3} to 10^{3}. What are the limiting values of α, β, and γ as $\lambda \to 0$? Does this make intuitive sense? What are the limiting values of α, β, and γ as $\lambda \to \infty$?

Solution:

The equations are taken from Section 7.3.2, and Figure 7.2 shows the results. The limiting values of α, β, and γ are all 0 as $\lambda \to 0$. This makes sense because if $\lambda \to 0$ that means that $R \to \infty$, or else $\sigma_w \to 0$. In either case we ignore the measurements and base our state estimate entirely on the system model. As $\lambda \to \infty$ we see from Figure 7.2 that $\alpha \to 1$, $\beta \to 2$, and $\gamma \to 4$.

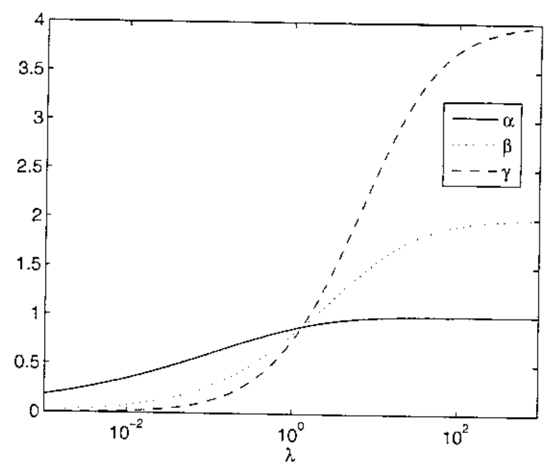

Figure 7.2 Solution to Problem 7.12

7.13 A simple model of the ingestion and metabolism of a drug is given as

$$
\begin{aligned}
\dot{x}_1 &= -k_1 x_1 + u \\
\dot{x}_2 &= k_1 x_1 - k_2 x_2 \\
y(t_k) &= x_2(t_k) + v(t_k)
\end{aligned}
$$

where the units of time are days, x_1 is the mass of the drug in the gastrointestinal tract, x_2 is the mass of the drug in the bloodstream, and u is the ingestion rate of the drug. Suppose that $k_1 = k_2 = 1$. The measurement noise $v(t_k)$ is zero-mean and unity variance. The initial state, estimate, and covariance are

$$
\begin{aligned}
x(0) &= \begin{bmatrix} 0.8 \\ 0 \end{bmatrix} \\
\hat{x}(0) &= x(0) \\
P(0) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}
$$

It is known from physical constraints that $x_1 \in [0.8, 1]$.

a) Discretize the system with a step size of 1 hour.

b) Implement the discrete-time Kalman filter, the projection-based constrained Kalman filter with $W = I$, and the pdf truncation constrained filter. Run simulations of these filters for a three-day period. Plot the magnitude of the x_1 estimation error for the three filters. Which filter appears to perform best? Which filter appears to perform worst?

Solution:

a). Starting with the initial state equation $\dot{x} = Ax + Bu$, discretization with a step size of Δt results in

$$\begin{aligned} x_{k+1} &= Fx_k + Gu_k \\ F &= e^{A\Delta t} \\ G &= (F - I)A^{-1}B \end{aligned}$$

b). Figure 7.3 shows a typical plot of estimation errors. Your results may vary depending on the particular noise history that was realized during your simulation. Sometimes the projection based constrained filter performs best, and sometimes the pdf truncation constrained filter performs best. The unconstrained filter always performs worst.

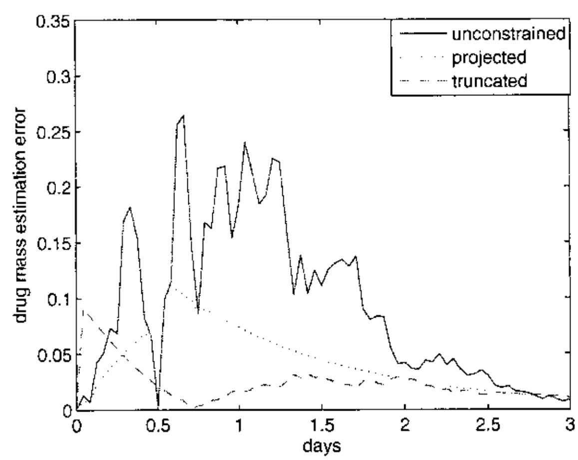

Figure 7.3 Solution to Problem 7.13

CHAPTER 8

The continuous-time Kalman filter

Problems

Written exercises

8.1 Suppose you have two discrete-time systems with identity transition matrices driven with stationary zero-mean white noise. The first system has a sample period of T, and the second system has a sample period of T/n for some integer $n > 1$. The noise in the first system has a covariance of Q. What should the covariance of the noise in the second system be in order for both states to have the same covariance at times kT $(k = 0, 1, 2, \cdots)$?

Solution:

Use x to denote the state of the first system, and z to denote the state of the second system. The variance of the state of the first system is $E(x_k^2) = kQ$. If the variance of the noise in the second system is Q', then the variance of the state of the second system is $E(z_j^2) = jQ'$. In order for these variances to both occur at the same time we must have $kT = jT/n$, which means that $j = kn$. In order for the variances to have the same value we must have $kQ = jQ' = knQ'$, which means that $Q' = Q/n$.

8.2 Show that for a general time-varying matrix $Y(t)$, if $\dot{Y} = AY + YA^T$, where A is a constant matrix, then $Y(t) = \exp(At)Y(0)\exp(A^T t)$.

Solution:

Differentiating the equation $Y(t) = \exp(At)Y(0)\exp(A^T t)$ gives

$$\begin{aligned}
\dot{Y} &= A\exp(At)Y(0)\exp(A^T t) + \exp(At)Y(0)\exp(A^T t)A^T \\
&= AY + YA^T
\end{aligned}$$

Also note that substituting $t = 0$ into the equation $Y(t) = \exp(At)Y(0)\exp(A^T t)$ results in $Y(0) = Y(0)$.
QED

8.3 Suppose you have a third-order Newtonian system with

$$\begin{aligned}
A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\
C &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\
Q &= \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \\
R &= 1
\end{aligned}$$

with $P(0) = I$.

a) What is the rank of $\dot{P}(0)$? How much computational savings in integration effort can be obtained by using the Chandrasekhar algorithm to find the Kalman gain for this system?

b) Find M_1 and M_2 such that $\dot{P}(0) = M_1 M_1^T - M_2 M_2^T$.

Solution:

a).

$$\begin{aligned}
\dot{P}(0) &= AP(0) + P(0)A^T - P(0)C^T R^{-1} C P(0) + Q \\
&= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\end{aligned}$$

The rank of $P(0)$ is $\alpha = 1$. The number of integrations for the Chandrasekhar algorithm is $n(\alpha + r) = 6$, and the number of integrations for the standard Kalman filter is $n(n+1)/2 = 6$, so the Chandrasekhar algorithm does not give any computational savings for this system.

b). The eigendata of $\dot{P}(0)$ are found as

$$D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$
\begin{aligned}
S &= \begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{3} & -1/\sqrt{2} & 1/\sqrt{6} \\ 1/\sqrt{3} & 0 & -\sqrt{2/3} \end{bmatrix} \\
&= \begin{bmatrix} S_{11} & S_{13} \\ S_{21} & S_{23} \\ S_{31} & S_{33} \end{bmatrix}
\end{aligned}
$$

where S_{11}, S_{21}, and S_{31} are scalars, and S_{13}, S_{23}, and S_{33} are 1×2 vectors. The S_{j2} parts of the matrix partition do not exist because $\dot{P}(0)$ does not have any negative eigenvalues. From this we can obtain

$$
\begin{aligned}
M_1 &= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T \\
M_2 &= 0
\end{aligned}
$$

8.4 Show that if S is upper triangular, then $\dot{S}$ and S^{-1} are also upper triangular.

Solution:

a). If $S(t)$ is upper triangular, then $S_{ij}(t) = 0$ for $i > j$. That means that $\dot{S}_{ij}(t) = 0$ for $i > j$, which means that $\dot{S}$ is upper triangular.
QED

b). Suppose that S is an upper triangular 3×3 matrix, and Q is its upper triangular inverse. Then

$$
\begin{aligned}
SQ &= \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ 0 & S_{22} & S_{23} \\ 0 & 0 & S_{33} \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ 0 & Q_{22} & Q_{23} \\ 0 & 0 & Q_{33} \end{bmatrix} \\
&= \begin{bmatrix} S_{11}Q_{11} & S_{11}Q_{12} + S_{12}Q_{22} & S_{11}Q_{13} + S_{12}Q_{23} + S_{13}Q_{33} \\ 0 & S_{22}Q_{22} & S_{22}Q_{23} + S_{23}Q_{33} \\ 0 & 0 & S_{33}Q_{33} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Equating the diagonal elements of these matrices uniquely solves for the Q_{ii} elements. Next equating the first upper diagonal elements uniquely solves for the $Q_{i,i+1}$ elements. Next equating the second upper diagonal element uniquely solves for the $Q_{i,i+2}$ element. If we proceed from this 3×3 case to the general case we see that the inverse of an upper triangular matrix is always upper triangular.
QED

8.5 Show that the product of upper triangular matrices is another upper triangular matrix.

Solution:

Suppose Q and S are upper triangular. Then

$$\begin{aligned} QS &= \begin{bmatrix} Q_{11} & \cdots & Q_{1n} \\ 0 & \ddots & \vdots \\ \ddots & 0 & Q_{nn} \end{bmatrix} \begin{bmatrix} S_{11} & \cdots & S_{1n} \\ 0 & \ddots & \vdots \\ \ddots & 0 & S_{nn} \end{bmatrix} \\ &= \begin{bmatrix} Q_{11}S_{11} & \cdots & \sum Q_{1j}S_{jn} \\ 0 & \ddots & \vdots \\ \ddots & 0 & Q_{nn}S_{nn} \end{bmatrix} \end{aligned}$$

From this expression we see that QS is upper triangular.
QED

8.6 Find the steady-state solution of the differential Riccati equation for a scalar system. Show from your solution how the steady-state solution changes with A, C, Q, and R, and give intuitive explanations.

Solution:

The differential Riccati equation for a scalar system is

$$\dot{P} = \frac{-P^2C^2}{R} + 2AP + Q$$

Setting $\dot{P} = 0$ and solving the resultant quadratic equation for P gives

$$P = \frac{AR + \sqrt{A^2R^2 + QRC^2}}{C^2}$$

From this equation we see that P increases as A increases, because a larger A means a more dynamic state, which is more difficult to estimate. We see that P decreases as C increases, because a larger C means greater observability. We see that P increases as Q and R increase, because larger process and measurement noise makes the state more difficult to estimate.

8.7 Consider the system of Example 8.3 except with process noise that has a covariance of diag$(0, q)$. Find an analytical expression for the steady-state estimation-error covariance.

Solution:

The steady-state CARE is given as

$$0 = AP + PA^T + Q - PC^TR^{-1}CP$$

Making substitutions and equating the elements on the left and right sides of the steady-state CARE gives

$$\begin{aligned} 0 &= 2P_{12} - P_{11}^2/R \\ 0 &= P_{22} - P_{11}P_{12}/R \\ 0 &= q - P_{12}^2/R \end{aligned}$$

Solving for the elements of P gives

$$\begin{aligned} P_{11} &= \sqrt{2R^{3/2}q^{1/2}} \\ P_{12} &= \sqrt{Rq} \\ P_{22} &= \sqrt{2R^{1/2}q^{3/2}} \end{aligned}$$

8.8 Show that if $a_1 \neq a_2$ and $q_{12} \neq 0$ in the system of Example 8.5, then (A, C) is detectable and (A, G) is stabilizable for all matrices G such that $GG^T = Q$.

Solution:

The observability matrix of (A, C) can be written as

$$\begin{aligned} \mathcal{O} &= \begin{bmatrix} C \\ CA \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ a_1 & 0 \\ 0 & a_2 \end{bmatrix} \end{aligned}$$

The rank of $\mathcal{O}$ is equal to two, so the system is observable regardless of the value of a_1, a_2, or q_{12}. Detectability is a subset of observability. The controllability matrix of (A, G) can be written as

$$\mathcal{C} = \begin{bmatrix} G & AG \end{bmatrix}$$

Recall that for general matrices X and Y we have $\rho(XY) \leq \min[\rho(X), \rho(Y)]$, so if the rank of $\mathcal{C}G^T$ is two, then the rank of $\mathcal{C}$ is two. $\mathcal{C}G^T$ can be written as

$$\begin{aligned} \mathcal{C}G^T &= \begin{bmatrix} G & AG \end{bmatrix} G^T \\ &= \begin{bmatrix} GG^T & AGG^T \end{bmatrix} \\ &= \begin{bmatrix} Q & AQ \end{bmatrix} \\ &= \begin{bmatrix} q_{11} & q_{12} & a_1 q_{11} & a_1 q_{12} \\ q_{12} & q_{22} & a_2 q_{12} & a_2 q_{22} \end{bmatrix} \end{aligned}$$

The rank of the above matrix is two if $a_1 \neq a_2$ and $q_{12} \neq 0$, so the rank of $\mathcal{C}$ must be equal to two, so the system is controllable. Stabilizability is a subset of controllability. Note that if $a_1 = a_2$, then the rank of $\mathcal{C}G^T$ might be one (for example, if $q_{11} = q_{12} = q_{22}$). Also note that if $q_{12} = 0$, then the rank of $\mathcal{C}G^T$ might be one (for example, if $q_{11} = 0$).
QED

8.9 Show that if $a_1 = a_2 < 0$, $q_{12} \neq 0$, and $|Q| = 0$ in the system of Example 8.5, then (A, C) is detectable and (A, G) is stabilizable for all matrices G such that $GG^T = Q$.

Solution:

The detectability proof follows the same procedure as in Problem 8.8; detectability is independent of the values of a_1, a_2, q_{12}, or $|Q|$. If $a_1 = a_2 < 0$, then the system is

stable. Stability is a subset of stabilizability. As with detectability, stabilizability is independent of the values of q_{12} or $|Q|$.
QED

8.10 Show that if $a_1 = a_2 > 0$, $q_{12} \neq 0$, and $|Q| = 0$ in the system of Example 8.5, then (A, C) is detectable and (A, G) is controllable on the imaginary axis, but (A, G) is not stabilizable for all matrices G such that $GG^T = Q$.

Solution:

The detectability proof follows the same procedure as in Problem 8.8; detectability is independent of the values of a_1, a_2, q_{12}, or $|Q|$. If $a_1 = a_2 > 0$ then (A, G) is controllable on the imaginary axis because the poles of the system will be in the right half plane (off the imaginary axis). However, the system will be unstable, so it will not be stabilizable unless it is controllable. In order to examine controllability, we write the matrix CG^T as shown in Problem 8.8, but since $|Q| = 0$ we can substitute $q_{12} = \sqrt{q_{11}q_{22}}$.

$$CG^T = \begin{bmatrix} q_{11} & \sqrt{q_{11}q_{22}} & a_1 q_{11} & a_1\sqrt{q_{11}q_{22}} \\ \sqrt{q_{11}q_{22}} & q_{22} & a_2\sqrt{q_{11}q_{22}} & a_2 q_{22} \end{bmatrix}$$

In order for (A, G) to be controllable, the above matrix must be nonsingular. That means that $(CG^T)(CG^T)^T$ must have a nonzero determinant. A few lines of algebra shows that this determinant is equal to $(a_1 - a_2)^2$, which is equal to zero for $a_1 = a_2$. Therefore the matrix pair (A, G) is uncontrollable if $a_1 = a_2 > 0$ and $|Q| = 0$.
QED

Computer exercises

8.11 Consider the discrete-time system $x_{k+1} = x_k + w_k$ with the initial condition $x_0 = 0$. The sample time is T and the variance of the zero-mean process noise w_k is equal to $2T$. Simulate the system a few thousand times for 10 s with: (a) $T = 0.5$ s; (b) $T = 0.4$ s; (c) $T = 0.2$ s. Use the value of x_k at $t = 10$ s to obtain a statistical estimate of $P(10) = E[x^2(10)]$.

a) What is your estimate of $P(10)$ for the three sample times given?

b) What is the analytically derived value for $P(10)$?

Solution

a). A few thousand simulations give estimated values of $P(10)$ for $T = 0.5$, $T = 0.4$, and $T = 0.2$, that are equal to 20.5, 20.5, and 20.0 respectively. Your results may vary slightly since these numbers are statistically determined estimates.

b). $E(x_m^2) = mQ$, where Q is the variance of the process noise. When $T = 0.5$ the time will be 10 s at $m = 20$, and $Q = 2T = 1$. Therefore $E(x_m^2) = 20$.

8.12 Consider the continuous-time scalar system

$$\dot{x} = -x + w$$

$$y = x + v$$

where $w(t)$ and $v(t)$ are continuous-time white noise with variances $Q_c = 2$ and $R_c = 1$ respectively. Design a continuous-time Kalman filter to estimate x.

a) What is the theoretical steady-state variance of the estimation error?

b) Simulate the system for 1000 s with discretization step sizes of 0.4, 0.2, and 0.1 s. What are the resulting experimental estimation-error variances?

Solution

The steady-state estimation-error variance is computed as follows.

a).

$$\begin{aligned} \dot{P} &= -PC^T R_c^{-1} CP + AP + PA^T + Q_c \\ 0 &= -P^2 - 2P + 2 \\ P &= \sqrt{3} - 1 \\ &\approx 0.73 \end{aligned}$$

b). For $\Delta t = 0.4$, 0.2, and 0.1, I get experimental estimation-error variances of 0.82, 0.80, and 0.80 respectively. Your results may vary slightly since these numbers are numerically determined estimates.

8.13 Simulate the system of Problem 8.7 for 10 seconds with $q = 2$ and $R = 3$. Plot the elements of the estimation-error covariance matrix as a function of time. Compare the experimental RMS estimation errors when using a time-varying Kalman gain and a constant Kalman gain.

Solution:

Figure 8.1 shows the estimation-error covariance as a function of time. The performance of the Kalman filter seems to be about the same whether a constant Kalman gain is used or a time-varying Kalman gain is used.

8.14 Repeat Problem 8.13 using the correlated noise filter when the process noise that affects the second state is equal to the measurement noise. How much do the estimation-error variances decrease due to the correlation between the two noise terms?

Solution:

Figure 8.2 shows the estimation-error covariance as a function of time. The performance of the Kalman filter seems to be about the same whether a constant Kalman gain is used or a time-varying Kalman gain is used. The steady-state estimation-error variances were about 3.83 and 3.13 for the filter without correlated noise (Problem 8.13), and they are about 1.95 and 2.41 for the filter with correlated noise (this problem), so they decreased by about 49% and 23% respectively.

8.15 Consider the system of Example 8.5 with $R = I$.

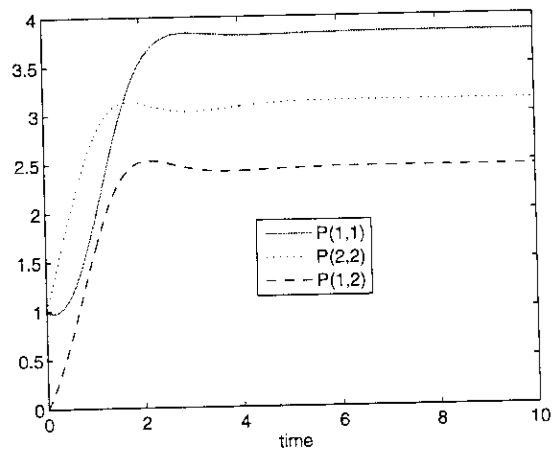

Figure 8.1 Solution to Problem 8.13

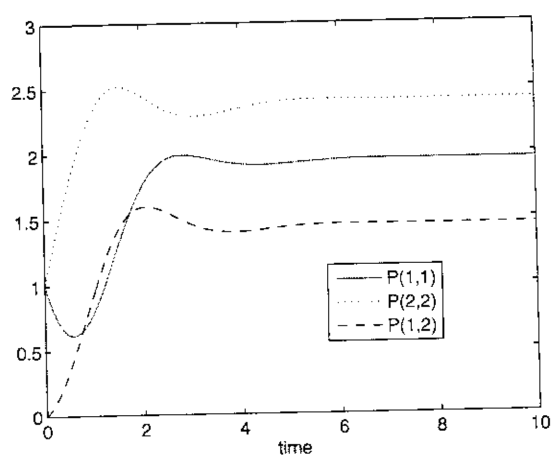

Figure 8.2 Solution to Problem 8.14

a) Integrate the Riccati equation with $a_1 = 1$, $a_2 = 2$, $q_{11} = q_{12} = q_{22} = 1$, and $P(0) = I$. Plot the Riccati equation solution as a function of time and verify that its steady-state value matches the results of Equation (8.139) and MATLAB's CARE function.

b) Integrate the Riccati equation with $a_1 = a_2 = -1$, $q_{11} = 1$, $q_{12} = 2$, $q_{22} = 4$, and $P(0) = I$. Plot the Riccati equation solution as a function of time and verify that its steady-state value matches the results of Equation (8.140) and MATLAB's CARE function.

c) Integrate the Riccati equation with $a_1 = a_2 = 1$, $q_{11} = 1$, $q_{12} = 2$, $q_{22} = 4$, and $P(0) = I$. Plot the Riccati equation solution as a function of time and verify that its steady-state value matches the results of Equation (8.139) and MATLAB's CARE function.

d) Integrate the Riccati equation with $a_1 = a_2 = 1$, $q_{11} = 1$, $q_{12} = 2$, $q_{22} = 4$, and $P(0) = 0$. [Note that this is the same as part (c) except for $P(0)$.] Plot the Riccati equation solution as a function of time and verify that its steady-state value matches the results of Equation (8.140). Does it match the results of MATLAB's CARE function? Does it result in a stable steady-state Kalman filter?

Solution:

Figure 8.3 shows the results of the Riccati equation integrations. For part (a) the steady-state solution from Figure 8.3 is

$$P = \begin{bmatrix} 2.3868 & 0.2774 \\ 0.2774 & 4.2188 \end{bmatrix}$$

which matches MATLAB's CARE result and Equation (8.139). For part (b) the steady-state solution from Figure 8.3 is

$$P = \begin{bmatrix} 0.2899 & 0.5798 \\ 0.5798 & 1.1596 \end{bmatrix}$$

which matches MATLAB's CARE result and Equation (8.140). For part (c) the steady-state solution from Figure 8.3 is

$$P = \begin{bmatrix} 2.2899 & 0.5798 \\ 0.5798 & 3.1596 \end{bmatrix}$$

which matches MATLAB's CARE result and Equation (8.139). For part (c) the steady-state solution from Figure 8.3 is

$$P = \begin{bmatrix} 0.6899 & 1.3798 \\ 1.3798 & 2.7596 \end{bmatrix}$$

which matches Equation (8.140). It does not match MATLAB's CARE result. The resulting eigenvalues of $(A - KC)$ are -2.45 and 1, so it does not result in a stable steady-state filter.

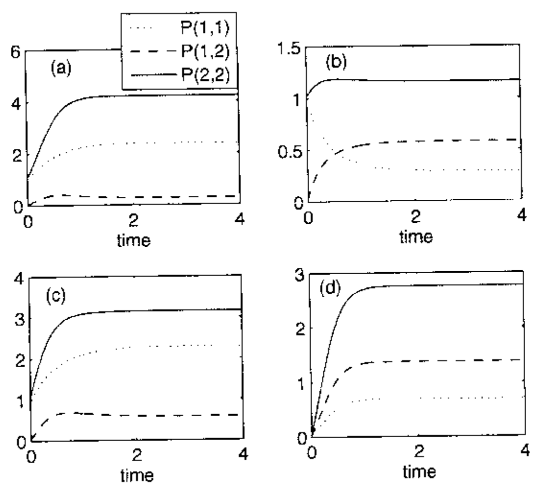

Figure 8.3 Solution to Problem 8.15

CHAPTER 9

Optimal smoothing

Problems

Written exercises

9.1 Prove or disprove the following conjecture: The trace of the inverse of a matrix is equal to the inverse of the trace of the matrix.

Solution:

Consider the matrix $A = \text{diag}(2, 2)$. The trace of its inverse is equal to 1, and the inverse of its trace is equal to $1/4$. This proves that the conjecture is false.

9.2 Show that $(A+B)^{-1} = B^{-1}(AB^{-1}+I)^{-1}$.

Solution:

Multiplying $(A+B)$ by its supposed inverse gives

$$\begin{aligned}(A+B)\left[B^{-1}(AB^{-1}+I)^{-1}\right] &= (AB^{-1}+I)(AB^{-1}+I)^{-1} \\ &= I\end{aligned}$$

This proves that the inverse of $(A + B)$ is equal to $B^{-1}(AB^{-1} + I)^{-1}$.
QED

9.3 Derive Equation (9.83).

Solution:

Apply the matrix inversion lemma of Equation (1.39) to the first line of Equation (9.83) with the substitutions $A = (P_{fm}^{+})^{-1}$, $B = I$, $C = I$, and $D = P_{bm}^{-}$. Equation (9.83) immediately follows.

9.4 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$.

a) What is the steady-state value of the *a priori* estimation-error covariance P_k^-?

b) Suppose that after the Kalman filter has reached steady state, the fixed-point smoother begins to operate. Find a closed-form solution to the covariance of the smoothed estimate Π_k as a function of the time index k. What is the limiting value of Π_k as $k \to \infty$?

Solution:

a). From Equation (9.25) we obtain the steady-state value of the *a priori* estimation-error covariance as

$$\begin{aligned}
P &= P(1-L) + Q \\
&= \frac{PR}{P+R} + Q \\
P &= \frac{1}{2}\left(Q \pm \sqrt{Q^2 + 4QR}\right) \\
&= 2Q \qquad \text{for } R = 2Q
\end{aligned}$$

b). From Equation (9.25) we obtain the covariance of the smoothed estimate for $k \geq j$ as

$$\begin{aligned}
\Sigma_k &= Q2^{1+j-k} \\
\Pi_{k+1} &= \Pi_k - \Sigma_k \lambda_k \\
&= \Pi_k - \left(Q2^{1+j-k}\right)\left(\frac{Q2^{1+j-k}}{4Q}\right) \\
&= \Pi_k - Q4^{j-k} \\
\Pi_k &= Q\left(2 - \sum_{m=0}^{k-j+1} 4^{-m}\right)
\end{aligned}$$

The summation on the right side of the above equation converges to 4/3, so the steady-state value of the covariance of the smoothed estimate is given as

$$\lim_{k \to \infty} \Pi_k = 2Q/3$$

9.5 Repeat Problem 9.4 for the case $R = 12Q$. What is the percent improvement in the estimation-error covariance due to smoothing? Explain why the percent improvement due to smoothing for this case differs in the way that it does from the results of Problem 9.4.

Solution:

a). From Equation (9.25) we obtain the steady-state value of the *a priori* estimation-error covariance as

$$\begin{aligned} P &= P(1-L)+Q \\ &= \frac{PR}{P+R}+Q \\ P &= \frac{1}{2}\left(Q \pm \sqrt{Q^2+4QR}\right) \\ &= 4Q \qquad \text{for } R = 12Q \end{aligned}$$

b). From Equation (9.25) we obtain the covariance of the smoothed estimate for $k \geq j$ as

$$\begin{aligned} \Sigma_k &= 4Q(3/4)^{k-j} \\ \Pi_{k+1} &= \Pi_k - \Sigma_k \lambda_k \\ &= \Pi_k - \left(4Q(3/4)^{k-j}\right)\left(\frac{4Q(3/4)^{k-j}}{16Q}\right) \\ &= \Pi_k - Q(9/16)^{k-j} \\ \Pi_k &= Q\left(4 - \sum_{m=0}^{k-j-1} (9/16)^m\right) \end{aligned}$$

The summation on the right side of the above equation converges to 16/7, so the steady-state value of the covariance of the smoothed estimate is given as

$$\lim_{k\to\infty} \Pi_k = 12Q/7$$

c). The percent improvement due to smoothing for Problem 9.4 was $(2Q - 2Q/3)/2Q = 67\%$. The percent improvement for this problem is $(4Q - 12Q/7)/4Q = 57\%$. The percent improvement for this problem is smaller than for Problem 9.4 because the measurement noise is larger.

9.6 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$. Suppose that the fixed-lag smoother for this system is in steady state so that $P_{k+1}^- = P_k^-$, $L_{k+1,i} = L_{k,i}$, $P_{k+1}^{i,i} = P_k^{i,i}$, and $P_{k+1}^{0,i} = P_k^{0,i}$, for $i = 1, \cdots, N+1$. Find closed-form expressions for P_k^-, $L_{k,i}$, $P_k^{i,i}$, and $P_k^{0,i}$ as functions of i. What is the limit as $i \to \infty$ of $L_{k,i}$, $P_k^{i,i}$, and $P_k^{0,i}$?

Solution:

From Equation (9.25) we obtain the steady-state value of the *a priori* estimation-error covariance as

$$P_k^- = P_k^-(1-L_k)+Q$$

$$\begin{aligned}
&= \frac{P_k^- R}{P_k^- + R} + Q \\
P_k^- &= \frac{1}{2}\left(Q \pm \sqrt{Q^2 + 4QR}\right) \\
&= 2Q \qquad \text{for } R = 2Q
\end{aligned}$$

From the fixed-lag smoother equations on page 278 we obtain a closed-form expression for $P_k^{0,i}$ as follows.

$$\begin{aligned}
P_k^{0,i} &= P_k^{0,i-1}(1 - L_{k,0}) \\
&= \frac{1}{2} P_k^{0,i-1} \\
P_k^{0,i} &= Q 2^{1-i} \qquad i = 1, \cdots, N+1
\end{aligned}$$

We obtain a closed-form expression for $L_{k,i}$ as follows.

$$\begin{aligned}
L_{k,i} &= P_k^{0,i-1}(P_k^{0,0} + R)^{-1} \\
&= 2^{-i} \qquad i = 1, \cdots, N+1
\end{aligned}$$

We obtain a closed-form expression for $P_k^{i,i}$ as follows.

$$\begin{aligned}
P_k^{i,i} &= P_k^{i-1,i-1} - P_k^{0,i-1} L_{k,i}^T \\
&= P_k^{i-1,i-1} + (Q 2^{2-i})(2^{-i}) \\
P_k^{i,i} &= Q\left(2 - \sum_{j=0}^{i} 4^{-i}\right) \qquad i = 1, \cdots, N+1
\end{aligned}$$

From these expressions we find

$$\begin{aligned}
\lim_{i\to\infty} P_k^{0,i} &= 0 \\
\lim_{i\to\infty} L_{k,i} &= 0 \\
\lim_{i\to\infty} P_k^{i,i} &= 2Q/3
\end{aligned}$$

9.7 Suppose you have a fixed-lag smoother as shown in Equation (9.43) that is in steady state. How do the eigenvalues of the fixed-lag smoother relate to the eigenvalues of the standard Kalman filter? What do you conclude about the stability of the fixed-lag smoother?

Solution:

The system matrix of the fixed-lag smoother is given as

$$\begin{bmatrix}
F_k - L_{k,0}H_k & 0 & \cdots & 0 \\
I - L_{k,1}H_k & 0 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
-L_{k,N+1}H_k & \cdots & I & 0
\end{bmatrix}$$

Since this matrix is triangular, its eigenvalues are the eigenvalues of the diagonal blocks. The eigenvalues are equal to the eigenvalues of $(F_k - L_{k,0}H_k)$, plus an additional $n(N+1)$ zeros, where n is the number of system states. The smoother eigenvalues are equal to the eigenvalues of the standard Kalman filter for the system, plus the additional zeros. The fixed-lag smoother therefore inherits the stability properties of the standard Kalman filter for the system.

9.8 Solve Equation (9.10) for $(y_k - H_k\hat{x}_k^-)$ [assuming that $\rho(L_k) = r$, where r is the number of measurements in the system]. Substitute the resulting expression for $(y_k - H_k\hat{x}_k^-)$ in the fixed-lag smoother equation for $\hat{x}_{k+1-i,k}$ to show that the smoothed state estimate can be driven by the state estimates without any input from the measurements [And79].

Solution:

Equation (9.10) can be rearranged as

$$y_k - H_k\hat{x}_k^- = (L_k^T L_k)^{-1} L_k^T (\hat{x}_{k+1}^- - F_k\hat{x}_k^-)$$

Substituting this into the fixed-lag smoother equation for $\hat{x}_{k+1-i,k}$ gives

$$\hat{x}_{k+1-i,k} = \hat{x}_{k+2-i,k} + L_{k,i}(L_k^T L_k)^{-1} L_k^T (\hat{x}_{k+1}^- - F_k\hat{x}_k^-)$$

9.9 Suppose that $\hat{x}_f$ and $\hat{x}_b$ are unbiased estimates of x, and $\hat{x} = K_f\hat{x}_f + K_b\hat{x}_b$. Show that if $\hat{x}$ is an unbiased estimate of x, then we must have $K_f + K_b = I$.

Solution:

Suppose that $E(\hat{x}_f) = E(\hat{x}_b) = \bar{x}$. Then

$$\begin{aligned} E(\hat{x}) &= E(K_f\hat{x}_f + K_b\hat{x}_b) \\ &= K_f E(\hat{x}_f) + K_b E(\hat{x}_b) \\ &= (K_f + K_b)\bar{x} \end{aligned}$$

This shows that if we want $E(\hat{x}) = \bar{x}$ then we must have $K_f + K_b = I$.
QED

9.10 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$. Use the forward–backward smoother of Section 9.4.1 to find the steady-state value of the covariance of the smoothed state estimate.

Solution:

Using Equation (9.67) to solve for the steady-state value of the forward Kalman filter estimation-error covariance gives $P_{fk}^+ = Q$. Using Equation (9.76) to solve for the steady-state value of the backward Kalman filter estimation-error covariance gives $P_{bk}^- = 2Q$. Using Equation (9.78) to solve for the steady-state value of the smoothed estimation-error covariance gives

$$\begin{aligned} P_k &= \left[(P_{fk}^+)^{-1} + (P_{bk}^-)^{-1}\right]^{-1} \\ &= 2Q/3 \end{aligned}$$

9.11 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$. Use the RTS smoother of Section 9.4.2 to find the steady-state value of the covariance of the smoothed state estimate.

Solution:

Using Equation (9.136) to solve for the steady-state value of the forward Kalman filter estimation-error covariance gives $P_{fk}^- = 2Q$ and $P_{fk}^+ = Q$. We then use Equation (9.138) to solve for the steady-state value of the RTS smoother estimation-error covariance to obtain $P_k = 2Q/3$.

9.12 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$. Suppose that the forward filter has reached steady state. Use the RTS smoother of Section 9.4.2 to find the covariance of the smoothed state estimate for $k = N, N-1, N-2, N-3$, and $N-4$. At what point does the covariance of the smoothed state estimate get within 1% of its steady-state value?

Solution:

We can use Equation (9.136) to find the steady-state values $P_{fk}^+ = Q$, $P_{fk}^- = 2Q$, and $K_k = 1/2$. Next we use Equation (9.138) to find

$$
\begin{aligned}
P_N &= Q \\
P_{N-1} &= 3Q/4 = 0.75Q \\
P_{N-2} &= 11Q/16 = 0.6875Q \\
P_{N-3} &= 43Q/64 \approx 0.6719Q \\
P_{N-4} &= 171Q/256 \approx 0.6680Q
\end{aligned}
$$

The steady-state value of P_k can be found to be equal to $2Q/3 \approx 0.6667Q$ (see Problem 9.11). P_k reaches within 1% of $2Q/3$ at $k = N-3$.

9.13 Repeat Problem 9.12 for $R = 12Q$. How do you intuitively explain the quicker convergence of P_k to steady state?

Solution:

We can use Equation (9.136) to find the steady-state values $P_{fk}^+ = 3Q$, $P_{fk}^- = 4Q$, and $K_k = 1/4$. Next we use Equation (9.138) to find

$$
\begin{aligned}
P_N &= 3Q \\
P_{N-1} &= 47Q/16 \approx 2.9375Q \\
P_{N-2} &= 751Q/256 \approx 2.9336Q \\
P_{N-3} &= 12015Q/4096 \approx 2.9333Q \\
P_{N-4} &= 192239Q/65536 \approx 2.9333Q
\end{aligned}
$$

The steady-state value of P_k can be found to be equal to $44Q/15 \approx 2.9333Q$. P_k reaches within 1% of its steady-state value at $k = N-1$. This is quicker than Problem 9.12 because in this problem we have more measurement noise, and systems with more measurement noise are not as smoothable (see Section 9.2.1). If the system is not as smoothable that means that P_k will not improve as much from its initial value, which means that P_k will reach its steady-state value quicker.

9.14 Use the RTS smoother equations to show that constant states are not smoothable. That is, if $F = I$ and $Q = 0$, then $P_k = P_{fN}^+$ for all k.

Solution:

First note that if $F = I$ and $Q = 0$ then Equation (9.136) shows that $P_{fk}^- = P_{f,k-1}^+$. Next use Equation (9.138) to see that

$$\begin{aligned} P_k &= P_{fk}^+ - P_{fk}^+ \mathcal{I}_{f,k+1}^- (P_{f,k+1}^- - P_{k+1}) \mathcal{I}_{f,k+1}^- P_{fk}^+ \\ &= P_{fk}^+ - P_{f,k+1}^- \mathcal{I}_{f,k+1}^- (P_{f,k+1}^- - P_{k+1}) \mathcal{I}_{f,k+1}^- P_{f,k+1}^- \\ &= P_{fk}^+ - (P_{f,k+1}^- - P_{k+1}) \\ &= P_{k+1} \end{aligned}$$

Since $P_N = P_{fN}^+$, we see from the above that $P_k = P_{fN}^+$ for all k.
QED

Computer exercises

9.15 Consider the second-order system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t)$$

where $\omega = 6$ rad/s is the natural frequency of the system, and $\zeta = 0.16$ is the damping ratio. The input $w(t)$ is continuous-time white noise with a variance of 0.01. Measurements of the first state are taken every 0.5 s:

$$y(t_k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t_k) + v(t_k)$$

where $v(t_k)$ is discrete-time white noise with a variance of 10^{-4}. The initial state, estimate, and covariance are

$$\begin{aligned} x(0) &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ \hat{x}(0) &= x(0) \\ P(0) &= \begin{bmatrix} 10^{-5} & 0 \\ 0 & 10^{-2} \end{bmatrix} \end{aligned}$$

a) Discretize the system equation.
b) Implement the discrete-time Kalman filter and the RTS smoother for 10 s (20 time steps). Plot the variance of the estimation error of the first state for the forward filter and for the RTS smoother on a single plot. Do the same for the second state. Why is the second state more smoothable than the first state?

Solution:

a). Note that the continuous-time process noise covariance is

$$Q_c = \begin{bmatrix} 0 & 0 \\ 0 & 0.01 \end{bmatrix}$$

Discretization with a step size of Δt gives

$$\begin{aligned} F &= e^{A\Delta t} \\ Q &= Q_c \Delta t \end{aligned}$$

The discrete-time system is given as

$$\begin{aligned} x_{k+1} &= F x_k + w_k \\ w_k &\sim (0, Q) \end{aligned}$$

b). Figure 9.1 shows the forward and smoothed variances. The second state is more smoothable (i.e., there is more improvement from the forward variance to the smoothed variance) because the process noise affects the second state directly, but affects the first state only indirectly through integration of the first state. Since smoothability is proportional to the magnitude of the process noise, the second state is more smoothable than the first state.

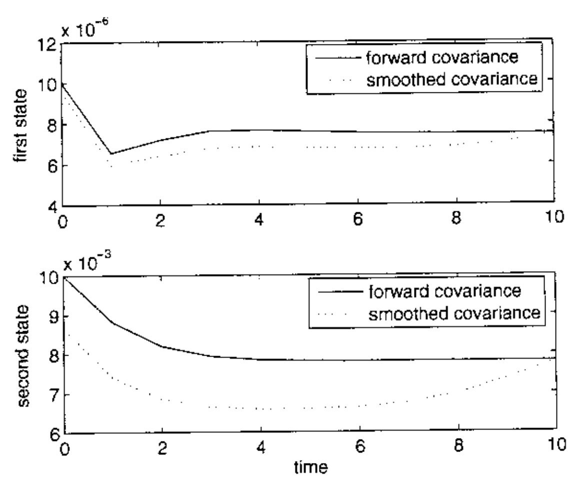

Figure 9.1 Solution to Problem 9.15. The process noise covariance is equal to 0.01

9.16 Repeat Problem 9.15 with the continuous-time process noise $w(t)$ having a variance of 1. How does this change the smoothability of the states?

Solution:

Figure 9.2 shows the forward and smoothed variances. Compared with Figure 9.1, it is seen that the states are much more smoothable with the higher process noise variance. Higher process noise variance means greater smoothability.

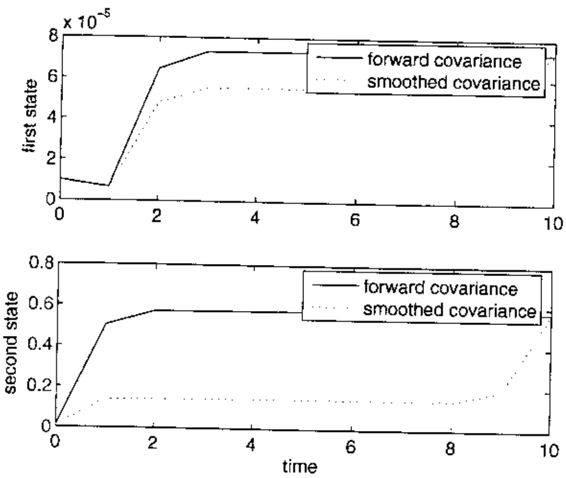

Figure 9.2 Solution to Problem 9.16. The process noise covariance is equal to 1

9.17 Design a fixed-interval smoother for the system described in Problem 5.11 to estimate the state at each time on the basis of measurements at all 10 time steps.

a) Plot the *a posteriori* covariance of the forward state estimate and the covariance of the smoothed state estimate as a function of time for both states.

b) What are the percent improvements in the estimation-error variances due to smoothing for the two states at the initial time? Why is there so much more improvement for one state than for the other state?

c) Simulate the system and smoother a hundred times or so, each simulation with a different noise history. On the basis of your simulations, derive a numerical estimate of the smoother estimation-error variances of the two states at the initial time. How do your numerical variances compare with the theoretical variances obtained in part (b)?

Solution:

a). Figure 9.3 shows the forward variance and the smoothed variance for the two states as functions of time.

b). The initial *a posteriori* variances of the two states are 500 and 200, specified in the problem statement. The smoothed variances of the two states at the final time are 167 and 8.1. The improvement in estimation accuracy due to smoothing is 67% for the first state and 96% for the second state. The reason for the difference is that the process noise covariance is $\text{diag}(0, 10)$, so the process noise has a more direct effect on the second state. Recalling from Section 9.2.2 that constant states are not smoothable, we expect less process noise to result in less smoothability, and that is exactly what we see in this problem.

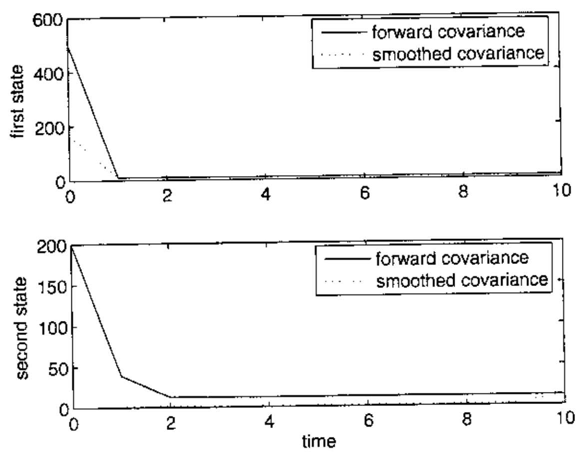

Figure 9.3 Solution to Problem 9.17(a)

c). The variance of the estimation error of the smoother at the initial time, based on 100 simulations, is 117 for the first state and 6.4 for the second state. These numbers are reasonably close to the theoretical variances of 167 and 8.1 that were derived in part (b).

CHAPTER 10

Additional topics in Kalman filtering

Problems

Written exercises

10.1 In this problem we consider the scalar system

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \end{aligned}$$

where w_k and v_k are white and uncorrelated with respective variances Q and R, which are unknown. A suboptimal steady-state value of K is used in the state estimator since Q and R are unknown.

a) Use the expression for P_k^- along with the first expression for P_k^+ in Equation (5.19) to find the steady-state value of P_k^- as a function of the suboptimal value of K and the true values of Q and R. [Note that the first expression for P_k^+ in Equation (5.19) does not depend on the value for K_k being optimal.]

b) Now suppose that $E(r_k^2)$ and $E(r_{k+1}r_k)$ are found numerically as the filter runs. Find the true value of R and the steady-state value of P_k^- as a function of $E(r_k^2)$ and $E(r_{k+1}r_k)$.

c) Use your results from parts (a) and (b) to find the true value of Q.

Solution:

a). From Equation (5.19) we obtain the steady-state value of P_k^- as

$$P^- = \frac{RK^2 + Q}{2K - K^2}$$

K in this equation is the gain that is used in the filter (based on the wrong Q and R), but Q and R in this equation are the true variances of the process and measurement noise.

b). From Section 10.1 we have

$$\begin{aligned} E(r_k^2) &= P^- + R \\ E(r_{k+1}r_k) &= (1-K)P^- + R \end{aligned}$$

Solving these equations for P^- and R gives

$$\begin{aligned} P^- &= \frac{E(r_k^2) - E(r_{k+1}r_k)}{K} \\ R &= \frac{(K-1)E(r_k^2) + E(r_{k+1}r_k)}{K} \end{aligned}$$

This gives the true values of P^- and R as a function of the suboptimal gain K and the numerically determined quantities $E(r_k^2)$ and $E(r_{k+1}r_k)$.

c). Solving part (a) for Q gives

$$Q = (2K - K^2)P^- - RK^2$$

where K is the suboptimal filter gain that is used in the state estimator, and P^- and R were solved in part (b).

10.2 Show that the innovations $r = y - C\hat{x}$ of the continuous-time Kalman filter is white with covariance R.

Solution:

This solution is taken from [Gel74]. The system and filter equations are given as

$$\begin{aligned} \dot{x} &= Ax + w, \qquad w \sim (0, Q) \\ y &= Cx + v, \qquad v \sim (0, R) \\ \dot{\hat{x}} &= A\hat{x} + Kr \\ r &= y - C\hat{x} \end{aligned}$$

If we define the estimation error as $\tilde{x} = x - \hat{x}$, the innovations and its covariance can be written as

$$\begin{aligned} r &= -C\tilde{x} + v \\ E[r(t_2)r^T(t_1)] &= C(t_2)E[\tilde{x}(t_2)\tilde{x}^T(t_1)]C^T(t_1) - \\ & \quad C(t_2)E[\tilde{x}(t_2)v^T(t_1)] + R(t_1)\delta(t_2 - t_1) \end{aligned}$$

We see that $\tilde{x}$ satisfies the equation

$$\dot{\tilde{x}} = (A - KC)\tilde{x} - w + Kv$$

which has the solution

$$\tilde{x}(t_2) = \phi(t_2, t_1)\tilde{x}(t_1) - \int_{t_1}^{t_2} \phi(t_2, \tau)[w(\tau) - K(\tau)v(\tau)]\, d\tau$$

where $\phi(t_2, t_1)$ is the state transition matrix of the $\dot{\tilde{x}}$ equation. From this equation we compute

$$\begin{aligned} E[\tilde{x}(t_2)\tilde{x}^T(t_1)] &= \phi(t_2, t_1)P(t_1) \\ E[\tilde{x}(t_2)v^T(t_1)] &= \phi(t_2, t_1)K(t_1)R(t_1) \end{aligned}$$

Combining these results gives

$$E[r(t_2)r^T(t_1)] = C(t_2)\phi(t_2, t_1)[P(t_1)C^T(t_1) - K(t_1)R(t_1)] + R(t_1)\delta(t_2 - t_1)$$

Realizing that $K(t_1) = P(t_1)C^T(t_1)R^{-1}(t_1)$ results in

$$E[r(t_2)r(t_1)] = R(t_1)\delta(t_2 - t_1)$$

QED

10.3 Consider the system described in Problem 5.1. Find the steady-state variance of the Kalman filter innovations when $Q = R$ and when $Q = 2R$.

Solution:

The system and Kalman filter covariance equations are given as

$$\begin{aligned} x_k &= \frac{1}{2}x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \\ P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ P_k^+ &= (I - K_k H_k)P_k^- \end{aligned}$$

From these equations we find the steady-state values

$$\begin{aligned} P^- &= \frac{1}{8}\left[4Q - 3R + \sqrt{(4Q + 3R)^2 + 16QR}\right] \\ E(r^2) &= HP^-H^T + R \\ &= \frac{1}{8}\left[4Q + 5R + \sqrt{(4Q + 3R)^2 + 16QR}\right] \\ &= \begin{cases} (9R + R\sqrt{65})/8 \approx 2.13R & (Q = R) \\ (13R + R\sqrt{153})/8 \approx 3.17R & (Q = 2R) \end{cases} \end{aligned}$$

10.4 Consider the system of Problem 10.3 with $Q = R = 1$. Suppose the Kalman filter for the system has reached steady state. At time k the innovations $r_k = y_k - \hat{x}_k^-$.

a) Find an approximate value for $\mathrm{pdf}(y_k|p)$ (where p is the model used in the Kalman filter) if $r_k = 0$, if $r_k = 1$, and if $r_k = 2$.

b) Suppose that the use of model p_1 gives $r_k = 0$, model p_2 gives $r_k = 1$, and model p_3 gives $r_k = 2$. Further suppose that $\Pr(p_1|y_{k-1}) = 1/4$, $\Pr(p_2|y_{k-1}) = 1/4$, and $\Pr(p_3|y_{k-1}) = 1/2$. Find $\Pr(p_j|y_k)$ for $j = 1, 2, 3$.

Solution:

a). From Problem 10.3 we compute $S = 2.13$. From Equation (10.25) we compute

$$\begin{aligned}
\mathrm{pdf}(y_k|p) &\approx \frac{\exp(-r_k^T S_k^{-1} r_k/2)}{(2\pi)^{q/2}|S_k|^{1/2}} \\
&= \frac{\exp(-r_k^2/2S)}{\sqrt{2\pi S}} \\
&= \begin{cases} 0.27 & (r_k = 0) \\ 0.22 & (r_k = 1) \\ 0.11 & (r_k = 2) \end{cases}
\end{aligned}$$

b). From Equation (10.29) we obtain

$$\begin{aligned}
\Pr(p_j|y_k) &= \frac{\mathrm{pdf}(y_k|p_j)\Pr(p_j|y_{k-1})}{\sum_{i=1}^{N}\mathrm{pdf}(y_k|p_i)\Pr(p_i|y_{k-1})} \\
&= \begin{cases} 0.38 & (j = 1) \\ 0.31 & (j = 2) \\ 0.31 & (j = 3) \end{cases}
\end{aligned}$$

10.5 Consider the system described in Example 4.1 where the measurement consists of the predator population. Suppose that we want to estimate $x(1) + x(2)$, the sum of the predator and prey populations. Create an equivalent system with transformed states such that our goal is to estimate the first element of the transformed state vector.

Solution:

Since we want to estimate $x(1) + x(2)$ we set $T_1^T = \begin{bmatrix} 1 & 1 \end{bmatrix}$. We then pick an arbitrary S so that the matrix T in Equation (10.39) is nonsingular. For example, we could have

$$T = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

We then perform the algebra shown in Equation (10.41) to obtain

$$\begin{aligned}
x_{k+1} &= \begin{bmatrix} -0.2 & 1.6 \\ 0.4 & 1.4 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} w_k \\
y_k &= \begin{bmatrix} 1 & -1 \end{bmatrix} x_k + v_k
\end{aligned}$$

The first state of this new system is equal to the sum of the states of the original system.

10.6 Consider the system

$$
\begin{aligned}
x_{k+1} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k \\
y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k
\end{aligned}
$$

where w_k and v_k are uncorrelated zero-mean white noise processes with variances q and R, respectively.

a) Use Anderson's approach to reduced-order filtering to estimate the first element of the state vector. Find steady-state values for $\tilde{P}$, $\tilde{\tilde{P}}$, Σ, $\tilde{\Pi}$, $\tilde{\tilde{\Pi}}$, and P. Find the steady-state gain K of the reduced-order filter.

b) Use the full-order filter to estimate the entire state vector. Find steady-state values for P and K.

c) Comment on the comparison between your answer for P in part (a) and part (b).

Solution:

a). Use Equation (10.47) to find

$$
\begin{aligned}
\tilde{P} &= q \\
\tilde{\tilde{P}} &= q \\
\Sigma &= 0 \\
\tilde{\Pi} &= Kq \\
&= q^2/(q+R) \\
\tilde{\tilde{\Pi}} &= 0 \\
P &= q(1-K)^2 + RK^2 \\
&= qR/(q+R) \\
K &= q/(q+R)
\end{aligned}
$$

b). Use Equation (5.19) to find

$$
\begin{aligned}
P &= \begin{bmatrix} qR/(q+R) & 0 \\ 0 & q \end{bmatrix} \\
K &= \begin{bmatrix} q/(q+R) \\ 0 \end{bmatrix}
\end{aligned}
$$

c). In this simple example the full-order filter estimates only the first element of the state vector (since the second element is unobservable). Therefore the portion of the full-order estimator that estimates the first state is equivalent to the reduced-order estimator. That is why P in part (b) is equal to P_{11} in part (a).

10.7 Consider the reduced-order filter of Example 10.3 with the initial condition $\tilde{\tilde{P}}_0^+ = 1$.

a) Find analytical expressions for the steady-state values of $\tilde{K}$, α, $\tilde{P}^+$, Σ^+, $\tilde{\tilde{P}}^+$, $\tilde{P}^-$, Σ^-, and $\tilde{\tilde{P}}^-$.

b) What does the reduced-order filter indicate for the steady-state *a posteriori* estimation-error variance of the first state? Find an analytical expression for the true steady-state *a posteriori* estimation-error variance of the first state when the reduced-order filter is used. Your answer should be a function of $x(2)$. Solve for the true steady-state *a posteriori* estimation-error variance of the first state when $x(2) = 0$, when $x(2) = 1$, and when $x(2) = 2$.

c) What is the steady-state *a posteriori* estimation-error variance of the first state when the full-order filter is used?

Solution:

a). Use Equation (10.67). First we easily solve $\tilde{\tilde{P}}^- = \tilde{\tilde{P}}^+ = 1$. Next we take the Σ^- expression and substitute for Σ^+, $\tilde{K}$, and α to obtain

$$\Sigma^- = \frac{-(\tilde{P}^- + \Sigma^-)(1 + \Sigma^-)}{\tilde{P}^- + 2\Sigma^- + 1 + R} + \Sigma^-$$

This means that either $\tilde{P}^- = -\Sigma^-$ or $\Sigma^- = -1$. Next we take the $\tilde{P}^-$ expression and substitute for $\tilde{K}$. Solving for $\tilde{P}^-$ gives

$$\tilde{P}^- = \frac{1}{2}\left[1 - 2\Sigma^- + \sqrt{4\Sigma^- + 5 + 4R}\right]$$

This shows that $\tilde{P}^- \neq -\Sigma^-$. That means that $\Sigma^- = -1$. We can then solve for the remaining quantities as

$$\begin{aligned}
\tilde{P}^- &= \frac{3+\sqrt{5}}{2} \\
\tilde{P}^+ &= \frac{1+\sqrt{5}}{2} \\
\alpha &= \frac{3+\sqrt{5}}{2} \\
\tilde{K} &= \frac{1+\sqrt{5}}{3+\sqrt{5}} \\
\Sigma^+ &= -1
\end{aligned}$$

b). The reduced-order filter indicates that the estimation-error variance of the first state is

$$\begin{aligned}
\tilde{P}^+ &= \frac{1+\sqrt{5}}{2} \\
&\approx 1.62
\end{aligned}$$

The true estimation-error variance of the first state can be found as follows.

$$\begin{aligned}
\hat{\tilde{x}}_{k+1}^+ &= \hat{\tilde{x}}_k^+ + K(y_{k+1} - \hat{\tilde{x}}_k^+) \\
&= \hat{\tilde{x}}_k^+ + K(x_{k+1}(1) + x_{k+1}(2) + v_{k+1} - \hat{\tilde{x}}_k^+)
\end{aligned}$$

Subtract both sides from $x_{k+1}(1)$ to obtain

$$\begin{aligned} e_{k+1} &= x_{k+1}(1) - \hat{\bar{x}}_{k+1}^{+} \\ &= (1-K)e_k + (1-K)w_k(1) - K(x_k(2) + v_{k+1}) \end{aligned}$$

Square both sides and take the expected value to obtain

$$E(e_{k+1}^2) = (1-K)^2 E(e_k^2) + (1-K)^2 Q(1,1) + K^2 x_k^2(2) + K^2 R$$

In steady state $E(e_{k+1}^2) = E(e_k^2)$, and we solve the above equation as

$$E(e^2) = \frac{(1-K)^2 Q(1,1) + K^2 x_k^2(2) + K^2 R}{1-(1-K)^2}$$

We have $Q(1,1) = R = 1$. The steady-state estimation-error variance is a function of the constant $x(2)$.

$$E(e^2) = \begin{cases} 0.62 & (x(2)=0) \\ 1.07 & (x(2)=1) \\ 2.41 & (x(2)=2) \end{cases}$$

c). From Equation (5.19) we solve for the steady-state *a posteriori* estimation-error covariance P as follows.

$$\begin{aligned} P &= P^- - KHP^- \\ &= (FPF^T + Q) - (PH^T R^{-1})H(FPF^T + Q) \\ P_{11} &= P_{11} + 1 - (P_{11} + P_{12})(P_{11} + P_{12} + 1) \\ P_{12} &= P_{12} - (P_{12} + P_{22})(P_{11} + P_{12} + 1) \\ P_{22} &= P_{22} - (P_{12} + P_{22})^2 \end{aligned}$$

From these equations we see that

$$\begin{aligned} P_{12} &= -P_{22} \\ P_{11} &= \frac{1}{2}\left(-2P_{12} - 1 + \sqrt{5}\right) \end{aligned}$$

The exact value of the steady-state covariance depends on initial conditions. Going back to Theorem 23 we see that our problem does not have a unique positive semidefinite DARE solution because (F, G) is not stabilizable for all G such that $GG^T = Q$.

10.8 Verify that the two expressions in Equation (10.98) are respectively equal to the cross-covariance of x and y, and the covariance of y.

Solution:

The first expression can be derived as follows.

$$\begin{aligned} P_{xy} &= E[(x-\bar{x})(y-\bar{y})^T] \\ &= E[(x-\hat{x}^-)(Hx + v - H\hat{x}^-)^T] \\ &= E[(x-\hat{x}^-)(x-\hat{x}^-)^T]H^T + E[(x-\hat{x}^-)v^T] \\ &= P^-H^T + 0 \end{aligned}$$

where all quantities are defined at time k. The second expression can be derived as follows.

$$\begin{aligned}
P_y &= E[(y-\bar{y})(y-\bar{y})^T] \\
&= E[(Hx+v-H\hat{x}^-)(Hx+v-H\hat{x}^-)^T] \\
&= HE[(x-\hat{x}^-)(x-\hat{x}^-)^T]H^T + HE[(x-\hat{x}^-)v^T] + E[v(x-\hat{x}^-)^T]H^T + E(vv^T) \\
&= HP^-H^T + 0 + 0 + R
\end{aligned}$$

10.9 Suppose you have the linear system $x_{k+1} = Fx_k + w_k$, where $w_k \sim (0, Q_k)$ is zero-mean white noise. Define $w(k+2,k)$ as the cumulative effect of all of the process noise on the state from time k to time $(k+2)$. What are the mean and covariance of $w(k+2,k)$?

Solution:

Note that

$$\begin{aligned}
x_{k+2} &= Fx_{k+1} + w_{k+1} \\
&= F^2x_k + Fw_k + w_{k+1} \\
&= F^2x_k + w(k+2,k)
\end{aligned}$$

where $w(k+2,k)$ is the cumulative effect of all of the process noise on the state from time k to time $(k+2)$. Its mean and covariance are calculated as follows.

$$\begin{aligned}
E[w(k+2,k)] &= E[Fw_k + w_{k+1}] \\
&= 0 \\
E[w(k+2,k)w^T(k+2,k)] &= E[(Fw_k + w_{k+1})(Fw_k + w_{k+1})^T] \\
&= FQ_kF^T + Q_{k+1}
\end{aligned}$$

10.10 Suppose that a Kalman filter is running with

$$\begin{aligned}
F &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\
H &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\
Q &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\
R &= 1 \\
P^+(k) &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}$$

An out-of-sequence measurement from time $(k-1)$ is received at the filter.

a) What was the value of $P^-(k)$?

b) Use the delayed-measurement filter to find the quantities $P_w(k,k-1)$, $P_{xw}(k,k-1)$, $P(k-1,k)$, $P_{xy}(k,k-1)$, and $P(k,k-1)$.

c) Realizing that the measurement at time $(k-1)$ was not received at time $(k-1)$, derive the value of $P^-(k-1)$. Now suppose that the measurement was received in the correct sequence at time $(k-1)$. Use the standard Kalman filter equations to compute $P^+(k-1)$, $P^-(k)$, and $P^+(k)$. How does your computed value of $P^+(k)$ compare with the value of $P(k,k-1)$ that you computed in part (b) of this problem?

Solution:

a). From Equation (5.19) we find

$$\begin{aligned} P^+(k) &= \left[(P^-(k))^{-1} + H^T R^{-1} H \right]^{-1} \\ P^-(k) &= \left[(P^+(k))^{-1} - H^T R^{-1} H \right]^{-1} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

b). $S(k)$ is found as

$$\begin{aligned} S(k) &= HP^-(k)H^T + R \\ &= 2 \end{aligned}$$

$Q(k,k-1)$ is equal to Q. $P_w(k,k-1)$ is found as

$$\begin{aligned} P_w(k,k-1) &= Q(k,k-1) - Q(k,k-1)H^T S^{-1}(k) H Q(k,k-1) \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

$P_{xw}(k,k-1)$ is found as

$$\begin{aligned} P_{xw}(k,k-1) &= Q(k,k-1) - P^-(k)H^T S^{-1}(k) H Q(k,k-1) \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

$F(k-1,k)$ is found as

$$\begin{aligned} F(k-1,k) &= F^{-1} \\ &= \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

$P(k-1,k)$ is found as

$$\begin{aligned} P(k-1,k) &= F(k-1,k) \left\{ P(k) - P_{xw}(k,k-1) - P_{xw}^T(k,k-1) + \right. \\ & \quad \left. P_w(k,k-1) \right\} F^T(k-1,k) \\ &= \begin{bmatrix} 1/2 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned}$$

$S(k-1)$ is found as

$$\begin{aligned} S(k-1) &= HP(k-1,k)H^T + R \\ &= 3/2 \end{aligned}$$

$P_{xy}(k, k-1)$ is found as

$$\begin{aligned} P_{xy}(k, k-1) &= [P(k) - P_{xw}(k, k-1)]F^T(k-1, k)H^T \\ &= \begin{bmatrix} 1/2 \\ 0 \end{bmatrix} \end{aligned}$$

$P(k, k-1)$ is found as

$$\begin{aligned} P(k, k-1) &= P(k) - P_{xy}(k, k-1)S^{-1}(k-1)P_{xy}^T(k, k-1) \\ &= \begin{bmatrix} 1/3 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

c). In part (b) we saw that $P^-(k) = I$. We can use Equation (5.19) to derive

$$\begin{aligned} P^+(k-1) &= F^{-1}\left[P^-(k) - Q\right]F^{-T} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned}$$

Realizing that we did not receive any measurements at time $(k-1)$, we know that $P^-(k-1) = P^+(k-1)$. Now suppose that we received a measurement at time $(k-1)$. From the standard Kalman filter formulas in Equation (5.19) we derive

$$\begin{aligned} K(k-1) &= P^-(k-1)H^T\left[HP^-(k-1)H^T + R\right]^{-1} \\ &= \begin{bmatrix} 1/2 \\ 0 \end{bmatrix} \\ P^+(k-1) &= [I - K(k-1)H]\,P^-(k-1) \\ &= \begin{bmatrix} 1/2 & 0 \\ 0 & 0 \end{bmatrix} \\ P^-(k) &= FP^+(k-1)F^T + Q \\ &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix} \\ P^+(k) &= \left(\left[P^-(k)\right]^{-1} + H^TR^{-1}H\right)^{-1} \\ &= \begin{bmatrix} 1/3 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

As expected, the value of $P^+(k)$ is equal to the value of $P(k, k-1)$ that we computed in part (b).

10.11 Under what conditions will P_y in Equation (10.100) be invertible for all k?

Solution:

P_y will be invertible if R is invertible, or if HP^-H^T is invertible. HP^-H^T will be invertible if H is full rank and P^- is invertible. P^- will be invertible if Q is invertible, or if FP^+F^T is invertible. FP^+F^T will be invertible if F is full rank and

P^+ is invertible. P^+ will be invertible if P^- is invertible or $HR^{-1}H^T$ is invertible (see Equation (5.19)).

In summary, P_y will be invertible if R is full rank, or H is full rank and (Q is full rank or (F is full rank and P_0^+ is full rank)).

Computer exercises

10.12 Consider the equations

$$\begin{aligned} 300x + 400y &= 700 \\ 100x + 133y &= 233 \end{aligned}$$

a) What is the solution of these equations?
b) What is the solution of these equations if each constant in the second equation increases by 1?
c) What is the condition number of the original set of equations?

Solution:

a). $x = 1$, $y = 1$.

b). $x = -1$, $y = 2.5$. A slight change in the second equation results in a large change in the solution.

c). The condition number is 2777, which is pretty large.

10.13 Repeat Problem 10.12 for the equations

$$\begin{aligned} 300x + 400y &= 700 \\ 100x + 200y &= 200 \end{aligned}$$

Comment on the difference between this set of equations and the set given in Problem 10.12.

Solution:

a). $x = 3$, $y = -0.5$.

b). $x = 3.03$, $y = -0.52$.

c). The condition number is 15, which is pretty small. The equations of Problem 10.12 have a large condition number, which means that a small change in the equations could lead to a large change in the solution. The equations in this problem, however, have a small condition number, which means that a small change in the equations always lead to a small change in the solution.

10.14 Tire tread is measured every τ weeks. After τ weeks, 20% of the tread has worn off, so we can model the dynamics of the tread height as $x_{k+1} = fx_k + w_k$, where $f = 0.8$, and w_k is zero-mean white noise with a variance of 0.01. We measure the tread height every τ weeks with zero-mean white measurement noise that has a variance of 0.01. The initial tread height is known to be exactly 1 cm. Write a program to simulate the system and a Kalman filter to estimate the tread height.

a) Run the program for 10 time steps per tire, and for 1000 tires. What is the mean of the 10,000 measurement residuals?

b) Suppose the Kalman filter designer incorrectly believes that 30% of the tread wears off every τ weeks. What is the mean of the 10,000 measurement residuals in this case?

c) Suppose the Kalman filter designer incorrectly believes that 10% of the tread wears off every τ weeks. What is the mean of the 10,000 measurement residuals in this case?

Solution

a). Depending on the noise history, the mean of the residuals will be somewhere between ± 0.002.

b). If the filter uses $f = 0.7$, the mean of the residuals is about 0.06.

c). If the filter uses $f = 0.9$, the mean of the residuals is about -0.07.

10.15 Consider the system described in Problem 10.14. Suppose the engineer does not know the true value of f but knows the initial probabilities $\Pr(f = 0.8) = \Pr(f = 0.85) = \Pr(f = 0.9) = 1/3$. Run the multiple-model estimator for 10 time steps on 100 tires to estimate f. The f probabilities at each time step can be taken as the mean of the 100 f probabilities that are obtained from the 100 tire simulations, and similarly for the f estimate at each time step. Plot the f probabilities and the f estimate as a function of time.

Solution:

Figure 10.1 shows the f probabilities and the f estimates as a function of time.

10.16 Consider a scalar system with $F = H = 1$ and nominal noise variances $Q = R = 5$. The true but unknown noise variances $\tilde{Q}$ and $\tilde{R}$ are given as

$$\begin{aligned}
\tilde{Q} &= (1+\alpha)Q \\
\tilde{R} &= (1+\beta)R \\
E(\alpha^2) &= \sigma_1^2 = 1/2 \\
E(\beta^2) &= \sigma_2^2 = 1
\end{aligned}$$

where α and β are independent zero-mean random variables. The variance of the *a posteriori* estimation error is P if $\alpha = \beta = 0$. In general, α and β are nonzero and the variance of the estimation error is $P + \Delta P$. Plot P, $E(\Delta P^2)$, and $(P + E(\Delta P^2))$ as a function of K for $K \in [0.3, 0.7]$. What are the minimizing values of K for the three plots?

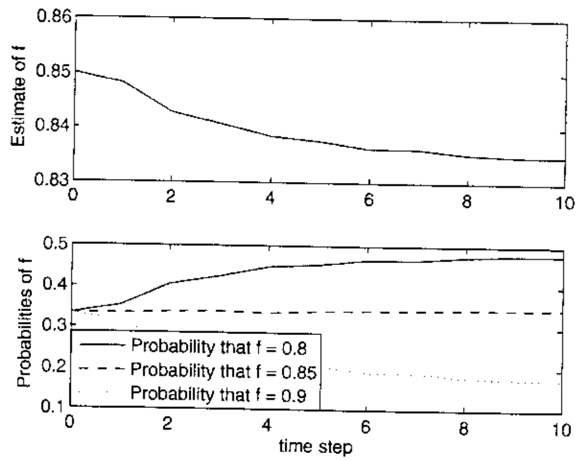

Figure 10.1 Solution to Problem 10.15

Solution:

We can use Equation (10.75) to find

$$P = \frac{(1-K)^2 Q + K^2 R}{2K - K^2}$$

We can use Equation (10.79) to find

$$\begin{aligned}
\Delta P &= \frac{\alpha(1-K)^2 Q + \beta K^2 R}{2K - K^2} \\
E(\Delta P^2) &= \frac{\sigma_1^2 (1-K)^4 Q^2 + \sigma_2^2 K^4 R^2}{(2K - K^2)^2}
\end{aligned}$$

Plotting these equations as functions of K gives Figure 10.2. The minimizing values of K are found to be

$$\begin{aligned}
\text{argmin } P &= 0.620 \\
\text{argmin } E(\Delta P^2) &= 0.476 \\
\text{argmin } P + E(\Delta P^2) &= 0.512
\end{aligned}$$

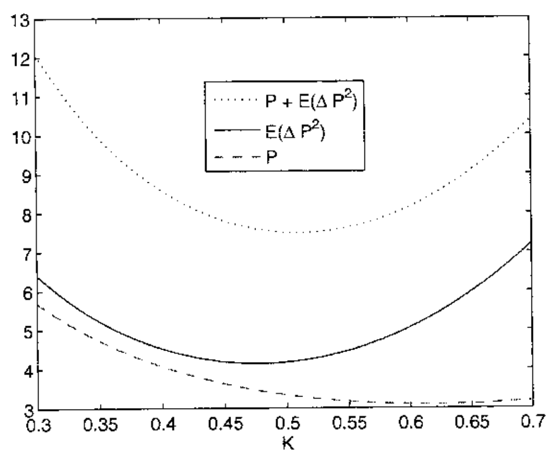

Figure 10.2 Solution to Problem 10.16

CHAPTER 11

The H_∞ filter

Problems

Written exercises

11.1 Show that $(I + A)^{-1}A = A(I + A)^{-1}$.

Solution:

First solve for $(I + A)^{-1}A$ as follows.

$$\begin{aligned}
(I + A)^{-1}(I + A) &= I \\
(I + A)^{-1} + (I + A)^{-1}A &= I \\
(I + A)^{-1}A &= I - (I + A)^{-1}
\end{aligned}$$

Next solve for $A(I + A)^{-1}$ as follows.

$$\begin{aligned}
(I + A)(I + A)^{-1} &= I \\
(I + A)^{-1} + A(I + A)^{-1} &= I \\
A(I + A)^{-1} &= I - (I + A)^{-1}
\end{aligned}$$

So we see that $(I+A)^{-1}A = A(I+A)^{-1}$.
QED

11.2 Consider a scalar system with $F = H = 1$ and with process noise and measurement noise variances Q and R. Suppose a state estimator of the form

$$\hat{x}_{k+1}^- = \hat{x}_k^- + K(y_k - \hat{x}_k^-)$$

is used to estimate the state, where K is a general estimator gain.

a) Find the optimal gain K if $R = 2Q$. Call this gain K_0. What is the resulting steady-state *a priori* estimation-error variance?

b) Suppose that $R = 0$. What is the optimal steady-state *a priori* estimation-error variance? What is the (suboptimal) steady-state *a priori* estimation-error variance if K_0 is used in the estimator? Repeat for $R = Q$ and $R = 5Q$.

Solution:

a). The optimal gain can be found a number of ways. We can use Equation (5.19) to find K_0. We could also find it as follows.

$$\begin{aligned}
e_{k+1} &= x_{k+1} - \hat{x}_{k+1}^- \\
&= (x_k + w_k) - \left[\hat{x}_k^- + K(y_k - \hat{x}_k^-)\right] \\
&= (x_k + w_k) - \left[\hat{x}_k^- + K(x_k + v_k - \hat{x}_k^-)\right] \\
&= (1-K)e_k + w_k - Kv_k \\
E(e_{k+1}^2) &= (1-K)^2 E(e_k^2) + Q + K^2 R
\end{aligned}$$

In steady state $E(e_{k+1}^2) = E(e_k^2)$ so we solve for the steady-state value as

$$E(e^2) = \frac{Q + K^2 R}{2K - K^2}$$

Now take the partial derivative with respect to K, set it equal to 0, and solve for K to get

$$K = \frac{-Q \pm \sqrt{Q^2 + 4QR}}{2R}$$

Usually we take the plus sign to get a stable filter. This gives

$$K_0 = 1/2 \quad (R = 2Q)$$

The steady-state value of P_k^- can be found by solving Equation (11.12), which gives

$$\begin{aligned}
P^- &= \frac{Q \pm \sqrt{Q^2 + 4QR}}{2} \\
&= 2Q \quad (R = 2Q)
\end{aligned}$$

where we took the plus sign in order to get a nonnegative variance.

b). If $R = 0$ then $y_k = x_k$ (no measurement noise) so our state estimation is perfect (use $K = 1$). That means the optimal $P^- = 0$. However, if K_0 is used in the estimator, then we substitute into the $E(e^2)$ equation in part (a) to obtain $E(e^2) = 4Q/3$.

If $R = Q$ then the optimal estimation-error variance is found from part (a) as $P^- = Q(1+\sqrt{5})/2 \approx 1.6Q$. However, if K_0 is used in the estimator, then we substitute into the $E(e^2)$ equation in part (a) to obtain $E(e^2) = 5Q/3 \approx 1.7Q$.

If $R = 5Q$ then the optimal estimation-error variance is found from part (a) as $P^- = Q(1+\sqrt{21})/2 \approx 2.8Q$. However, if K_0 is used in the estimator, then we substitute into the $E(e^2)$ equation in part (a) to obtain $E(e^2) = 3Q$.

11.3 Consider a scalar system with $F = H = 1$ and with process noise and measurement noise variances Q and $R = 2Q$. A Kalman filter is designed to estimate the state, but (unknown to the engineer) the process noise has a mean of $\bar{w}$.

- **a)** What is the steady-state value of the mean of the *a priori* estimation error?
- **b)** Introduce a new state-vector element that is equal to $\bar{w}$. Augment the new state-vector element to the original system so that a Kalman filter can be used to estimate both the original state element and the new state element. Find an analytical solution to the steady-state *a priori* estimation-error covariance for the augmented system.

Solution:

a). In the solution to Problem 11.2 we derived

$$e_{k+1} = (1-K)e_k + w_k - Kv_k$$

From this we derive

$$\bar{e}_{k+1} = (1-K)\bar{e}_k + \bar{w}$$

Solving for $\bar{e}_k$ gives

$$\bar{e}_k = \bar{w}\sum_{n=0}^{k-2}(1-K)^n + (1-K)^{k-1}e_1$$

The Kalman gain for this problem is $K = 1/2$ (as derived in the solution to Problem 11.2). Taking the limit of $\bar{e}_k$ as $n \to \infty$ gives

$$\bar{e} = 2\bar{w}$$

b). The augmented system is

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w_k \\ y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \end{aligned}$$

The first element of the state vector is the original state, the second element of the state vector is the noise bias $\bar{w}$, and the new w_k is now zero-mean. Equation (11.12) gives the steady-state *a priori* estimation-error covariance P as

$$P = FP(I + H^T R^{-1} HP)^{-1} F^T + Q$$

where $Q = \text{diag}(q, 0)$, and q is the variance of w_k. Multiplying out this equation and equating the elements on both sides gives

$$\begin{aligned}
P_{11} &= \frac{(P_{11}+P_{12})(1-P_{12}/R)}{1+P_{11}/R} + P_{12} + P_{22} + q \\
P_{12} &= \frac{-(P_{11}+P_{12})P_{12}/R}{1+P_{11}/R} + P_{12} + P_{22} \\
&= \frac{P_{12} - P_{12}^2/R}{1+P_{11}/R} + P_{22} \\
P_{22} &= \frac{-P_{12}^2/R}{1+P_{11}/R} + P_{22}
\end{aligned}$$

These equations can be solved to obtain

$$\begin{aligned}
P_{11} &= \frac{q+\sqrt{q^2+4qR}}{2} \\
&= 2q \quad (\text{when } R = 2q) \\
P_{12} &= 0 \\
P_{22} &= 0
\end{aligned}$$

11.4 Suppose that a Kalman filter is designed to estimate the state of a scalar system. The assumed system is given as

$$\begin{aligned}
x_{k+1} &= Fx_k + w_k \\
y_k &= Hx_k + v_k
\end{aligned}$$

where $w_k \sim (0, Q)$ and $v_k \sim (0, R)$ are uncorrelated zero-mean white noise processes. The actual system matrix is $\tilde{F} = F + \Delta F$.

a) Under what conditions is the mean of the steady-state value of the *a priori* state estimation error equal to zero?

b) What is the steady-state value of the *a priori* estimation-error variance P? How much larger is P because of the modeling error ΔF?

Solution:

a). The mean of the *a priori* estimation error is given as

$$\begin{aligned}
e_{k+1} &= x_{k+1} - \hat{x}_{k+1}^- \\
&= (Fx_k + \Delta F x_k + w_k) - \left[F\hat{x}_k^- + FK(y_k - H\hat{x}_k^-)\right] \\
&= (Fx_k + \Delta F x_k + w_k) - \left[F\hat{x}_k^- + FK(Hx_k + v_k - H\hat{x}_k^-)\right] \\
&= F(1-KH)e_k + \Delta F x_k + w_k - FKv_k \\
E(e_{k+1}) &= F(1-KH)E(e_k) + \Delta F E(x_k)
\end{aligned}$$

In steady state this becomes

$$E(e) = \frac{\Delta F E(x)}{1 - F(1-KH)}$$

This will be zero if the steady-state value $E(x) = 0$. But $E(x_{k+1})$ is given as

$$E(x_{k+1}) = \tilde{F}E(x_k)$$

This converges to zero if $|\tilde{F}| < 1$, or if $E(x_0) = 0$. These are the conditions for a zero-mean estimation error.

b). The *a priori* estimation-error variance can be found as follows.

$$E(e_{k+1}^2) = F^2(1-KH)^2E(e_k^2) + \Delta F^2 E(x_k^2) + Q + F^2K^2R$$

In steady state $E(e_{k+1}^2) = E(e_k^2)$ and we obtain

$$E(e^2) = \frac{\Delta F^2 E(x^2) + Q + F^2K^2R}{1 - F^2(1-KH)^2}$$

Now note that $E(x_{k+1}^2)$ can be found as

$$E(x_{k+1}^2) = \tilde{F}^2 E(x_k^2) + Q$$

This converges if $|\tilde{F}| < 1$, in which case the steady-state value of $E(x^2)$ is given as

$$E(x^2) = \frac{Q}{1 - \tilde{F}^2}$$

The steady-state estimation error is therefore given as

$$E(e^2) = \frac{\Delta F^2 Q/(1-\tilde{F}^2) + Q + F^2K^2R}{1 - F^2(1-KH)^2}$$

We see that the increase in $E(e^2)$ due to ΔF is given as

$$\Delta E(e^2) = \frac{\Delta F^2 Q}{(1-\tilde{F}^2)[1 - F^2(1-KH)^2]}$$

11.5 Find the stationary point of $(x_1^2 + x_1x_2 + x_2x_3)$ subject to the constraint $(x_1 + x_2 = 4)$ [Moo00].

Solution:

The augmented cost function is given as

$$J_a = x_1^2 + x_1x_2 + x_2x_3 + \lambda(x_1 + x_2 - 4)$$

The partial derivatives are given as

$$\begin{aligned}
\frac{\partial J_a}{\partial x_1} &= 2x_1 + \lambda \\
\frac{\partial J_a}{\partial x_2} &= x_1 + x_3 + \lambda \\
\frac{\partial J_a}{\partial x_3} &= x_2 \\
\frac{\partial J_a}{\partial \lambda} &= x_1 + x_2 - 4
\end{aligned}$$

Setting these partial derivatives equal to zero and solving gives

$$\begin{aligned} x_1 &= 4 \\ x_2 &= 0 \\ x_3 &= 4 \\ \lambda &= -8 \end{aligned}$$

11.6 Maximize $(14x - x^2 + 6y - y^2 + 7)$ subject to the constraints $(x + y \leq 2)$ and $(x + 2y \leq 3)$ [Lue84].

Solution:

One of four situations holds: (1) Neither of the constraints are active (i.e., the solution of the unconstrained optimization problem satisfies the constraints); (2) Only the first constraint is active; (3) Only the second constraint is active; or, (4) Both constraints are active.

If the first situation holds, then we solve the unconstrained problem as

$$\begin{aligned} \frac{\partial J}{\partial x} &= 14 - 2x = 0 \\ \frac{\partial J}{\partial y} &= 6 - 2y = 0 \end{aligned}$$

The solution to these equations is $x = 7$ and $y = 3$, but this does not satisfy the constraints.

If the second situation holds, we solve the problem with the first constraint as

$$\begin{aligned} J_a &= 14x - x^2 + 6y - y^2 + 7 + \lambda_1(x + y - 2) \\ \frac{\partial J_a}{\partial x} &= 14 - 2x + \lambda_1 = 0 \\ \frac{\partial J_a}{\partial y} &= 6 - 2y + \lambda_1 = 0 \\ \frac{\partial J_a}{\partial \lambda_1} &= x + y - 2 = 0 \end{aligned}$$

The solution to these equations is $x = 3$, $y = -1$, and $\lambda_1 = -8$. This satisfies both constraints and gives a cost function value of 33.

If the third situation holds, we solve the problem with the second constraint as

$$\begin{aligned} J_a &= 14x - x^2 + 6y - y^2 + 7 + \lambda_2(x + 2y - 3) \\ \frac{\partial J_a}{\partial x} &= 14 - 2x + \lambda_2 = 0 \\ \frac{\partial J_a}{\partial y} &= 6 - 2y + 2\lambda_2 = 0 \\ \frac{\partial J_a}{\partial \lambda_2} &= x + 2y - 3 = 0 \end{aligned}$$

The solution to these equations is $x = 5$, $y = -1$, and $\lambda_2 = -4$, but this does not satisfy the first constraint.

If the fourth situation holds, then we solve the problem with both constraints as

$$\begin{aligned}
J_a &= 14x - x^2 + 6y - y^2 + 7 + \lambda_1(x + y - 2) + \lambda_2(x + 2y - 3) \\
\frac{\partial J_a}{\partial x} &= 14 - 2x + \lambda_1 + \lambda_2 = 0 \\
\frac{\partial J_a}{\partial y} &= 6 - 2y + \lambda_1 + 2\lambda_2 = 0 \\
\frac{\partial J_a}{\partial \lambda_1} &= x + y - 2 = 0 \\
\frac{\partial J_a}{\partial \lambda_2} &= x + 2y - 3 = 0
\end{aligned}$$

The solution to these equations is $x = 1$, $y = 1$, $\lambda_1 = -20$, and $\lambda_2 = 8$. This satisfies both constraints and gives a cost function of 25.

Combining these results we see that the solution to the constrained problem is $x = 3$ and $y = -1$, which gives a cost function value of 33. Only the first constraint is active

11.7 Consider the system

$$\begin{aligned}
x_k &= \frac{1}{2}x_{k-1} + w_{k-1} \\
y_k &= x_k + v_k
\end{aligned}$$

Note that this is the system model for the radiation system described in Problem 5.1.

a) Find the steady-state value of P_k for the H_∞ filter, using a variable θ and $L = R = Q = S = 1$.

b) Find the bound on θ such that the steady-state H_∞ filter exists.

Solution:

a). Substituting $L = 1$ into Equation (11.89) gives the steady-state value of P as

$$\begin{aligned}
P &= \frac{P/4}{1 - \theta SP + P/R} + Q \\
&= \frac{PR + 4RQ - 4R\theta QSP + 4QP}{4R - 4R\theta SP + 4P}
\end{aligned}$$

Setting $R = Q = S = 1$ gives

$$\begin{aligned}
0 &= 4(1-\theta)P^2 + (4\theta - 1)P - 4 \\
P &= \frac{(1 - 4\theta) \pm \sqrt{(1-4\theta)^2 + 64(1-\theta)}}{8(1-\theta)}
\end{aligned}$$

We use a plus sign in order to get $P > 0$.

b). In order for the steady-state H_∞ filter to exist the discriminant in the above expression for P must be greater than or equal to zero.

$$\sqrt{(1-4\theta)^2 + 64(1-\theta)} \geq 0$$

This is true for $\theta \leq 1.25$ and $\theta \geq 3.25$. Another condition for the existence of the H_∞ filter is that Equation (11.90) must be satisfied.

$$\begin{aligned} P^{-1} - \theta \bar{S}_k + H_k^T R_k^{-1} H_k &> 0 \\ P^{-1} - \theta + 1 &> 0 \end{aligned}$$

This is always true if $\theta < 1$. If $\theta > 1$ then this inequality can be written as

$$P < \frac{1}{\theta - 1}$$

Substituting for P from part (a) into this equation gives

$$\begin{aligned} 1 - 4\theta + \sqrt{(1-4\theta)^2 + 64(1-\theta)} &> -8 \\ \sqrt{(1-4\theta)^2 + 64(1-\theta)} &> 4\theta - 9 \end{aligned}$$

This is satisfied if $\theta < 9/4$.

Combining all of these results gives the final answer: we require $\theta \leq 1.25$ in order for the steady-state H_∞ filter to exist

11.8 Suppose that you use a continuous-time H_∞ filter to estimate a constant on the basis of noisy measurements. The measurement noise is zero-mean and white with a covariance of R. Find the H_∞ estimator gain as a function of P_0, R, θ, and time. What is the limit of the estimator gain as $t \to \infty$? What is the maximum value of θ such that the H_∞ estimation problem has a solution? How does the value of θ influence the estimator gain?

Solution:

Equation (11.118) gives

$$\dot{P} = (\theta - 1/R)P$$

Proceeding along lines similar to Example 8.1 we find the solution of P as

$$P = \frac{RP_0}{R + P_0(1 - \theta R)t}$$

The H_∞ filter gain is then given as

$$\begin{aligned} K &= P/R \\ &= \frac{P_0}{R + P_0(1-\theta R)t} \end{aligned}$$

We see that

$$\lim_{t \to \infty} K = 0$$

From the expression for P we see that $\theta < 1/R$ in order for P to remain nonnegative. If $\theta = 0$ then we have the Kalman filter of Example 8.1. As θ increases the H_∞ estimator gain decreases more slowly with time.

11.9 Prove that $\mathcal{H}$ and $\tilde{\mathcal{H}}$ in Equations (11.134) and (11.137) are symplectic.

Solution:

As shown in Equation (7.86), a symplectic matrix A is a matrix such that

$$J^{-1}A^TJA = I \qquad \text{where } J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$$

J^{-1} is computed as

$$J^{-1} = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix}$$

If we take $\mathcal{H}$ and $\tilde{\mathcal{H}}$ from Equations (11.134) and (11.137) and substitute them for A in the above equation, we can verify after a couple of pages of tedious but straightforward algebra that the equalities are satisfied.
QED

11.10 Prove that the solution of the *a posteriori* H_∞ Riccati equation given in Equation (11.132) with $\theta = 0$ is equivalent to the solution of the steady-state *a priori* Kalman filter Riccati equation with $R = I$ and $Q = I$.

Solution:

The *a posteriori* H_∞ ARE is

$$\begin{aligned} \tilde{P} &= F\tilde{P}(H^TH\tilde{P} - \theta L^TL\tilde{P} + I)^{-1}F^T + I \\ &= F\tilde{P}(H^TH\tilde{P} + I)^{-1}F^T + I \end{aligned}$$

where we have substituted 0 for θ. The *a priori* Kalman filter ARE is obtained from Equation (5.19) as

$$\begin{aligned} P &= FPF^T - FKHPF^T + Q \\ &= FPF^T - FPH^T(HPH^T + R)^{-1}HPF^T + Q \\ &= FP\left[I - H^T(HPH^T + I)^{-1}HP\right]F^T + I \end{aligned}$$

where we have substituted I for Q and R. In order to prove that $\tilde{P}$ and P are equal, we need to prove that

$$I - H^T(HPH^T + I)^{-1}HP = (H^THP + I)^{-1}$$

This supposed equality can be written as

$$\begin{aligned} H^THP + I - H^T(HPH^T + I)^{-1}HP(H^THP + I) &= I \\ H^THP - H^T(HPH^T + I)^{-1}(HPH^T + I)HP &= 0 \\ H^THP - H^THP &= 0 \end{aligned}$$

The equality thus becomes evident.
QED

11.11 Prove that Σ in Equation (11.132) with $\theta = 0$ is equivalent to the solution of the steady-state *a posteriori* Kalman filter Riccati equation with $R = I$ and $Q = I$.

Solution:

The steady-state *a posteriori* Kalman filter Riccati equation solution is obtained from Equation (5.19) as

$$P^+ = \left[\left(P^-\right)^{-1} + H^T H\right]^{-1}$$

where P^- is the steady-state *a priori* Kalman filter Riccati equation solution, and we have substituted I for R. From Equation (11.132) we have

$$\Sigma = \left(\tilde{P}^{-1} + H^T H\right)^{-1}$$

where we have substituted 0 for θ. But from Problem 11.10 we know that $P^- = \tilde{P}$ when $Q = I$ and $R = I$ and $\theta = 0$. Therefore

$$\begin{aligned} \Sigma &= \left[\left(P^-\right)^{-1} + H^T H\right]^{-1} \\ &= P^+ \end{aligned}$$

QED

11.12 Find the *a posteriori* steady-state H_∞ filter for Example 11.5 when $\theta = 1/10$. Verify that the *a priori* and *a posteriori* Riccati equation solutions satisfy Equation (11.133).

Solution:

The *a posteriori* ARE for this problem is

$$\begin{aligned} \tilde{P} &= \frac{\tilde{P}}{\tilde{P} - \theta\tilde{P} + 1} + 1 \\ 0 &= (1-\theta)\tilde{P}^2 + (\theta - 1)\tilde{P} - 1 \\ \tilde{P} &= \frac{1 - \theta \pm \sqrt{\theta^2 - 6\theta + 5}}{2(1-\theta)} \end{aligned}$$

Substituting $\theta = 1/10$ into this equation gives $\tilde{P} = 5/3$ (we used the plus sign to give a positive $\tilde{P}$). Equation (11.133) says that

$$P^{-1} = \tilde{P}^{-1} - \theta L^T L$$

which is indeed satisfied for $P = 2$, $\theta = 1/10$, and $L = 1$ from Example 11.5, and $\tilde{P} = 5/3$ from above.

11.13 Find all possible solutions P to the *a priori* H_∞ filtering problem for Example 11.5 when $\theta = 0$. Next use Equation (11.139) to find the P solution. Repeat for $\theta = 1/10$. [Note that Equation (11.139) gives a negative solution for P and therefore cannot be used.]

Solution:

The *a priori* ARE is given in Example 11.5 as

$$\begin{aligned} P &= \frac{-\theta - 1 \pm \sqrt{\theta^2 - 6\theta + 5}}{2(2\theta - 1)} \\ &= \begin{cases} (1 \pm \sqrt{5})/2 & (\theta = 0) \\ (11 \pm 21)/16 & (\theta = 1/10) \end{cases} \end{aligned}$$

We use the positive solutions in order to solve the H_∞ filtering problem. The *a priori* Hamiltonian matrix associated with this problem is obtained from Equation (11.139) as

$$\mathcal{H} = \begin{bmatrix} 2 & 2\theta - 1 \\ -1 & 1 - \theta \end{bmatrix}$$

For $\theta = 0$ the eigendata are obtained as

$$\begin{aligned} \lambda &= \frac{2}{\sqrt{5}+3}, \frac{\sqrt{5}+3}{2} \\ v &= \begin{bmatrix} 2 \\ \sqrt{5}+1 \end{bmatrix}, \begin{bmatrix} \sqrt{5}+1 \\ -2 \end{bmatrix} \end{aligned}$$

Solving for P gives

$$\begin{aligned} P &= X_2/X_1 \\ &= \frac{-2}{\sqrt{5}+1} \\ &= \frac{1-\sqrt{5}}{2} \end{aligned}$$

This is the negative solution of the ARE and is therefore an unusable value for P.
Repeating for $\theta = 1/10$ gives

$$\mathcal{H} = \begin{bmatrix} 2 & -0.8 \\ -1 & 0.9 \end{bmatrix}$$

The eigendata are obtained as

$$\begin{aligned} \lambda &= 0.4, 2.5 \\ v &= \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 8 \\ -5 \end{bmatrix} \end{aligned}$$

Solving for P gives

$$\begin{aligned} P &= X_2/X_1 \\ &= -5/8 \end{aligned}$$

Again this is the negative solution of the ARE and is therefore an unusable value for P.

Computer exercises

11.14 Generate the time-varying solution to P_k for Problem 11.7 with $P_0 = 1$. What is the largest value of θ for which Equation (11.90) will be satisfied for all k up to and including $k = 20$? Answer to the nearest 0.01. Repeat for $k = 10$, $k = 5$, and $k = 1$.

Solution:

$$\theta_{\max} = \begin{cases} 1.25 & \text{for } k = 20 \\ 1.27 & \text{for } k = 10 \\ 1.33 & \text{for } k = 5 \\ 1.60 & \text{for } k = 1 \end{cases}$$

11.15 Consider the vehicle navigation problem described in Example 7.12. Design a Kalman filter and an H_∞ filter to estimate the states of the system. Use the following parameters.

$$\begin{aligned} T &= 3 \\ u_k &= 1 \\ Q &= \text{diag}(4, 4, 1, 1) \\ R &= \text{diag}(900, 900) \\ \text{heading angle} &= 0.9\pi \\ x(0) &= \hat{x}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned}$$

Simulate the system and the filters for 300 seconds. In the H_∞ filter use $S = L = I$ and $\theta = 0.0005$.

a) Plot the position estimation errors for the Kalman and H_∞ filters. What are the RMS position estimation errors for the two filters?

b) Now suppose that unknown to the filter designer, $u_k = 2$. Plot the position estimation errors for the Kalman and H_∞ filters. What are the RMS position estimation errors for the two filters?

c) What are the closed loop estimator eigenvalues for the Kalman and H_∞ filters? Do their relative magnitudes agree with your intuition?

d) Use MATLAB's DARE function to find the largest θ for which a steady-state solution exists to the H_∞ DARE. Answer to the nearest 0.0001. How well does the H_∞ filter work for this value of θ? What are the closed-loop eigenvalues of the H_∞ filter for this value of θ?

Solution:

a). The position estimation errors are calculated as the square root of the sum of the squares of the first two state element estimation errors. Figure 11.1 shows the position estimation errors for the two filters for the two different scenarios.

b). If $u_k = 1$ (as assumed by the filters) then the RMS position estimation errors are 28.6 for the Kalman filter and 31.0 for the H_∞ filter. If $u_k = 2$ (unknown to the filters) then the RMS position estimation errors are 106.4 for the Kalman filter and 77.2 for the H_∞ filter. Your results may vary depending on the particular noise history that you simulated. This illustrates the general idea that the Kalman filter is better if the system dynamics are well known, but the H_∞ filter is more robust to unmodeled dynamics.

c). The closed loop estimator eigenvalues are the eigenvalues of $F(I - KH)$. For the Kalman filter the eigenvalue magnitudes are all equal to 0.80. For the H_∞ filter the eigenvalue magnitudes are equal to 0.54 and 0.75. We see that the H_∞ filter eigenvalues are smaller than those of the Kalman filter, which means that the H_∞ filter is "faster" than the Kalman filter. I.e., the H_∞ filter is more responsive to input measurements, which agrees with intuition (less reliance on the system model, and more reliance on the measurements).

d). MATLAB's DARE function shows that $\theta = 0.0011$ is the largest θ for which a steady-state solution exists to the H_∞ DARE. The H_∞ filter does not work for this value of θ. The estimator eigenvalue magnitudes are equal to 14.45 and 0.93, which means that the H_∞ filter is not stable.

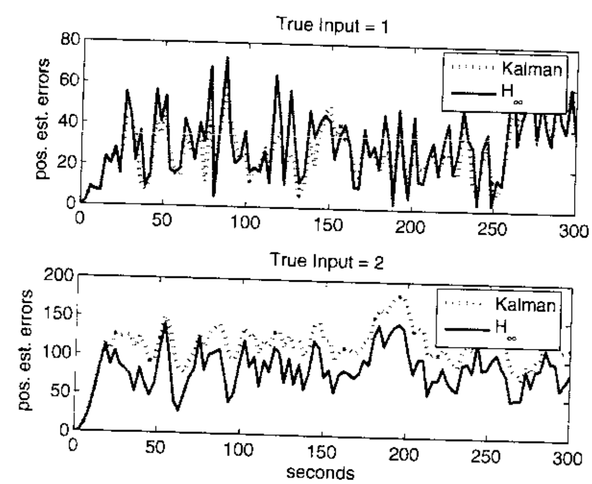

Figure 11.1 Solution to Problem 11.15

CHAPTER 12

Additional topics in H_∞ filtering

Problems

Written exercises

12.1 Consider the system described in Example 12.1 with $Q = R = 1$.

a) Find the steady-state *a priori* estimation-error variance P as a function of the estimator gain K.

b) Find $||G_{\tilde{x}e}||_\infty^2$, the square of the infinity-norm of the transfer function from the noise w and v to the *a priori* state estimation error $\tilde{x}$, as a function of the estimator gain K.

c) Find the estimator gain K that minimizes $(P + ||G_{\tilde{x}e}||_\infty^2)$.

Solution:

a). The *a priori* estimation error is found as follows.

$$\begin{aligned}
\tilde{x}_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\
&= x_k + w_k - \hat{x}_k - K(y_k - \hat{x}_k) \\
&= x_k + w_k - \hat{x}_k - K(x_k + v_k - \hat{x}_k)
\end{aligned}$$

$$\begin{aligned} &= (1-K)\tilde{x}_k + w_k - Kv_k \\ P_{k+1} &= (1-K)^2 P_k + Q + K^2 R \end{aligned}$$

In steady state this is solved as

$$P = \frac{Q + K^2 R}{1 - (1-K)^2}$$

With $Q = R = 1$ this becomes

$$P = \frac{1 + K^2}{2K - K^2}$$

b). From part (a) we have

$$\begin{aligned} \tilde{x}_{k+1} &= (1-K)\tilde{x}_k + w_k - Kv_k \\ (z - 1 + K)\tilde{X}(z) &= W(z) - KV(z) \end{aligned}$$

This shows that the transfer function $G_{\tilde{x}e}$ is given as

$$G_{\tilde{x}e} = \frac{1}{z - 1 + K} \begin{bmatrix} 1 & -K \end{bmatrix}$$

The square of the singular value is given as

$$\begin{aligned} \sigma^2(G_{\tilde{x}e}) &= G_{\tilde{x}e}(e^{j\theta}) G_{\tilde{x}e}^T(e^{-j\theta}) \\ &= \frac{1 + K^2}{1 + (K-1)(e^{j\theta} + e^{-j\theta}) + (1-K)^2} \\ &= \frac{1 + K^2}{1 + 2(K-1)\cos\theta + (1-K)^2} \end{aligned}$$

The infinity-norm squared is the supremum of the above value. If $K \le 1$ this occurs at $\theta = 0$

$$\begin{aligned} ||G_{\tilde{x}e}||_\infty^2 &= \sup_\theta \sigma^2(G_{\tilde{x}e}) \\ &= \frac{1 + K^2}{K^2} \qquad (K \le 1) \end{aligned}$$

If $K \ge 1$ the supremum of $\sigma^2(G_{\tilde{x}e})$ occurs at $\theta = \pi$.

$$\begin{aligned} ||G_{\tilde{x}e}||_\infty^2 &= \sup_\theta \sigma^2(G_{\tilde{x}e}) \\ &= \frac{1 + K^2}{K^2 - 4K + 4} \qquad (K \ge 1) \end{aligned}$$

c). $J = (P + ||G_{\tilde{x}e}||_\infty^2)$ is given as

$$J = \frac{1 + K^2}{2K - K^2} + \frac{1 + K^2}{K^2} \qquad (K \le 1)$$

Taking the partial with respect to K gives

$$\begin{aligned}\frac{\partial J}{\partial K} &= \frac{2K(K^3+3K-4)}{(2K^2-K^3)^2} \\ &= \frac{2K(K-1)(K^2+K+4))}{(2K^2-K^3)^2}\end{aligned}$$

Setting this equal to zero gives $K = 1$. We get the same answer if we work it out for $K \geq 1$.

12.2 Verify that if $\theta = 0$, the Riccati equation associated with the mixed Kalman/H_∞ filter in Equation (12.6) reduces to the Riccati equation associated with the Kalman filter.

Solution:

Substituting $\theta = 0$ into Equations (12.6) and (12.7) gives

$$\begin{aligned}P &= FPF^T + Q - P_a V^{-1} P_a^T \\ P_a &= FPH^T \\ V &= R + HPH^T\end{aligned}$$

Combining these equations gives

$$P = FPF^T + Q - FPH^T(HPH^T + R)^{-1}HPF^T$$

This is the same as the Kalman filter Riccati equation as given in Equation (7.42) (with $M = 0$).
QED

12.3 Suppose that the hybrid filter gain of Equation (12.13) is used for the system of Example 12.1 with $\theta = 1/2$. For what values of d will the hybrid filter be stable?

Solution:

From Example 11.2 recall that the Kalman gain $K_2 = (1+\sqrt{5})/(3+\sqrt{5}) = (\sqrt{5}-1)/2$, and when $\theta = 1/2$ the H_∞ gain $K_\infty = 1$. The closed-loop system matrix of the state estimator is

$$\begin{aligned}1 - K &= 1 - dK_2 - (1-d)K_\infty \\ &= d(3-\sqrt{5})/2\end{aligned}$$

For stability we require $|1 - K| < 1$ which means that $|d| < 2/(3-\sqrt{5}) \approx 2.62$.

12.4 Suppose that the robust filter of Section 12.2 is used for a system with n states and r measurements. What are the dimensions of M_1, M_2, Γ, and N?

Solution:

M_1 is $n \times m$, M_2 is $r \times m$, Γ is $m \times q$, and N is $q \times n$, where m and q can be any positive integers.

12.5 Suppose that a system matrix is given as

$$F = \begin{bmatrix} 0.4 \pm 0.2 & 0.4 \\ -0.4 & 1 \end{bmatrix}$$

(Note that this is the system matrix of Example 4.1 in case the effect of overcrowding on the predator population is uncertain.) Give an M_1 and N matrix that satisfy Equation (12.15) for this uncertainty.

Solution:

If Γ is a scalar then Equation (12.16) implies that $|\Gamma| \leq 1$. Then if $M_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and $N = \begin{bmatrix} 0.2 & 0 \end{bmatrix}$, we see that

$$M_1 \Gamma N = \begin{bmatrix} \Delta F_{11} & 0 \\ 0 & 0 \end{bmatrix}$$

where $\Delta F_{11} \in [-0.2, 0.2]$.

12.6 Consider an uncertain system with $F = -1$, $H = 1$, $Q = R = 1$, $M_1 = 1/5$, $M_2 = 0$, and $N = 1$. Suppose that $\epsilon = 0$ is used to design a robust mixed Kalman/H_∞ filter.

a) For what values of α will the steady-state value of $\tilde{P}$ in Equation (12.22) be real and positive?

b) For what values of α will the steady-state value of $\tilde{P}$ satisfy the second condition of Equation (12.24)?

Solution:

a). When $H = Q = R = 1$ and $M_2 = 0$, the steady-state version of Equation (12.22) can be written as

$$N^2 \tilde{P}^2 + (-\alpha + \alpha F^2 - N^2 - \alpha F^2 N^2 - \epsilon N^2)\tilde{P} + \alpha + \alpha^2 M_1^2 + \alpha\epsilon = 0$$

When $\epsilon = 0$ the solution to this can be written as

$$\tilde{P} = \frac{\alpha - \alpha F^2 + N^2 + \alpha F^2 N^2 \pm \sqrt{(\alpha - \alpha F^2 + N^2 + \alpha F^2 N^2)^2 - 4N^2(\alpha + \alpha^2 M_1^2)}}{2N^2}$$

When $F = -1$, $M_1 = 1/5$, and $N = 1$, this can be written as

$$\tilde{P} = \frac{1 + \alpha \pm \sqrt{21\alpha^2/25 - 2\alpha + 1}}{2}$$

The discriminant is nonnegative for $\alpha \leq 5/7$ or $\alpha \geq 5/3$, so this is the condition for $\tilde{P}$ to be real. For $\tilde{P}$ to be positive we require

$$\begin{aligned} \alpha &\in (0, 5/7] \text{ or} \\ \alpha &\in [5/3, \infty) \end{aligned}$$

Either the plus or minus sign can be used to give a positive solution in the $\tilde{P}$ expression if α is in the above range.

b). The second condition of Equation (12.24) can be written for this problem as

$$\begin{aligned} \alpha I - N\tilde{P}N^T &> 0 \\ \alpha - \tilde{P} &> 0 \end{aligned}$$

We will use the minus sign in the $\tilde{P}$ solution from part (a) to make the condition easier to satisfy. The above condition can then be written as

$$\begin{aligned} \alpha - 1 + \sqrt{21\alpha^2/25 - 2\alpha + 1} &> 0 \\ \alpha - 1 &> -\sqrt{21\alpha^2/25 - 2\alpha + 1} \end{aligned}$$

If $\alpha < 1$ this condition reduces to $4\alpha^2/25 < 0$, which is never satisfied. If $\alpha > 1$ this condition reduces to $4\alpha^2/25 > 0$, which is always satisfied. Combining this with the results from part (a) we see that $\alpha \geq 5/3$ ensures that the second condition of Equation (12.24) is satisfied.

12.7 Consider a constrained H_∞ state estimation problem with

$$\begin{aligned} F &= \begin{bmatrix} 1 & 1/2 \\ 1/2 & 0 \end{bmatrix} \\ G = H &= \begin{bmatrix} G_1 & 0 \end{bmatrix} \\ D &= \begin{bmatrix} 1 & 1 \end{bmatrix} \\ Q &= \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \end{aligned}$$

Find the steady-state constrained Riccati solution for P from Equation (12.50). For what values of G_1 will the condition of Equation (12.51) be satisfied?

Solution:

Substituting the values for F, G, H, D, and Q into the steady-state version of Equation (12.50) gives

$$\begin{aligned} P &= \left(I - D^T D\right) F P F^T \left(I - D^T D\right)^T + Q \\ \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix} &= \begin{bmatrix} P_{11}/4 + Q_1 & P_{11}/2 + P_{12}/4 \\ P_{11}/2 + P_{12}/4 & P_{11} + P_{12} + P_{22}/4 + Q_2 \end{bmatrix} \end{aligned}$$

Equating elements on the right and left side of the above equation and solving for the elements of P gives

$$\begin{aligned} P_{11} &= 4Q_{11}/3 \\ P_{12} &= 8Q_{11}/9 \\ P_{22} &= 4/3 + 80Q_{11}/27 \end{aligned}$$

Equation (12.51) can be written for this problem as

$$\begin{aligned} 1 - GPG^T &\geq 0 \\ 1 - G_1^2 P_{11} &\geq 0 \\ G_1 &\leq \sqrt{3/(4Q_{11})} \end{aligned}$$

Computer exercises

12.8 Consider a two-state Newtonian system as discussed in Example 9.1 with $T = 1$, $a = 1$, and $R = 1$.

a) What is the steady-state Kalman gain?

b) What is the maximum θ for which the H_∞ estimator exists? Answer to the nearest 0.01. What is the H_∞ gain for this value of θ?

c) What is the H_∞ gain when $\theta = 0.5$? Plot the maximum estimator eigenvalue magnitude as a function of d for the hybrid filter of Equation (12.13) when $\theta = 0.5$.

Solution:

a). The steady-state Kalman gain is found by solving the DARE described in Section 7.3 as $K = \begin{bmatrix} 0.7691 & 0.4805 \end{bmatrix}^T$.

b). By iterative trial and error we find the maximum θ for Equation (11.90) to hold as $\theta = 0.77$. The gain for this value of θ is given as $K = \begin{bmatrix} 466.5385 & 704.1662 \end{bmatrix}^T$.

c). When $\theta = 0.5$ the H_∞ gain is given as $K = \begin{bmatrix} 1.5751 & 1.6540 \end{bmatrix}^T$. Figure 12.1 shows the maximum estimator eigenvalue magnitude as a function of d when $\theta = 0.5$.

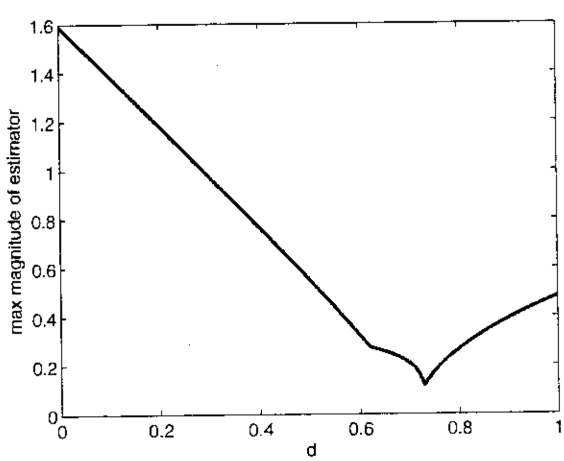

Figure 12.1 Solution to Problem 12.8

12.9 Implement the time-varying Riccati equations for the robust mixed Kalman/H_∞ filter for $F = 1/2$, $H = Q = R = 1$, $M_1 = 1/4$, $M_2 = 0$, $N = 1$, $\epsilon = 0$, $\theta = 1/10$, and $S_1 = S_2 = 1$.

a) At what time do the conditions of Equation (12.24) fail to be satisfied when $\alpha = 2$? Repeat for $\alpha = 3$, 4, 5, and 6.

b) What is the steady-state theoretical bound on the estimation error when $\alpha = 10$? Repeat for $\alpha = 20$, 30, and 40.

Solution:

a). The times at which the conditions of Equation (12.24) fail to be satisfied when $\alpha = \{2, 3, 4, 5, 6\}$ are $k = \{1, 3, 6, 17, \text{never}\}$.

b). The steady-state theoretical bound P on the estimation error when $\alpha = \{10, 20, 30, 40\}$ are $\{6.9, 10.2, 12.8, 15.2\}$.

12.10 Consider a constrained H_∞ state estimation problem with

$$\begin{aligned} F &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\ H &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\ G &= \begin{bmatrix} G_1 & 0 \end{bmatrix} \\ D &= \begin{bmatrix} 1 & 1 \end{bmatrix} \\ Q &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Implement the Σ_k and P_k expressions from Equation (12.50).

a) What is the largest value of G_1 for which P_k reaches a positive definite steady-state solution that satisfies the condition given in Equation (12.51)? Answer to the nearest 0.01. What is the resulting steady-state value of P?

b) Set G_1 equal to 1% of the maximum G_1 that you found in part (a). What is the new steady-state value of P? Give an intuitive explanation for why P gets smaller when G_1 gets smaller.

Solution:

a). By manual iteration we find that the largest G_1 that satisfies the required conditions is $G_1 = 0.44$. This gives the following steady-state P.

$$P = \begin{bmatrix} 4.9824 & 4.9735 \\ 4.9735 & 7.9577 \end{bmatrix}$$

b). When $G_1 = 0.0044$ we get the following steady-state P.

$$P = \begin{bmatrix} 4.5520 & 4.3323 \\ 4.3323 & 6.9325 \end{bmatrix}$$

This is smaller than the result from part (a), which agrees with intuition. G_1 is proportional to the magnitude of the adversary's noise input. As G_1 decreases, the noise input decreases, which means that we should expect to be able to attain a smaller estimation-error variance.

CHAPTER 13

Nonlinear Kalman filtering

Problems

Written exercises

13.1 Consider the scalar system

$$\begin{aligned} \dot{x} &= -x + w \\ y &= x + v \end{aligned}$$

The process noise has a mean value of 2, and the measurement noise has a mean value of 3. Redefine the noise quantities and the state to obtain an equivalent system of the form

$$\begin{aligned} \dot{x}' &= Ax' + Bu + w' \\ y &= Cx' + v' \end{aligned}$$

so that the new noise quantities w' and v' both have mean values of 0.

Solution:

Define $w' = w - 2$ and $v' = v - 3$. This gives

$$\dot{x} = -x + 2 + w'$$

Optimal State Estimation Solution Manual, First Edition. By Dan J. Simon

$$y = x + 3 + v'$$

Now define $x' = x + 3$. This gives

$$\begin{aligned} \dot{x}' &= -x' + 5 + w' \\ y &= x' + v' \end{aligned}$$

13.2 Consider the scalar system

$$\dot{x} = -x + u + w$$

w is zero-mean process noise with a variance of Q. The control has a mean value of u_0, an uncertainty of 2 (one standard deviation), and is uncorrelated with w. Rewrite the system equations to obtain an equivalent system with a normalized control that is perfectly known. What is the variance of the new process noise term in the transformed system equation?

Solution:

Note that $\dot{x} = -x + u_0 + \Delta u + w$, where u_0 is known, Δu is zero-mean noise with a variance of 4, and w is zero-mean noise with a variance of Q. Since Δu and w are uncorrelated, this can be written as

$$\begin{aligned} \dot{x} &= -x + u_0 + w' \\ w' &\sim (0, Q + 4) \end{aligned}$$

where u_0 is perfectly known.

13.3 Suppose that x is a constant scalar, and $y_k = \sqrt{x}(1 + v_k)$ are noisy measurements, where $v_k \sim N(0, R)$.

a) An intuitive way to estimate x is to set $\hat{x}_k = y_k^2$. Compute the mean and variance of the estimation error for this estimate. Your answer should be a function of x and R. Hint: recall that $E(v_k^3) = 0$ and $E(v_k^4) = 3R^2$.

b) Perhaps a better estimate for x_k could be obtained by averaging all previous values of y_k^2. That is,

$$\hat{x}_k = \frac{1}{k} \sum_{i=1}^{k} y_i^2$$

Compute the mean and variance of the estimation error for this estimate. Your answer should be a function of k, x, and R. Note that if you substitute $k = 1$ into your solution, you should get the same answer as part (a). What is the variance as $k \to \infty$?

c) Write the extended Kalman filter equations to estimate x. What is the theoretical mean and variance of the EKF estimate as $k \to \infty$?

Solution:

a). The mean of the estimation error is computed as

$$E(x - \hat{x}_k) = E(x - y_k^2)$$

$$\begin{aligned}
&= E[x - x(1+v_k^2)] \\
&= -E(2xv_k + xv_k^2) \\
&= -xR
\end{aligned}$$

The variance of the estimation error is computed as

$$\begin{aligned}
E[(x-\hat{x})^2] &= E[(x - y_k^2)^2] \\
&= E(4x^2v_k^2 + x^2v_k^4 + 4x^2v_k^3) \\
&= 4x^2R + 3x^2R^2
\end{aligned}$$

b). The mean of the estimation error is computed as

$$\begin{aligned}
E(x-\hat{x}) &= E(x - \frac{1}{k}\sum_{i=1}^{k} y_i^2) \\
&= E(x - \frac{1}{k}\sum_{i=1}^{k} x(1+v_i)^2) \\
&= E(x - x - \frac{1}{k}\sum_{i=1}^{k} 2xv_i - \frac{1}{k}\sum_{i=1}^{k} xv_i^2) \\
&= -xR/k
\end{aligned}$$

The variance of the estimation error is computed as

$$\begin{aligned}
E[(x-\hat{x})^2] &= E[(x - \frac{1}{k}\sum_{i=1}^{k} y_i^2)^2] \\
&= E\left[\frac{1}{k}(2xv_i + xv_i^2)\right]^2 \\
&= \frac{1}{k^2}\sum_{i,j=1}^{k} E(2xv_i + xv_i^2)(2xv_j + xv_j^2) \\
&= \frac{1}{k^2}\sum_{i,j=1}^{k} E(4x^2v_iv_j + 2x^2v_i^2v_j + 2x^2v_iv_j^2 + x^2v_i^2v_j^2) \\
&= \frac{1}{k^2}\sum_{i,j=1}^{k} E(4x^2v_iv_j + x^2v_i^2v_j^2) \\
&= \frac{1}{k^2}(4x^2Rk + x^2 3R^2k + k(k-1)x^2R^2) \\
&= \frac{4x^2R + (k+2)x^2R^2}{k}
\end{aligned}$$

As $k \to \infty$ the variance approaches x^2R^2.

c). The extended Kalman filter equations are given as

$$P_k^- = P_{k-1}^+$$

$$
\begin{aligned}
\hat{x}_k^- &= \hat{x}_{k-1}^+ \\
H_k &= \frac{1}{2}\left(\hat{x}_k^-\right)^{-1/2} \\
M_k &= \sqrt{\hat{x}_k^-} \\
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + M_k R M_k^T)^{-1} \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - \sqrt{\hat{x}_k^-}) \\
P_k^+ &= (1 - K_k H_k) P_k^-
\end{aligned}
$$

As $k \to \infty$, $P_k^+ \to 0$. This is because $Q = 0$ (i.e., there is no process noise). This can be seen by solving the above equations for the steady-state value of P_k^+.

13.4 Consider the system

$$
\begin{aligned}
x_{k+1} &= x_k + w_k \\
y_k &= x_k + v_k^2
\end{aligned}
$$

where w_k and v_k are uniformly distributed, uncorrelated, zero-mean white noise processes with variances Q and R, respectively.

a) What is the mean of the *a posteriori* estimation error for the discrete EKF?

b) Modify the measurement equation by subtracting the known bias of the measurement noise so that the modified measurement noise is zero-mean. What is the variance of the modified measurement noise?

Solution:

a). The M_k quantity is given as

$$
\begin{aligned}
M_k &= \frac{\partial h}{\partial v} \\
&= 2v_k \\
&= 0
\end{aligned}
$$

where we have evaluated the partial derivative at the nominal value $v_k = 0$. The Kalman gain is given as

$$
\begin{aligned}
K_k &= P_k^- \left(H_k P_k^- H_k^T + M_k R_k M_k^T\right)^{-1} \\
&= 1
\end{aligned}
$$

where we have used the fact that $H_k = 1$. The state estimate is given as

$$
\begin{aligned}
\hat{x}_k^+ &= \hat{x}_k^- + K_k\left(y_k - h(\hat{x}_k^-, 0)\right) \\
&= y_k
\end{aligned}
$$

The estimation error is then given as

$$
e_k = x_k - \hat{x}_k^+
$$

$$
\begin{aligned}
&= x_k - y_k \\
&= x_k - x_k - v_k^2 \\
&= -v_k^2 \\
E(e_k) &= E(-v_k^2) \\
&= -R
\end{aligned}
$$

b). The mean of the measurement noise v_k^2 is R, so the modified measurement equation is

$$
\begin{aligned}
y'_k &= y_k - R \\
&= x_k - R + v_k^2 \\
&= x_k + \tilde{v}_k
\end{aligned}
$$

The mean of the modified measurement noise $\tilde{v}_k$ is zero. The variance of $\tilde{v}_k$ is found as

$$
\begin{aligned}
E(\tilde{v}_k^2) &= E[(v_k^2 - R)^2] \\
&= E(v_k^4) - 2RE(v_k^2) + R^2 \\
&= E(v_k^4) - R^2
\end{aligned}
$$

Knowing that v_k is uniformly distributed on $[-c, c]$ with a variance of R, we can solve for $c = \sqrt{3R}$. We can then solve for

$$
\begin{aligned}
E(v_k^4) &= \frac{1}{2c} \int_{-c}^{c} v^4 \, dv \\
&= c^4/5 \\
&= 9R^2/5
\end{aligned}
$$

Substituting this back into the expression for $E(\tilde{v}_k^2)$ gives

$$
E(\tilde{v}_k^2) = 4R^2/5
$$

13.5 Consider the nonlinear system

$$
\begin{aligned}
x_{k+1} &= -x_k^2 + u_k + w_k \\
y_k &= 4x_k^2 + v_k
\end{aligned}
$$

Find the nominal values for x_k and y_k when $x_0 = 0$ and $u_k = 1$.

Solution:

The nominal noise values are $w_k = v_k = 0$. Substituting these values into the state and measurement equations gives the nominal trajectory

$$
\begin{aligned}
x_k &= (-1)^k \\
y_k &= 4(-1)^k
\end{aligned}
$$

13.6 Consider the system $x_{k+1} = x_k^2 + w_k$, where w_k is zero-mean. The initial state x_0 is uniformly distributed between 0 and 1. An EKF is initialized with $\hat{x}_0^+ = E(x_0)$. What is $E(x_1)$? What is $\hat{x}_1^-$? This problem illustrates the fact that the state estimate of an EKF is not always equal to the expected value of the state.

Solution:

$$
\begin{aligned}
E(x_1) &= E(x_0^2 + w_0) \\
&= E_{w_0}\left[\int_{x_0} (x_0^2 + w_0)\mathrm{pdf}(x_0)\, dx_0\right] \\
&= E_{w_0}\left[\int_0^1 (x_0^2 + w_0)\, dx_0\right] \\
&= E_{w_0}\left[\left(x_0^3/3 + w_0 x_0\right)_0^1\right] \\
&= E_{w_0}\left[1/3 + w_0\right] \\
&= 1/3
\end{aligned}
$$

$$
\begin{aligned}
\hat{x}_1^- &= [\hat{x}_0^+]^2 \\
&= [E(x_0)]^2 \\
&= 1/4
\end{aligned}
$$

13.7 Find the terminal velocity of the falling body of Example 13.2 if the terminal velocity occurs at an altitude of 1 mile.

Solution:

The terminal velocity is reached when $\dot{x}_2 = 0$. This gives

$$\rho_0 \exp(-x_1/k)x_2^2/2x_3 - g = 0$$

At a 1 mile altitude, $x_1 = 5280$ feet. Solving the above equation for x_2 gives

$$
\begin{aligned}
x_2 &= \sqrt{2gx_3 \exp(x_1/k)/\rho_0} \\
&= 6940 \text{ ft/s}
\end{aligned}
$$

13.8 Consider the hybrid scalar system

$$
\begin{aligned}
\dot{x} &= f(x) + w, \qquad w \sim N(0, Q) \\
y_k &= h(x_k) + v_k, \qquad v_k \sim N(0, R)
\end{aligned}
$$

The estimator that is used for the system is

$$\hat{x}_k = a + by_k + cy_k^2$$

Suppose that the state $x(t)$ is normally distributed with a mean of zero and a variance of P_x.

- **a)** Find an equation relating a, b, and c that must be satisfied in order for $\hat{x}_k$ to be an unbiased estimate of $x(t_k)$ [Gel74].
- **b)** Find values of a, b, and c so that $\hat{x}_k$ is the minimum-variance estimate. Assume that $h(x)$ is an odd function of x.

Solution:

a).

$$\begin{aligned} E(\hat{x}_k) &= E(a + by_k + cy_k^2) \\ &= a + bE[h(x_k)] + cE[h^2(x_k)] + cR \end{aligned}$$

In order for $\hat{x}_k$ to be an unbiased estimate of $x(t_k)$, the above quantity must be equal to the mean of $x(t_k)$, which is zero. Therefore

$$a + bE[h(x_k)] + cE[h^2(x_k)] + cR = 0$$

b). The variance of the estimation error is

$$\begin{aligned} P &= E[(x - \hat{x})^2] \\ &= E[(x - a - by - cy^2)^2] \\ &= E[x^2 - 2ax - 2bx(h + v) - 2cx(h + v)^2 + a^2 + 2ab(h + v) + \\ &\quad 2ac(h + v)^2 + b^2(h + v)^2 + 2bc(h + v)^3 + c^2(h + v)^4] \end{aligned}$$

where we have omitted subscripts for ease of notation. Realizing that h and v are independent, v is zero-mean, x and v are independent, x is zero-mean, and odd moments of zero-mean symmetrically distributed random variables are zero (see Section 2.2), we obtain

$$\begin{aligned} P &= P_x - 2bE(xh) - 2cE(xh^2) + a^2 + 2abE(h) + 2acE(h^2 + v^2) + \\ &\quad b^2E(h^2 + v^2) + 2bcE(h^3 + 3hv^2) + c^2E(h^4 + 4h^2v^2 + v^4) \end{aligned}$$

Taking the partial derivatives with respect to a, b, and c in order to find the minimum gives

$$\begin{aligned} \frac{\partial P}{\partial a} &= 2a - 2bE(h) + 2cE(h^2 + v^2) \\ \frac{\partial P}{\partial b} &= -2E(xh) + 2aE(h) + 2bE(h^2 + v^2) + 2cE(h^3 + 3hv^2) \\ \frac{\partial P}{\partial c} &= -2E(xh^2) + 2aE(h^2 + v^2) + 2bE(h^3 + 3hv^2) + 2cE(h^4 + 4h^2v^2 + v^4) \end{aligned}$$

Since $h(x)$ is an odd function of x, and x is zero-mean with a symmetric pdf, we know that $h(x)$, $h^3(x)$, and $xh^2(x)$ are also zero-mean. Setting the above partial derivatives equal to zero gives

$$\begin{aligned} 2a + 2cE(h^2 + v^2) &= 0 \\ -2E(xh) + 2bE(h^2 + v^2) &= 0 \\ 2aE(h^2 + v^2) + 2cE(h^4 + 4h^2v^2 + v^4) &= 0 \end{aligned}$$

These equations are solved for

$$\begin{aligned} a = c &= 0 \\ b &= \frac{E(xh)}{E(h^2) + R} \end{aligned}$$

13.9 Suppose for a scalar system that $P_k^- = 1$, $R = 1$, and $H = 3$. What is the value of P_k^+ as given by Equation (5.19)? What will be the computed value of P_k^+ if $H = 2$ is used instead? What will be the computed value of P_k^+ if $H = 1$ is used instead? This illustrates how the iterated Kalman filter gets a more accurate estimate of P_k^+ by using a more accurate value for H_k.

Solution:

$$\begin{aligned}
P_k^+ &= P_k^- - P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} H_k P_k^- \\
&= P_k^- - \frac{\left(P_k^-\right)^2 H^2}{P_k^- H^2 + R} \\
&= \begin{cases} 0.1 & (H = 3) \\ 0.2 & (H = 2) \\ 0.5 & (H = 1) \end{cases}
\end{aligned}$$

13.10 Consider a system with the measurement equation $y_k = x_k^2 + v_k$. At time k the *a priori* state estimate is $\hat{x}_k^- = 1$, the true state is $x_k = 5$, and the measurement is $y_k = 25$. The *a priori* estimation-error variance is $P_k^- = 1$, and the measurement noise variance is $R_k = 4$. Use the iterated EKF algorithm to find $\hat{x}_{k,1}^+$ and $\hat{x}_{k,2}^+$. Although the iterated EKF does not always improve the *a posteriori* state estimate, this problem illustrates how it usually does.

Solution:

At the first iteration we execute the standard EKF equations.

$$\begin{aligned}
H_{k,0} &= 2\hat{x}_{k,0}^+ \\
&= 2 \\
K_{k,0} &= P_k^- H_{k,0} / \left(P_k^- H_{k,0}^2 + R_k\right) \\
&= 1/4 \\
P_{k,1}^+ &= P_k^- - K_{k,0} H_{k,0} P_k^- \\
&= 1/2 \\
\hat{x}_{k,1}^+ &= \hat{x}_k^- + K_{k,0} \left[y_k - \left(\hat{x}_k^-\right)^2\right] \\
&= 7
\end{aligned}$$

At the second iteration we execute the following equations.

$$\begin{aligned}
H_{k,1} &= 2\hat{x}_{k,1}^+ \\
&= 14 \\
K_{k,1} &= P_k^- H_{k,1} / \left(P_k^- H_{k,1}^2 + R_k\right) \\
&= 7/100 \\
P_{k,2}^+ &= P_k^- - K_{k,1} H_{k,1} P_k^- \\
&= 1/50
\end{aligned}$$

$$\begin{aligned} \hat{x}_{k,2}^+ &= \hat{x}_k^- + K_{k,1}\left[y_k - \left(\hat{x}_{k,1}^+\right)^2 - H_{k,1}\left(\hat{x}_k^- - \hat{x}_{k,1}^+\right)\right] \\ &= 5.2 \end{aligned}$$

13.11 Prove Lemma 6 for scalar random variables x.

Solution:

For scalar $x \sim N(0, P)$ the first expression in the lemma reduces to

$$E(Ax^3) = 0$$

We know this is true because zero-mean RVs with symmetric pdf's have odd central moments that are equal to zero (see Section 2.2).

The second expression in the lemma reduces to

$$E(ABx^4) = 3ABP^2$$

We know this is true because the fourth central moment of a Gaussian RV is equal to $3P^2$, as can be seen from most standard probability texts [Pee01, Pap02].

13.12 Suppose you have the process equation $\dot{x} = x^2 + w$ and the state estimate $\hat{x}_k^+ = 0$. What is the differential equation for propagating $\hat{x}$ to the next measurement time using the first-order EKF? What is the differential equation using the second-order EKF?

Solution:

The first-order EKF uses the equation

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x}) \\ &= 0 \end{aligned}$$

The second-order EKF uses the equation

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x}) + P \\ &= \hat{x}^2 + P \end{aligned}$$

13.13 Consider the measurement equation $y_k = x_k^2 + v_k$, where $v_k \sim (0, R)$. Suppose that $P_k^- = 1$, and $\hat{x}_k^- = 1$ is unbiased.

a) What is the expected value of $\hat{x}_k^+$ if the first-order EKF is used for the measurement update? Based on your expression for $E(\hat{x}_k^+)$, how does the bias of the state estimate change with R? Does this make intuitive sense?

b) What is the expected value of $\hat{x}_k^+$ if the second-order EKF is used for the measurement update?

Solution:

a). Following the development in Section 13.3.2 we see that the first-order EKF gives

$$E(\hat{x}_k^+) = x_k - \pi_k$$

$$
\begin{aligned}
&= x_k - K_k D_k P_k^-/2 \\
&= x_k - \left(\frac{P_k^- H_k}{H_k^2 P_k^- + R} \right) (P_k^-)
\end{aligned}
$$

Substituting $P_k^- = 1$ and $H_k = 2$ into this equation gives

$$E(\hat{x}_k^+) = x_k - \frac{2}{4+R}$$

We see that the bias decreases as R increases. This is because more noise makes the nonlinearity of the measurement equation less significant.

b). Following the development in Section 13.3.2 we see that the second-order EKF gives

$$E(\hat{x}_k^+) = x_k$$

13.14 Consider the system

$$
\begin{aligned}
z_{k+1} &= a z_k + w_k, \qquad w_k \sim (0, Q) \\
y_k &= z_k + v_k, \qquad v_k \sim (0, R)
\end{aligned}
$$

with unknown parameter a. Suppose that an EKF is used to estimate the state z_k and the parameter a. Further suppose that the artificial noise term used in the estimation of a is zero, and the EKF converges to the correct value of a with zero variance. Show that the EKF in this situation is equivalent to the standard Kalman filter for the scalar system when a is known.

Solution:

The augmented state vector is $x_k = \begin{bmatrix} z_k & a \end{bmatrix}^T$. The F, H, and Q' matrices used in the EKF for parameter estimation are given as

$$
\begin{aligned}
F_k &= \frac{\partial f}{\partial x_k} \\
&= \begin{bmatrix} \hat{a}_k^- & \hat{z}_k^- \\ 0 & 1 \end{bmatrix} \\
Q' &= \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} \\
H &= \begin{bmatrix} 1 & 0 \end{bmatrix}
\end{aligned}
$$

Assuming that $\hat{a}_k^-$ has reached the steady-state value of a, and $P_k^+ = \mathrm{diag}(P_{k1}^+, 0)$, the EKF equations for P_k^- are

$$
\begin{aligned}
P_k^- &= \begin{bmatrix} P_{k1}^- & 0 \\ 0 & 0 \end{bmatrix} \\
&= F_k P_k^+ F_k^T + Q' \\
&= \begin{bmatrix} a^2 P_{k1}^+ + Q & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}
$$

The Kalman gain for the EKF is

$$\begin{aligned}
K_k &= \begin{bmatrix} K_{k1} \\ K_{k2} \end{bmatrix} \\
&= P_k^- H_k^T \left(H_k P_k^- H_k^T + R\right)^{-1} \\
&= \frac{1}{P_{k1}^- + R} \begin{bmatrix} P_{k1}^- \\ 0 \end{bmatrix}
\end{aligned}$$

The state estimate equations of the EKF (assuming $\hat{a}_k^-$ has converged to a) are given as

$$\begin{aligned}
\hat{x}_k^- &= \begin{bmatrix} \hat{z}_k^- \\ \hat{a}_k^- \end{bmatrix} \\
&= \begin{bmatrix} a\hat{z}_{k-1}^+ \\ a \end{bmatrix} \\
\hat{x}_k^+ &= \begin{bmatrix} \hat{z}_k^+ \\ \hat{a}_k^+ \end{bmatrix} \\
&= \hat{x}_k^- + K_k \left(y_k - \hat{z}_k^-\right) \\
&= \begin{bmatrix} \hat{z}_k^- + K_{k1}(y_k - \hat{z}_k^-) \\ \hat{a}_k^- \end{bmatrix}
\end{aligned}$$

The measurement update equation for the EKF covariance is

$$\begin{aligned}
P_k^+ &= \begin{bmatrix} P_{k1}^+ & 0 \\ 0 & 0 \end{bmatrix} \\
&= P_k^- - K_k H_k P_k^- \\
&= \begin{bmatrix} P_{k1}^- - K_{k1} P_{k1}^- & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned}$$

We see that the above equations for P_{k1}^-, K_{k1}, $\hat{z}_k^-$, $\hat{z}_k^+$, and P_{k1}^+ are the standard Kalman filter equations for the original system in the case that the parameter a is known.

Computer exercises

13.15 Write a program that implements the moving average filter and the extended Kalman filter for the system described in Problem 13.3. Use $R = 1$, $x = 1$, $P_0^+ = 1$, and $\hat{x}_0 = 2$. Which filter appears to perform better?

Solution:

See Figure 13.1. It appears that the Kalman filter performs much better than the moving average filter. (Recall that the true value of $x = 1$.)

13.16 A planar model for a satellite orbiting around the earth can be modeled as

$$\ddot{r} = r\dot{\theta}^2 - \frac{GM}{r^2} + w$$

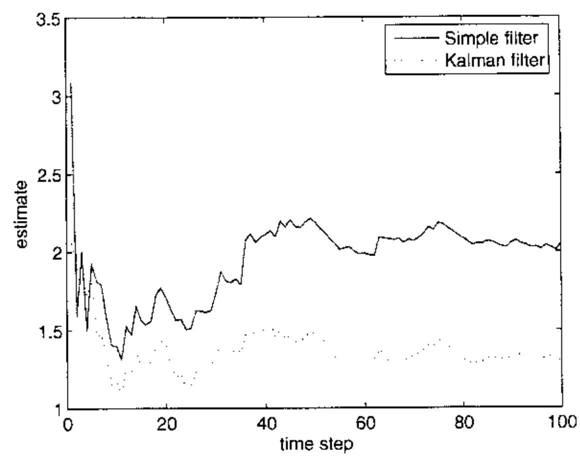

Figure 13.1 Solution to Problem 13.15

$$\ddot{\theta} = \frac{-2\dot{\theta}\dot{r}}{r}$$

where r is the distance of the satellite from the center of the earth, θ is the angular position of the satellite in its orbit, $G = 6.6742 \times 10^{-11}\text{m}^3/\text{kg/s}^2$ is the universal gravitational constant, $M = 5.98 \times 10^{24}$ kg is the mass of the earth. and $w \sim (0, 10^{-6})$ is random noise due to space debris, atmospheric drag, outgassing, and so on.

a) Write a state-space model for this system with $x_1 = r$, $x_2 = \dot{r}$, $x_3 = \theta$, and $x_4 = \dot{\theta}$.

b) What must $\dot{\theta}$ be equal to in order for the orbit to have a constant radius when $w = 0$?

c) Linearize the model around the point $r = r_0$, $\dot{r} = 0$, $\theta = \omega_0 T$, $\dot{\theta} = \omega_0$. What are the eigenvalues of the system matrix for the linearized system when $r_0 = 6.57 \times 10^6$ m? What would you estimate to be the largest integration step size that could be used to simulate the system? (Hint: recall that for a second-order transfer function with imaginary poles $\pm ja$, the time constant is equal to $1/a$.)

d) Suppose that measurements of the satellite radius and angular position are obtained every minute, with error standard deviations of 100 meters and 0.1 radians, respectively. Simulate the linearized Kalman filter for three hours. Initialize the system with $x(0) = \begin{bmatrix} r_0 & 0 & 0 & 1.1\omega_0 \end{bmatrix}$, $\hat{x}(0) = x(0)$, and $P(0) = \text{diag}(0, 0, 0, 0)$. Plot the radius estimation error as a function of time. Why is the performance so poor? How could you modify the linearized Kalman filter to get better performance?

e) Implement an extended Kalman filter and plot the radius estimation error as a function of time. How does the performance compare with the linearized Kalman filter?

Solution:

a). The state-space description is

$$\begin{aligned} \dot{x} &= f(x) \\ \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} &= \begin{bmatrix} x_2 \\ x_1 x_4^2 - \frac{GM}{x_1^2} + w \\ x_4 \\ \frac{-2x_2 x_4}{x_1} \end{bmatrix} \end{aligned}$$

b). In order for the orbit to have a constant radius when $w = 0$, we have $\ddot{r} = r\dot{\theta}^2 - GM/r^2 = 0$, which means $\dot{\theta} = \sqrt{GM/r^3}$.

c). The linearized system matrix is given as

$$\begin{aligned} F &= \frac{\partial f}{\partial x} \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ x_4^2 + 2GM/x_1^3 & 0 & 0 & 2x_1 x_4 \\ 0 & 0 & 0 & 1 \\ 2x_2 x_4/x_1^2 & -2x_4/x_1 & 0 & 2x_2/x_1 \end{bmatrix} \end{aligned}$$

Note that at the linearization point we have $\dot{r} = 0$, which means $GM = r_0^3 \omega_0^2$. Evaluating F at the linearization point gives

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega_0^2 & 0 & 0 & 2r_0\omega_0 \\ 0 & 0 & 0 & 1 \\ 0 & -2\omega_0/r_0 & 0 & 0 \end{bmatrix}$$

The eigenvalues of this matrix at the linearization point are 0, 0, $\pm j0.0012$. This gives a time constant of $1/0.0012 = 833$ seconds. This means an integration step size of 100 seconds should give adequate simulation results.

d). Figure 13.2 shows the radius estimation error for the linearized Kalman filter. The performance is poor because Q is small and $P(0) = 0$, so the filter practically ignores the measurements and simply extrapolates the linear model. We could get a better estimate of radius by simply processing the radius measurements directly (recall the radius measurements have a standard deviation of 100 m). This would be equivalent to using artificial noise (a larger Q matrix) in the linearized Kalman filter.

e). Figure 13.3 shows the radius estimation error for the extended Kalman filter. Performance is orders of magnitude better than the linearized Kalman filter.

13.17 Implement the hybrid EKF with a measurement period of $0.1s$ for the system described in Example 13.1. Assume that the winding current measurement noises have a standard deviation of 0.1 amps. Create a table showing the experimental standard deviation of the motor velocity estimation error as a function of the standard deviation of the control input uncertainties q_1 and q_2. Use control input standard deviations from 0 to 0.1 volts in steps of 0.01 (i.e., $\sigma_q = 0$, $\sigma_q = 0.01$,

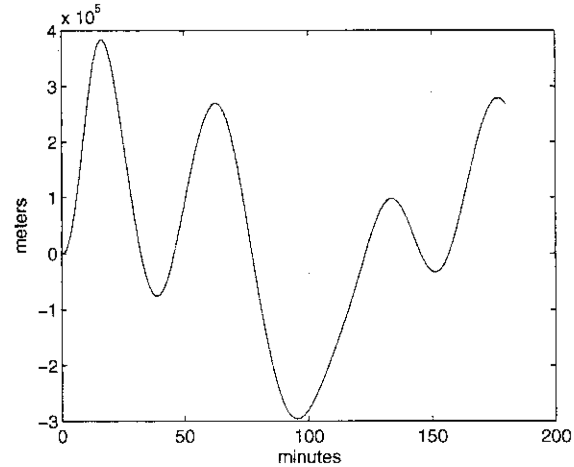

Figure 13.2 Solution to Problem 13.16. Linearized Kalman filter radius estimation error

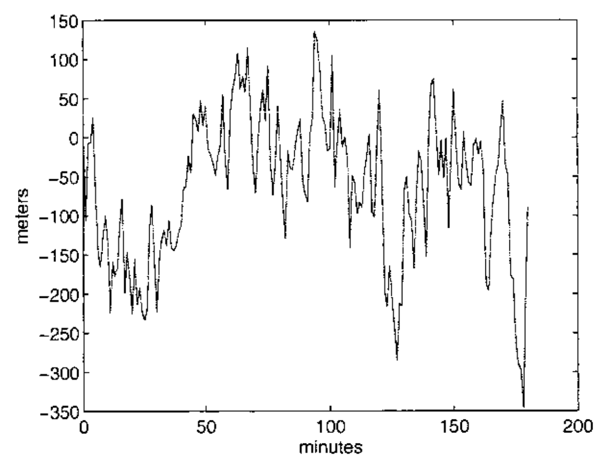

Figure 13.3 Solution to Problem 13.16. Extended Kalman filter radius estimation error

$\cdots$, $\sigma_q = 0.1$). In order to make a fair comparison, you should either run several simulations for each value of σ_q and average the results, or else initialize the random seed in your software so that each simulation runs with the same random noise history.

Solution:

13.18 Derive the first-order EKF, second-order EKF, and iterated EKF (with one iteration) for the scalar system

$$x_{k+1} = x_k^2 + w_k$$

Table 13.1 Problem 13.17 Solution. Velocity estimation error (one standard deviation) as a function of control input noise (one standard deviation). Your results may vary depending on your noise history.

Control input noise (volts)	Velocity estimation error (rad/sec)
0	0.002
0.01	0.042
0.02	0.16
0.03	0.35
0.04	0.58
0.05	0.83
0.06	1.09
0.07	1.35
0.08	1.59
0.09	1.82
0.10	2.04

$$y_k = x_k^2 + v_k$$

where w_k and v_k are independent zero-mean white noise terms with variances 0.1 and 1, respectively. Simulate the first-order, second-order, and iterated extended Kalman filters for five time steps. Set the initial state to 1, the initial estimation-error variance to 1, and the initial state estimate to 2. Compute the RMS error of the filter estimates. How does the performance of the filters compare? (Note that you need more than one simulation, in general, to obtain a fair comparison of filter performance.)

Solution:

I ran 100 simulations. The average RMS estimation errors of the filters were 10.1, 0.6, and 2.6 for the first-order, second-order, and iterated EKFs. The first-order EKF performed best in 7 out of the 100 simulations, the second-order EKF performed best in 90 out of the 100 simulations, and the iterated EKF performed best in 3 out of the 100 simulations.

The second-order EKF outperformed the first-order EKF in 93 of the simulations, and the iterated EKF outperformed the first-order EKF in 87 of the simulations.

13.19 Use the following procedure [Sor71b] to approximate a uniform pdf that is defined on ± 1 with M Gaussian pdfs; that is, $U(-1,1) \approx \sum_{i=1}^{M} a_i N(\mu_i, \sigma_i^2)$.

- Select the weighting coefficients so that $a_i = 1/M$ for all i.

- Select the means of the Gaussian pdfs to be equally spaced on the range $[-1, 1]$ with $\mu_{i+1} - \mu_i = 2/(M+1)$.

- Select the variances σ_i of the Gaussian pdfs to all be the same and to minimize the RMS difference between $U(-1,1)$ and $\sum_{i=1}^{M} a_i N(\mu_i, \sigma_i^2)$ over the range $[-1,1]$.

The above approach reduces the approximation problem to a one-dimensional optimization problem, which can be solved in a number of different ways (for example, using the golden search method [Pre92]). Plot the true pdf and the approximate pdf for $M = 3$, 5, and 10, and compare the RMS errors.

Solution:

a). $a_i = 1/M$ $(i = 1, \cdots, M)$.

b).

$$\mu_i = -1 + \frac{2i}{M+1} \qquad (i = 1, \cdots, M)$$

c). Using a golden search method we find

$$\sigma_i = \begin{cases} 0.430 & (M=3) \\ 0.444 & (M=5) \\ 0.461 & (M=10) \end{cases}$$

Figure 13.4 shows the pdf approximation for different values of M.

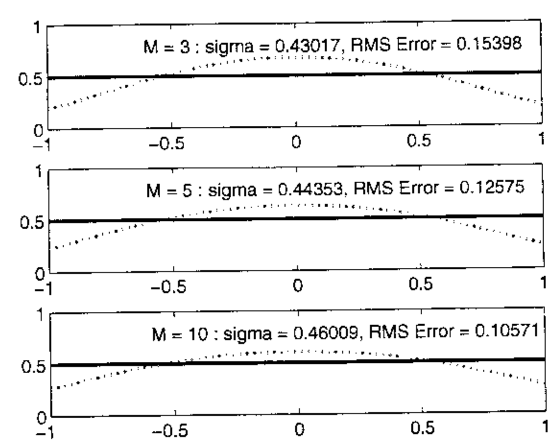

Figure 13.4 Uniform pdf approximation by M Gaussian pdfs

13.20 Suppose you have a scalar system given as

$$\begin{aligned} x_{k+1} &= x_k \\ y_k &= x_k^2 + v_k \end{aligned}$$

where v_k is white Gaussian noise with a variance of 0.01. The pdf of the initial state x_0 is uniform between -1 and $+1$. Note from the measurement equation that there is no way to distinguish between a positive state and a negative state.

a) What will the extended Kalman filter estimate of the system be equal to?

b) The pdf of x_0 can be approximated with two Gaussian pdfs, each with a variance of 0.43, and with respective means of $-1/3$ and $+1/3$. Suppose that $x_0 = -1/2$. Plot the true state and the individual state estimates of a two-term Gaussian sum filter for 20 time steps. Plot the Gaussian pdfs at the final time for each estimate of the two-term Gaussian sum filter.

Solution:

a). The EKF estimate of the state will be equal to zero for all time.

b). See Figure 13.5.

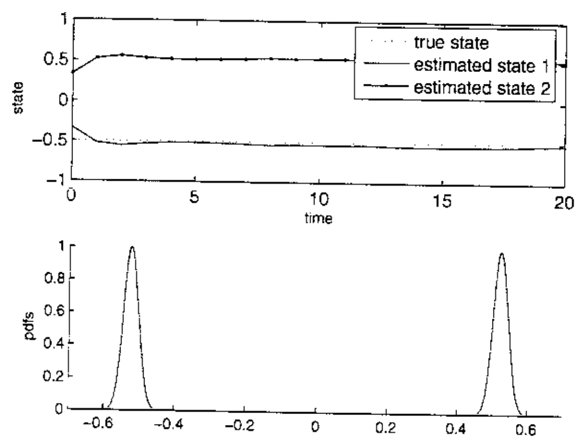

Figure 13.5 (a) True state and two state estimates of the two-term Gaussian sum filter; (b) pdfs of the two state estimates at of the two-term Gaussian sum filter (at time step 20)

13.21 Consider the problem of tracking a moving vehicle in two dimensions (north is one dimension and east is the other dimension). The vehicle's acceleration in the north and east directions consists of independent white noise. Two tracking stations, located at north–east coordinates (N_1, E_1) and (N_2, E_2), respectively, measure the range to the vehicle. The system model can therefore be written as

$$\begin{bmatrix} n_{k-1} \\ e_{k+1} \\ \dot{n}_{k-1} \\ \dot{e}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_k \\ e_k \\ \dot{n}_k \\ \dot{e}_k \end{bmatrix} + w_k$$

$$y_k = \begin{bmatrix} \sqrt{(n_k - N_1)^2 + (e_k - E_1)^2} \\ \sqrt{(n_k - N_2)^2 + (e_k - E_2)^2} \end{bmatrix} + v_k$$

where n_k and e_k are the vehicle's north and east coordinates at time step k, T is the time step of the system, w_k is the zero-mean process noise, and v_k is the zero-mean measurement noise. Suppose that the time step $T = 0.1s$, the process noise covariance $Q = \text{diag}(0, 0, 4, 4)$, and the measurement noise covariance $R = \text{diag}(1, 1)$. The tracking stations are located at $(N_1, E_1) = (20, 0)$, and $(N_2, E_2) = (0, 20)$. The initial state of the vehicle $x_0 = \begin{bmatrix} 0 & 0 & 50 & 50 \end{bmatrix}^T$ and is perfectly known. Design an extended Kalman filter to estimate the state of the vehicle. Run the simulation for 60 s. Plot the estimation error for the four states. What is the experimental standard deviation of the estimation error for each of the four states? Based on the steady-state covariance matrix of the filter, what is the theoretical standard deviation of the estimation error for each of the four states?

Solution:

Actually the covariance matrix that comes out of the EKF does not (in general) converge to a steady-state value. Figure 13.6 shows typical estimation errors. The standard deviations of the estimation error are typically between 3 and 30 for the two position states, and between 4 and 10 for the two velocity states, depending on the particular noise history that is realized.

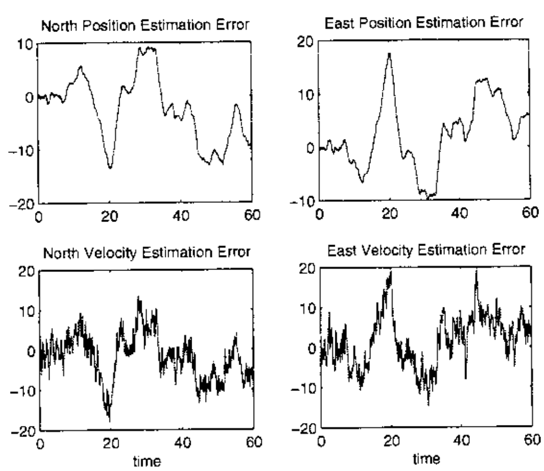

Figure 13.6 Solution to Problem 13.21

13.22 Consider the system

$$\begin{aligned} x_{k+1} &= \phi x_k + w_k \\ y_k &= x_k \end{aligned}$$

where $w_k \sim (0, 1)$, and $\phi = 0.9$ is an unknown constant. Design an extended Kalman filter to estimate ϕ. Simulate the filter for 100 time steps with $x_0 = 1$, $P_0 = I$, $\hat{x}_0 = 0$, and $\hat{\phi}_0 = 0$. Hand in your source code and a plot showing $\hat{\phi}$ as a function of time.

Solution:

The augmented state is defined as $X_k = \begin{bmatrix} x_k & \phi_k \end{bmatrix}^T$. The EKF equations are

$$\begin{aligned}
Q &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\
F_{k-1} &= \begin{bmatrix} \hat{\phi}_k^+ & \hat{x}_k^+ \\ 0 & 1 \end{bmatrix} \\
P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + Q \\
\hat{x}_k^- &= \hat{\phi}_{k-1}^+ \hat{x}_{k-1}^+ \\
\hat{\phi}_k^- &= \hat{\phi}_{k-1}^+ \\
\hat{X}_k^- &= \begin{bmatrix} \hat{x}_k^- & \hat{\phi}_k^- \end{bmatrix}^T \\
H_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T)^{-1} \\
\hat{X}_k^+ &= \hat{X}_k^- + K_k (y_k - \hat{x}_k^-) \\
P_k^+ &= (I - K_k H_k) P_k^-
\end{aligned}$$

Simulation results are shown in Figure 13.7

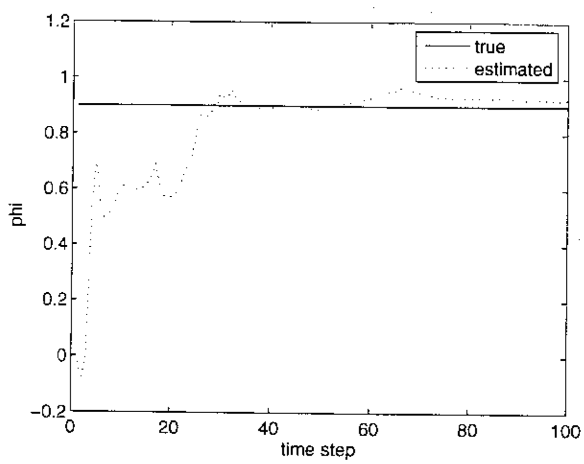

Figure 13.7 Solution to Problem 13.22

13.23 Simulate Example 13.4 with artificial parameter noise variance values $\sigma_p^2 = 0$, 1, and 100. How does a change in the artificial parameter noise variance affect the filter's estimate of $-\omega_n^2$?

Solution:

Figure 13.8 shows an example of parameter estimation convergence for different values of σ_p^2. As σ_p^2 increases, the parameter estimate tends to converge to the true parameter value more quickly, but the estimate also tends to be more noisy.

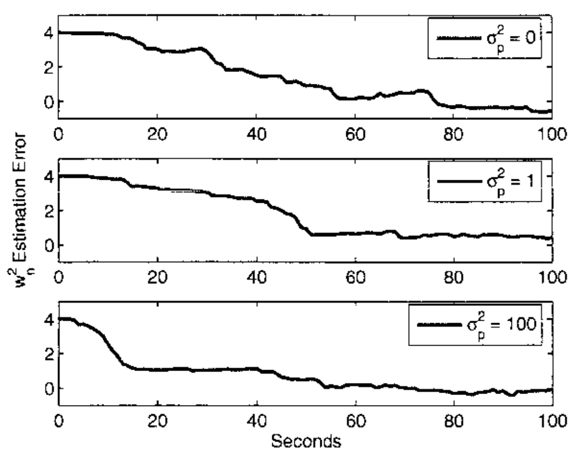

Figure 13.8 Solution to Problem 13.23

CHAPTER 14

The unscented Kalman filter

Problems

Written exercises

14.1 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = x^2$. What is $\bar{y}$? What is the first-order approximation to $\bar{y}$? What is the second-order approximation to $\bar{y}$?

Solution:

We have $y = h(x)$, where $h(x) = x^2$. The mean of y is

$$\begin{aligned}
\bar{y} &= E(x^2) \\
&= \int x^2 \text{pdf}(x)\, dx \\
&= \frac{1}{2}\int_{-1}^{1} x^2\, dx \\
&= 1/3
\end{aligned}$$

The first-order approximation to $\bar{y}$ is

$$\begin{aligned}\bar{y}_1 &= h(\bar{x}) \\ &= 0\end{aligned}$$

The second-order approximation to $\bar{y}$ is

$$\begin{aligned}\bar{y}_2 &= h(\bar{x}) + \frac{1}{2}E\left(D_{\tilde{x}}^2 h\right) \\ &= \frac{1}{2}E\left[\left(\tilde{x}\frac{\partial}{\partial x}\right)^2 h(x)\bigg|_{\bar{x}}\right] \\ &= \frac{1}{2}E\left(\tilde{x}^2\frac{\partial^2 h(x)}{\partial x^2}\bigg|_{\bar{x}}\right) \\ &= E(x^2) \\ &= 1/3\end{aligned}$$

14.2 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is $\bar{y}$? What is the first-order approximation to $\bar{y}$? What is the second-order approximation to $\bar{y}$? What is the third-order approximation to $\bar{y}$? What is the fourth-order approximation to $\bar{y}$?

Solution:

We have $y = h(x)$, where $h(x) = e^x$. The mean of y is

$$\begin{aligned}\bar{y} &= E(e^x) \\ &= \int e^x \,\mathrm{pdf}(x)\, dx \\ &= \frac{1}{2}\int_{-1}^{1} e^x\, dx \\ &= (e - e^{-1})/2 \\ &\approx 1.1752\end{aligned}$$

The first-order approximation to $\bar{y}$ is

$$\begin{aligned}\bar{y}_1 &= h(\bar{x}) \\ &= 1\end{aligned}$$

The second-order approximation to $\bar{y}$ is

$$\begin{aligned}\bar{y}_2 &= h(\bar{x}) + \frac{1}{2}E\left(D_{\tilde{x}}^2 h\right) \\ &= 1 + \frac{1}{2}E\left[\left(\tilde{x}\frac{\partial}{\partial x}\right)^2 h(x)\bigg|_{\bar{x}}\right] \\ &= 1 + \frac{1}{2}E\left(\tilde{x}^2\frac{\partial^2 h(x)}{\partial x^2}\bigg|_{\bar{x}}\right) \\ &= 1 + E(x^2)/2 \\ &= 7/6 \\ &\approx 1.1667\end{aligned}$$

The third-order approximation to $\bar{y}$ is the same as the second order approximation.

$$\begin{aligned}\bar{y}_3 &= 7/6 \\ &\approx 1.1667\end{aligned}$$

The fourth-order approximation to $\bar{y}$ is

$$\begin{aligned}\bar{y}_4 &= h(\bar{x}) + \frac{1}{2}E\left(D_{\tilde{x}}^2 h\right) + \frac{1}{24}E\left(D_{\tilde{x}}^4 h\right) \\ &= 1 + E(x^2)/2 + E(x^4)/24 \\ &= 141/120 \\ &\approx 1.1750\end{aligned}$$

14.3 Suppose the RV x is uniformly distributed on $[-1,1]$, and $y = e^x$. What is the variance of y? What is the first-order approximation to the variance of y? What is the fourth-order approximation to the variance of y?

Solution:

We have $y = h(x)$, where $h(x) = e^x$. The variance of y is

$$\begin{aligned}P_y &= E[(y-\bar{y})^2] \\ &= E[x^4 - 2\bar{y}x^2 + \bar{y}^2] \\ &= \frac{1}{2}\int_{-1}^{1}(x^4 - 2\bar{y}x^2)\,dx + \bar{y}^2 \\ &= 1/5 - 2\bar{y}/3 + \bar{y}^2 \\ &\approx 0.7976\end{aligned}$$

(Note that $\bar{y}$ was computed in Problem 14.2.) The first-order approximation to P_y is computed as

$$\begin{aligned}P_{y1} &= HP_xH^T \\ &= P_x \\ &= 1/3 \\ &\approx 0.3333\end{aligned}$$

The fourth-order approximation to P_y is obtained from Equation (14.23) as

$$\begin{aligned}P_{y4} &= HP_xH^T + E\left[x^4/6 + x^4/4 + x^4/6\right] - E\left(x^2/2\right)E\left(x^2/2\right) \\ &= 1/3 + 7/60 - 1/36 \\ &\approx 0.4222\end{aligned}$$

14.4 Suppose the RV x is uniformly distributed on $[-1,1]$, and $y = e^x$. What is $\bar{y}$? What is the unscented approximation to $\bar{y}$?

Solution:

$\bar{y}$ was obtained in Problem 14.2 as

$$\begin{aligned} \bar{y} &= (e - e^{-1})/2 \\ &\approx 1.1752 \end{aligned}$$

We can obtain $P = E(x^2) = 1/3$. Then we obtain

$$\begin{aligned} \tilde{x}^{(1)} &= \sqrt{1/3} \\ \tilde{x}^{(2)} &= -\sqrt{1/3} \end{aligned}$$

Then we obtain the sigma points $x^{(i)} = \bar{x} + \tilde{x}^{(i)}$ as

$$\begin{aligned} x^{(1)} &= \sqrt{1/3} \\ x^{(2)} &= -\sqrt{1/3} \end{aligned}$$

The transformed sigma points $y^{(i)} = h(x^{(i)})$ are obtained as

$$\begin{aligned} y^{(1)} &= \exp\left(\sqrt{1/3}\right) \\ y^{(2)} &= \exp\left(-\sqrt{1/3}\right) \end{aligned}$$

The unscented approximation to $\bar{y}$ is finally obtained as

$$\begin{aligned} \bar{y}_u &= \frac{1}{2}\left[\exp\left(\sqrt{1/3}\right) + \exp\left(-\sqrt{1/3}\right)\right] \\ &\approx 1.1713 \end{aligned}$$

14.5 Consider the matrix

$$P = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$$

Find an upper triangular matrix S (using only paper and pencil) such that $S^T S = P$. Find a lower triangular matrix S such that $S^T S = P$. (Note the difference between your solution to this problem and the solution to Problem 6.7.)

Solution:

If S is upper triangular then

$$\begin{aligned} S^T S &= \begin{bmatrix} S_{11} & 0 \\ S_{12} & S_{22} \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} \\ &= \begin{bmatrix} S_{11}^2 & S_{11}S_{12} \\ S_{11}S_{12} & S_{12}^2 + S_{22}^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix} \end{aligned}$$

Equating the elements of the matrices on right sides of the above equations gives $S_{11} = 1$, $S_{12} = 3$, and $S_{22} = 0$.

$$S = \begin{bmatrix} 1 & 3 \\ 0 & 0 \end{bmatrix}$$

Repeating for lower triangular S gives

$$S = \begin{bmatrix} 0 & 0 \\ 1 & 3 \end{bmatrix}$$

14.6 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is the variance of y? What is unscented approximation to the variance of y?

Solution:

In Problem 14.3 we obtained the variance of y as

$$\begin{aligned} P_y &= 1/5 - 2\bar{y}/3 + \bar{y}^2 \\ &\approx 0.7976 \end{aligned}$$

The unscented approximation is obtained from Equation (14.43) as follows.

$$\begin{aligned} P_u &= \frac{1}{2}\left[\left(y^{(1)} - \bar{y}_u\right)^2 + \left(y^{(2)} - \bar{y}_u\right)^2\right] \\ &= \frac{1}{2}\left[\frac{1}{4}\left(\exp\left(\sqrt{1/3}\right) - \exp\left(-\sqrt{1/3}\right)\right)^2 + \frac{1}{4}\left(\exp\left(-\sqrt{1/3}\right) - \exp\left(\sqrt{1/3}\right)\right)^2\right] \\ &= \frac{1}{4}\left[\exp\left(\sqrt{1/3}\right) - \exp\left(-\sqrt{1/3}\right)\right]^2 \\ &\approx 0.3721 \end{aligned}$$

where $y^{(1)}$, $y^{(2)}$, and $\bar{y}_u$ were obtained in Problem 14.4.

14.7 Show that for a system with an identity transition matrix, the UKF algorithm gives $\hat{x}_k^- = \hat{x}_{k-1}^+$.

Solution:

From Equations (14.58)–(14.60) and $f(x) = x$ we obtain

$$\begin{aligned} \hat{x}_k^- &= \frac{1}{2n}\left[\sum_{i=1}^{n} \hat{x}_{k-1}^+ + \left(\sqrt{nP_{k-1}^+}\right)_i + \sum_{i=1}^{n} \hat{x}_{k-1}^+ - \left(\sqrt{nP_{k-1}^+}\right)_i\right] \\ &= \frac{1}{2n}\sum_{i=1}^{2n} \hat{x}_{k-1}^+ \\ &= \hat{x}_{k-1}^+ \end{aligned}$$

QED

14.8 Show that for a system with $y_k = x_k$, the UKF gain K_k is positive definite.

Solution:

First define

$$P_x = \frac{1}{2n}\sum_{i=1}^{2n}(\hat{x}_k^{(i)} - \hat{x}_k^-)(\hat{x}_k^{(i)} - \hat{x}_k^-)^T$$

where $\hat{x}_k^{(i)}$ is defined in Equation (14.62). Note that P_x as defined above is positive definite. With $y_k = x_k$ we see from Equations (14.63)–(14.67) that

$$\begin{aligned} P_{xy} &= P_x \\ P_y &= P_x + R \\ K_k &= P_x (P_x + R)^{-1} \end{aligned}$$

Since both P_x and R are positive definite, K_k is also positive definite.
QED

14.9 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is $\bar{y}$? Use the generalized unscented transformation to approximate $\bar{y}$ with $\kappa = 0$, $\kappa = 1$, and $\kappa = 2$.

Solution:

In Problem 14.4 we obtained

$$\begin{aligned} \bar{y} &\approx 1.1752 \\ P &= 1/3 \end{aligned}$$

The unscented approximation to $\bar{y}$ is obtained as

$$\begin{aligned} \bar{y}_u &= \frac{\kappa}{\kappa+1} + \frac{1}{2(\kappa+1)} \left[\exp\left(\sqrt{(\kappa+1)/3}\right) + \exp\left(-\sqrt{(\kappa+1)/3}\right) \right] \\ &\approx \begin{cases} 1.1713 & (\kappa = 0) \\ 1.1761 & (\kappa = 1) \\ 1.1810 & (\kappa = 2) \end{cases} \end{aligned}$$

14.10 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is the variance of y? Use the generalized unscented transformation to approximate the variance of y with $\kappa = 0$, $\kappa = 1$, and $\kappa = 2$.

Solution:

In Problem 14.9 we obtained $\bar{y}_u$ for $\kappa = 0$, 1, and 2. In Problem 14.3 we obtained $P_y \approx 0.7976$. The generalized UKF approximation to the variance of y is

$$\begin{aligned} P_u &= \frac{\kappa}{\kappa+1} (1 - \bar{y}_u)^2 + \\ &\quad \frac{1}{2(\kappa+1)} \left[\left(\exp\left(\sqrt{(\kappa+1)/3}\right) - \bar{y}_u \right)^2 + \left(\exp\left(-\sqrt{(\kappa+1)/3}\right) - \bar{y}_u \right)^2 \right] \\ &= \begin{cases} 0.3721 & (\kappa = 0) \\ 0.4453 & (\kappa = 1) \\ 0.5259 & (\kappa = 2) \end{cases} \end{aligned}$$

14.11 Consider the simplex sigma-point algorithm. Prove that $\sum_i W^{(i)} \sigma_i^{(j)} = 0$ (i.e., the weighted sample mean of the $\sigma_i^{(j)}$ vectors is zero).

Solution:

For $j = 1$ we have

$$\begin{aligned}\sum_{i=0}^{2} W^{(i)}\sigma_i^{(1)} &= W^{(0)}0 - \frac{W^{(1)}}{\sqrt{2W^{(1)}}} + \frac{W^{(1)}}{\sqrt{2W^{(1)}}} \\ &= 0\end{aligned}$$

For $j = 2$ we have

$$\begin{aligned}\sum_{i=0}^{3} W^{(i)}\sigma_i^{(2)} &= W^{(0)}0 + W^{(1)}\begin{bmatrix} \frac{-1}{\sqrt{2W^{(1)}}} \\ \frac{-1}{\sqrt{2W^{(3)}}} \end{bmatrix} + W^{(1)}\begin{bmatrix} \frac{1}{\sqrt{2W^{(1)}}} \\ \frac{-1}{\sqrt{2W^{(3)}}} \end{bmatrix} + 2W^{(1)}\begin{bmatrix} 0 \\ \frac{1}{\sqrt{2W^{(3)}}} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}\end{aligned}$$

For $j = 3$ we have

$$\begin{aligned}\sum_{i=0}^{4} W^{(i)}\sigma_i^{(3)} &= W^{(0)}0 + W^{(1)}\begin{bmatrix} \frac{-1}{\sqrt{2W^{(1)}}} \\ \frac{-1}{\sqrt{2W^{(3)}}} \\ \frac{-1}{\sqrt{2W^{(4)}}} \end{bmatrix} + W^{(1)}\begin{bmatrix} \frac{1}{\sqrt{2W^{(1)}}} \\ \frac{-1}{\sqrt{2W^{(3)}}} \\ \frac{-1}{\sqrt{2W^{(4)}}} \end{bmatrix} + \\ &\quad 2W^{(1)}\begin{bmatrix} 0 \\ \frac{1}{\sqrt{2W^{(3)}}} \\ \frac{-1}{\sqrt{2W^{(4)}}} \end{bmatrix} + 4W^{(1)}\begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2W^{(4)}}} \end{bmatrix} \\ &= 0\end{aligned}$$

In general we see that

$$\begin{aligned}\bar{\sigma}^{(j)} &= \sum_{i=0}^{j+1} W^{(i)}\sigma_i^{(j)} \\ &= \begin{bmatrix} \bar{\sigma}^{(j-1)} \\ c \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ c \end{bmatrix}\end{aligned}$$

where c is given as

$$\begin{aligned}c &= \frac{-W^{(1)}}{\sqrt{2W^{(j+1)}}} + \frac{-W^{(1)}}{\sqrt{2W^{(j+1)}}} + \frac{-2W^{(1)}}{\sqrt{2W^{(j+1)}}} + \frac{-4W^{(1)}}{\sqrt{2W^{(j+1)}}} + \cdots + \frac{2^{j-1}W^{(1)}}{\sqrt{2W^{(j+1)}}} \\ &= \frac{W^{(1)}}{\sqrt{2W^{(j+1)}}}(-1 - 1 - 2 - 4 - \cdots - 2^{j-2} + 2^{j-1}) \\ &= 0\end{aligned}$$

QED

14.12 Prove that the sum of the weights in the simplex sigma-point algorithm is equal to 1.

Solution:

$$\begin{aligned}\sum_{i=0}^{n+1} W^{(i)} &= W^{(0)} + 2^{-n}(1-W^{(0)}) + 2^{-n}(1-W^{(0)}) + \sum_{i=3}^{n+1} 2^{i-2}2^{-n}(1-W^{(0)}) \\ &= W^{(0)} + 2^{1-n}(1-W^{(0)}) + 2^{-n}(1-W^{(0)})\sum_{i=1}^{n-1} 2^i \\ &= W^{(0)} + 2^{1-n}(1-W^{(0)}) + 2^{-n}(1-W^{(0)})(2^n-2) \\ &= W^{(0)} + 2^{1-n}(1-W^{(0)}) + (1-2^{1-n})(1-W^{(0)}) \\ &= W^{(0)} + (1-W^{(0)}) \\ &= 1\end{aligned}$$

QED

14.13 Consider the simplex sigma-point algorithm. Prove that the $\sum_i W^{(i)}x^{(i)} = \bar{x}$ (i.e., the weighted sample mean of the sigma points is equal to $\bar{x}$). (Hint: Use the results of Problems 14.11 and 14.12.)

Solution:

The sample mean of the $x^{(i)}$ vectors is found as

$$\begin{aligned}\sum_{i=0}^{n+1} W^{(i)}x^{(i)} &= \sum_{i=0}^{n+1} W^{(i)}\left(\bar{x} + \sqrt{P}\sigma_i^{(n)}\right) \\ &= \bar{x}\sum_{i=0}^{n+1} W^{(i)} + \sqrt{P}\sum_{i=0}^{n+1} W^{(i)}\sigma_i^{(n)}\end{aligned}$$

In Problem 14.11 we showed that the weighted sample mean of the $\sigma_i^{(n)}$ vectors is zero, and in Problem 14.12 we showed that the $W^{(i)}$ weights sum up to one. Therefore

$$\sum_{i=0}^{n+1} W^{(i)}x^{(i)} = \bar{x}$$

QED

Computer exercises

14.14 Design an unscented Kalman filter for the system described in Problem 13.21. Simulate the system and the filter for 60 s. Plot the estimation error for the four states. What is the experimental standard deviation of the estimation error for each of the four states? Based on the steady-state covariance matrix of the filter, what is the theoretical standard deviation of the estimation error for each of the four states? How does this compare with the extended Kalman filter results of Problem 13.21?

Solution:

Actually the covariance matrix that comes out of the UKF does not (in general) converge to a steady-state value. Figure 14.1 shows typical estimation errors. The standard deviations of the estimation error are typically between 10 and 30 for the two position states, and between 5 and 10 for the two velocity states, depending on the particular noise history that is realized. The UKF offers about the same performance as the EKF for this example. The nonlinearities in this system are not too severe, so the UKF does not offer an appreciable advantage over the EKF.

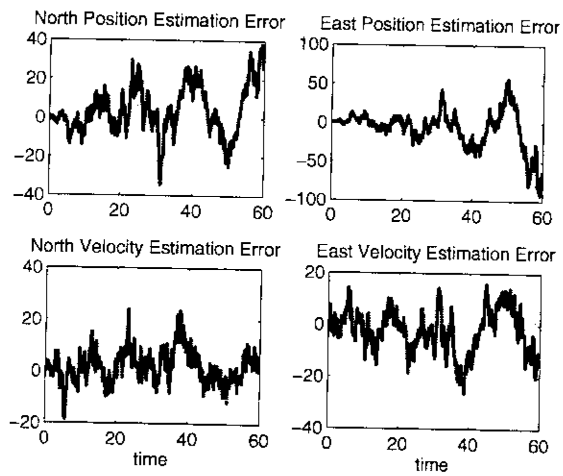

Figure 14.1 Solution to Problem 14.14

14.15 An inverted pendulum on a cart can be modeled as follows [Bay99, Che99].

$$\begin{aligned}\ddot{\theta} &= \frac{mgl\sin\theta(M+m) - ml\cos\theta(u + ml\dot{\theta}^2\sin\theta - B\dot{d})}{(J+ml^2)(M+m) - m^2l^2\cos^2\theta} \\ \ddot{d} &= \frac{u - ml\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta - B\dot{d}}{M+m}\end{aligned}$$

The quantities in the system model are as follows:

$$\begin{aligned}\theta(0) &= \text{initial angle (0.1 rad)} \\ d(0) &= \text{initial cart displacement (0 rad)} \\ m &= \text{pendulum mass (0.2 kg)} \\ M &= \text{cart mass (1 kg)} \\ g &= \text{acceleration due to gravity (9.81 m/s}^2\text{)} \\ B &= \text{coefficient of friction between cart and ground [0.1 N/(m/s)]} \\ l &= \text{pendulum length (1 m)} \\ r &= \text{pendulum mass radius (0.02 m)} \\ u &= \text{external force applied to cart}\end{aligned}$$

$$\begin{aligned} J &= \text{pendulum moment of inertia} \\ &= mr^2/2 \end{aligned}$$

where we have assumed that the pendulum mass is concentrated in a cylinder at the end of the pendulum. Define the state of the system as $x = \begin{bmatrix} d & \dot{d} & \theta & \dot{\theta} \end{bmatrix}^T$. The horizontal displacement d is measured every 5 ms with a standard deviation of 0.1 m. The continuous-time process noise is $Q_c = \text{diag}(0, 0.0004, 0, 0.04)$. The system can be linearized (so that an EKF can be used to estimate the state) by assuming that θ is small, so $\cos\theta \approx 1$, $\sin\theta \approx \theta$, and $\dot{\theta}^2 \approx 0$. Suppose that the feedback control signal is given as $u = 40\theta$ and the initial state is perfectly known. Write an EKF and a UKF to estimate the state, where the control is assumed by the filters to be $\hat{u} = 40\hat{\theta}$. Plot the true states and estimated states for a 2 second simulation. Which filter appears to perform better?

Solution:

The linearized equations are

$$\begin{aligned} \left[(J + ml^2)(M + m) - m^2l^2\right]\ddot{\theta} &= mgl\theta(M + m) - ml(u - B\dot{d}) \\ (M + m)\ddot{d} &= u - ml\ddot{\theta} - B\dot{d} \end{aligned}$$

These equations can be used to obtain the F matrix for the EKF. Figure 14.2 shows the true states and the UKF estimates. The UKF estimates are pretty good, but the EKF diverges (not shown in the figure). However, if the EKF knows the true control $u = 40\theta$, then the EKF does not diverge, although its performance is still much worse than the UKF.

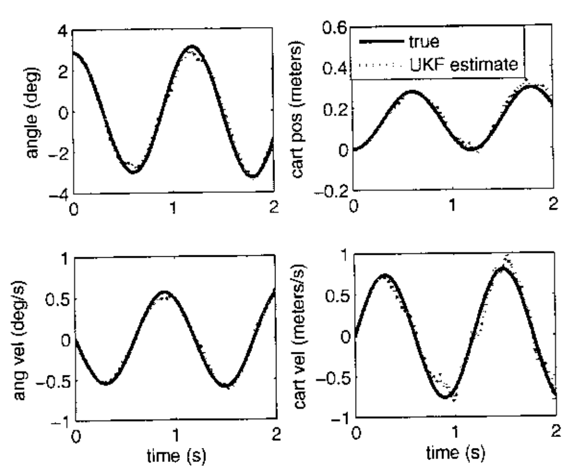

Figure 14.2 Solution to Problem 14.15

CHAPTER 15

The particle filter

Problems

Written exercises

15.1 Consider the scalar system

$$\begin{aligned} x_{k+1} &= x_k + w_k, \quad w_k \sim U(-1,1) \\ y_k &= x_k + v_k, \quad v_k \sim U(-1,1) \end{aligned}$$

where $x_0 \sim U(-1,1)$. Suppose that the first measurement $y_1 = 1$.

a) Use the recursive Bayesian state estimator to find $\text{pdf}(x_1|Y_0)$ and $\text{pdf}(x_1|Y_1)$.

b) What is the Kalman filter estimate $\hat{x}_1^+$? How is $\hat{x}_1^+$ related to $\text{pdf}(x_1|Y_1)$?

Solution:

a). Use the notation $u(x)$ to denote a pdf that is uniform on the domain $x \in [-1,1]$. Then we can write

$$p(x_1|Y_0) = \int p(x_1|x_0)p(x_0|Y_0)\,dx_0$$

$$
\begin{aligned}
&= \int u(x_1 - x_0) p(x_0)\, dx_0 \\
&= \frac{1}{2}\int_{-1}^{1} u(x_1 - x_0)\, dx_0
\end{aligned}
$$

Examination of the above integral shows that it is 0 for $x_1 < -2$, then it increases linearly to 1 as x_1 increases from -2 to 0, then it decreases linearly to 0 as x_1 increases to 2, and it is 0 for $x_1 > 2$. Therefore $p(x_1|Y_0)$ can be graphed as shown in the top of Figure 15.1. The *a posteriori* pdf of x_1 can be found from the equation

$$
p(x_1|Y_1) = \frac{p(y_1|x_1)p(x_1|Y_0)}{\int p(y_1|x_1)p(x_1|Y_0)\, dx_1}
$$

$p(y_1|x_1) = u(y_1 - x_1) = u(1 - x_1)$ (since $v_1 \sim U(-1,1)$). Therefore the product inside the integral in the denominator can be graphed as shown in the middle of Figure 15.1. Its area is equal to $1/4$. The *a posteriori* pdf of x_1 can then be written as

$$
\begin{aligned}
p(x_1|Y_1) &= 4p(y_1|x_1)p(x_1|Y_0) \\
&= 4u(1 - x_1)p(x_1|Y_0)
\end{aligned}
$$

which can be graphed as shown at the bottom of Figure 15.1.

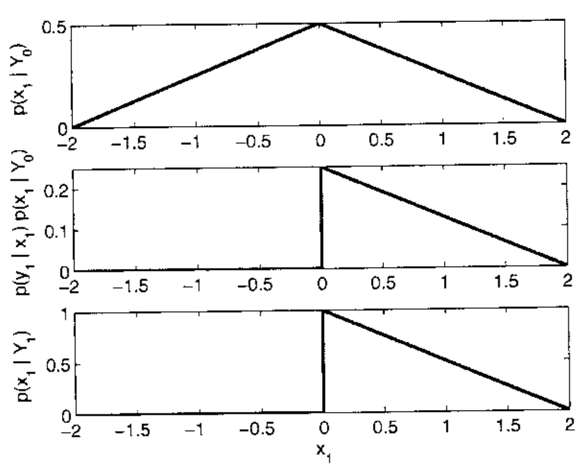

Figure 15.1 Solution to Problem 15.1a

b). Based on the pdf's of x_0, w_k, and v_k, we have $\hat{x}_0^+ = 0$, and $P_0^+ = Q = R = 1/3$. The Kalman filter equations give

$$
\begin{aligned}
\hat{x}_1^- &= 0 \\
P_1^- &= 2/3 \\
K_1 &= 2/3 \\
\hat{x}_1^+ &= 2/3
\end{aligned}
$$

Note that this value of $\hat{x}_1^+$ is the expected value of $(x_1|Y_1)$ as determined from the pdf that was found in part (a).

15.2 Suppose the pdf of an RV x is given as

$$\text{pdf}(x) = \begin{cases} 1 - x/2 & x \in [0,2] \\ 0 & \text{otherwise} \end{cases}$$

The value of x can be estimated several ways.

a) The maximum-likelihood estimate is written as $\hat{x} = \text{argmax}_x \text{pdf}(x)$. Find the maximum-likelihood estimate of x.

b) The min-max estimate of x is that value of $\hat{x}$ that minimizes the magnitude of the maximum estimation error. Find the min-max estimate of x.

c) The minimum mean square estimate of x is that value of $\hat{x}$ that minimizes $E[(x-\hat{x})^2]$. Find the minimum mean square estimate of x.

d) The expected value estimate of x is given as $\hat{x} = E(x)$. Find $E(x)$.

Solution:

a). pdf(x) attains its maximum at $x = 0$, so this is the maximum likelihood estimate of x.

b). Since x lies in the range $[0,2]$, the min-max estimate of x is equal to 1.

c). $E[(x-\hat{x})^2]$ is found as

$$\begin{aligned} E[(x-\hat{x})^2] &= \int_x (x-\hat{x})^2 \,\text{pdf}(x)\, dx \\ &= \int_0^2 (x-\hat{x})^2 (1 - x/2)\, dx \\ &= K + \hat{x}^2 - 4\hat{x}/3 \end{aligned}$$

where K is a constant. Differentiating with respect to $\hat{x}$, setting the result equal to 0, and solving for $\hat{x}$, gives the minimum mean square estimate as $\hat{x} = 2/3$.

d). The expected value of x is found as

$$\begin{aligned} E(x) &= \int_x x\,\text{pdf}(x)\, dx \\ &= 2/3 \end{aligned}$$

15.3 Suppose you have a measurement $y_k = x_k^2 + v_k$, where v_k has a triangular pdf that is given as

$$\text{pdf}(v_k) = \begin{cases} 1/2 + v_k/4 & v_k \in [-2,0] \\ 1/2 - v_k/4 & v_k \in [0,2] \\ 0 & \text{otherwise} \end{cases}$$

Suppose that five *a priori* particles $x_{k,i}^-$ are given as -2, -1, 0, 1, and 2, and that the measurement is obtained as $y_k = 1$. What are the normalized likelihoods q_i of each *a priori* particle $x_{k,i}^-$?

Solution:

The likelihoods are proportional to

$$\begin{aligned} q_i &= P\left[v_k = y^* - h(x_{k,i}^-)\right] \\ &= \text{pdf}_{v_k}\left(1 - (x_{k,i}^-)^2\right) \end{aligned}$$

The term in parentheses in the above expression is equal to -3, 0, 1, 0, -3 for the five particles. The pdf of v_k at these values is equal to 0, 1/2, 1/4, 1/2, and 0. If we normalize these values so they add up to 1 we get

$$q_i = \{\ 0,\quad 2/5,\quad 1/5,\quad 2/5,\quad 0\ \}$$

15.4 Suppose you have a measurement $y_k = v_k/x_k$, where $v_k \sim N(9, 1)$. Suppose that five *a priori* particles $x_{k,i}^-$ are given as 0.8, 0.9, 1.0, 1.1, and 1.2, and that the measurement is obtained as $y_k = 10$. What are the relative likelihoods q_i of each *a priori* particle $x_{k,i}^-$?

Solution:

The likelihoods are proportional to

$$\begin{aligned} q_i &= P\left(v_k = y^* x_{k,i}^-\right) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left[\frac{-(y^* x_{k,i}^- - 9)^2}{2}\right] \end{aligned}$$

Substituting for $y^* = 10$ and the five values of $x_{k,i}^-$ in the above equation gives

$$q_i = \{\ 0.2420,\quad 0.3989,\quad 0.2420,\quad 0.0540,\quad 0.0044\ \}$$

These likelihoods can then be scaled so they add up to one.

15.5 Suppose that five *a priori* particles are found to have probabilities 0.1, 0.1, 0.1, 0.2, and 0.5. The particles are resampled with the basic strategy depicted in Equation (15.20).

a) What is the probability that the first particle will be chosen as an *a posteriori* particle at least once?

b) What is the probability that the fifth particle will be chosen as an *a posteriori* particle at least once?

c) What is the probability that the five *a posteriori* particles will be equal to the five *a priori* particles (disregarding order)?

Solution:

a). Each *a posteriori* particle has a 0.9 probability of being equal to a particle other than the first. So the probability that the first particle will be chosen at least once is

$$\begin{aligned} P(\text{resampling } x_{k,1}^-) &= 1 - 0.9^5 \\ &= 41.95\% \end{aligned}$$

b). Each *a posteriori* particle has a 0.5 probability of being equal to a particle other than the fifth. So the probability that the fifth particle will be chosen at least once is

$$\begin{aligned} P(\text{resampling } x_{k,5}^-) &= 1 - 0.5^5 \\ &= 96.87\% \end{aligned}$$

c). The probability that $x_{k,m}^-$ will be resampled by $x_{k,m}^+$ is equal to q_m. The probability that all five *a priori* particles are resampled in order is equal to the product of the q_m probabilities. There are 5! ways of arranging the five *a priori* particles, so the total probability that all of the *a priori* particles are resampled is equal to $(5!)(0.1)(0.1)(0.1)(0.2)(0.5) = 12\%$.

15.6 Suppose you have the five particles $x_{k,i}^+ = \{\ 1,\ \ 2,\ \ 3,\ \ -2,\ \ 6\ \}$. What would you propose to use for the estimate of x_k? What would you estimate as the variance of $\hat{x}_k$?

Solution:

The most reasonable estimate for x_k is the mean of the particles, which is

$$\begin{aligned} \hat{x}_k &= (1+2+3-2+6)/5 \\ &= 2 \end{aligned}$$

The most reasonable estimate for the variance of $\hat{x}_k$ is

$$\begin{aligned} \text{var}(\hat{x}_k) &= \frac{1}{4}\sum_{i=1}^{5}\left(\hat{x}_k - x_{k,i}^+\right)^2 \\ &= 8.5 \end{aligned}$$

This gives an unbiased estimate for the variance of $\hat{x}_k$, as discussed in Problem 3.6.

15.7 Suppose that you have five particles -1, -1, 0, 1, and 1. You want to use the roughening procedure of Section 15.3.1.1 to add a uniform random variable with a variance of $KMN^{-1/n}$ to each particle. What range of K will give a probability of at least 1/8 that at least one of the roughened particles is less than -2?

Solution:

We have $M = 2$, $N = 5$, and $n = 1$, so our uniform random variable has a variance of $2K/5$, which means that it is distributed on the range $[-c, +c]$, where $c = \sqrt{6K/5}$. The probabilities that the roughened particles are less than -2 are equal to

$$\begin{aligned} P(x_1 < -2) &= \max\left(0, \frac{c-1}{2c}\right) \\ P(x_2 < -2) &= \max\left(0, \frac{c-1}{2c}\right) \\ P(x_3 < -2) &= \max\left(0, \frac{c-2}{2c}\right) \end{aligned}$$

$$
\begin{aligned}
P(x_4 < -2) &= \max\left(0, \frac{c-3}{2c}\right) \\
P(x_5 < -2) &= \max\left(0, \frac{c-3}{2c}\right)
\end{aligned}
$$

If $c > 3$ the probability that at least one roughened particle is less than -2 is equal to

$$
P(\text{at least one } < -2) = 1 - \left(1 - \frac{c-1}{2c}\right)^2 \left(1 - \frac{c-2}{2c}\right)\left(1 - \frac{c-3}{2c}\right)^2 \qquad (c > 3)
$$

The above expression is greater than 1/8 for all $c > 3$. If $c \in [2, 3]$ the probability that at least one roughened particle is less than -2 is equal to

$$
P(\text{at least one } < -2) = 1 - \left(1 - \frac{c-1}{2c}\right)^2 \left(1 - \frac{c-2}{2c}\right) \qquad (c \in [2.3])
$$

The above expression is greater than 1/8 for all $c \in [2, 3]$. If $c \in [1, 2]$ the probability that at least one roughened particle is less than -2 is equal to

$$
P(\text{at least one } < -2) = 1 - \left(1 - \frac{c-1}{2c}\right)^2 \qquad (c \in [1, 2])
$$

The above expression is greater than 1/8 for $c > (2 + \sqrt{14})/5$, which means that $K > (9 + 2\sqrt{14})/15$. Combining the above results gives the final answer that $K > (9 + 2\sqrt{14})/15 \approx 1.0989$ in order to obtain a probability greater than 1/8 that at least one roughened particle is less than -2.

15.8 Suppose you have the system equation $x_{k+1} = x_k$ and the measurement equation $y_k = x_k^2 + v_k$, where v_k has a triangular pdf that is given as

$$
\text{pdf}(v_k) = \begin{cases} 1/2 + v_k/4 & v_k \in [-2, 0] \\ 1/2 - v_k/4 & v_k \in [0, 2] \\ 0 & \text{otherwise} \end{cases}
$$

Suppose that five *a posteriori* particles $x_{k-1,i}^+$ are given as -2, -1, 0, 1, and 2, and that the measurement is obtained as $y_k = 1$. You want to use prior editing to ensure that the -2 particle has at least a 10% chance (after one roughening step) of being selected as an *a posteriori* particle at the next time step. What value of K should you use in your roughening step?

Solution:

The roughening step is performed by adding Δx to each *a posteriori* particle, where $\Delta x \sim (0, KMN^{-1/n}) = (0, 4K/5)$. Since Δx is uniform, it is distributed on the domain $[-c, +c]$ where $c = \sqrt{12K/5}$. The probability of resampling a particle $x_{k,i}^-$ is nonzero if

$$
\begin{aligned}
\left| y_k - \left(x_{k,i}^-\right)^2 \right| &< 2 \\
|x_{k,i}^-| &< \sqrt{3}
\end{aligned}
$$

So the probability of resampling a roughened particle $(-2 + \Delta x)$ is nonzero if $\Delta x > 2 - \sqrt{3}$. Since $\Delta x \sim U[-c, +c]$, this probability is

$$P(\text{resample}) = \frac{c - (2 - \sqrt{3})}{2c}$$

The above probability is greater than P if

$$c > \frac{2 - \sqrt{3}}{1 - 2P}$$

Substituting $P = 0.1$ gives $c > 0.33$ which means that $K > 0.047$.

15.9 Suppose you have two particles -1 and $+1$, both with *a priori* probabilities $1/2$. Use the kernel bandwidth $h = 1$ with the regularized particle filter to find the pdf approximations $\hat{p}(x_k = -2|y_k)$, $\hat{p}(x_k = -1|y_k)$, $\hat{p}(x_k = 0|y_k)$, $\hat{p}(x_k = 1|y_k)$, and $\hat{p}(x_k = 2|y_k)$. For what values of x_k is the pdf approximation $\hat{p}(x_k|y_k)$ equal to zero?

Solution:

Using Equations (15.36) and following, we find

$$\begin{aligned}
\mu &= 0 \\
S &= 2 \\
A &= \sqrt{2} \\
v_1 &= 2 \\
K(x) &= \begin{cases} \frac{3}{4}\left(1 - x^2\right) & |x| < 1 \\ 0 & \text{otherwise} \end{cases} \\
K_h(x) &= \begin{cases} \frac{3}{4\sqrt{2}}\left(1 - x^2/2\right) & |x| < \sqrt{2} \\ 0 & \text{otherwise} \end{cases} \\
\hat{p}(x_k|y_k) &= \frac{1}{2}\left(K_h(x_k + 1) + K_h(x_k - 1)\right)
\end{aligned}$$

Substituting $x_k = -2, -1, 0, 1$, and 2 into the above equation gives

$$\begin{aligned}
\hat{p}(x_k = -1|y_k) = \hat{p}(x_k = 0|y_k) = \hat{p}(x_k = 1|y_k) &= \frac{3}{8\sqrt{2}} \\
\hat{p}(x_k = -2|y_k) = \hat{p}(x_k = 2|y_k) &= \frac{3}{16\sqrt{2}}
\end{aligned}$$

$\hat{p}(x_k|y_k) = 0$ for $|x_k| > \sqrt{2} + 1$.

15.10 Suppose you have N resampling probabilities q_i with sample mean μ and sample variance S. What is the sample mean and variance of the auxiliary probabilities given by Equation (15.44)?

Solution:

The sample mean of the auxiliary probabilities is

$$\tilde{\mu} = \frac{1}{N}\sum \frac{(\alpha - 1)q_i + \mu}{\alpha}$$

$$
\begin{aligned}
&= \frac{1}{\alpha N}\left[(\alpha-1)\sum q_i + \sum \mu\right] \\
&= \frac{1}{\alpha}\left[(\alpha-1)\mu + \mu\right] \\
&= \frac{\alpha\mu}{\alpha} \\
&= \mu
\end{aligned}
$$

The sample variance of the auxiliary probabilities is

$$
\begin{aligned}
\tilde{S} &= \frac{1}{N-1}\sum\left[\frac{(\alpha-1)q_i+\mu}{\alpha} - \mu\right]^2 \\
&= \frac{1}{\alpha^2(N-1)}\sum\left[(\alpha-1)q_i + \mu - \alpha\mu\right]^2 \\
&= \frac{1}{\alpha^2(N-1)}\sum(\alpha-1)^2(q_i-\mu)^2 \\
&= \left(\frac{\alpha-1}{\alpha}\right)^2 \frac{1}{N-1}\sum(q_i-\mu)^2 \\
&= \left(\frac{\alpha-1}{\alpha}\right)^2 S
\end{aligned}
$$

Computer exercises

15.11 Plot the volume of the n-dimensional unit hypersphere as a function of n for $n \in [1, 20]$.

Solution:

Using the equations $v_1 = 2$, $v_2 = \pi$, $v_3 = 4\pi/3$, and $v_n = 2\pi v_{n-2}/n$, we come up with Figure 15.2.

15.12 Consider two particles $x_1 = 1$ and $x_2 = 2$, with equal probabilities. Generate the approximate pdf using the Epanechnikov kernel with bandwidth $h = h^*$. Generate two separate plots (on the same figure) of the two individual terms in the summation of Equation (15.29), and also generate a plot (on the same figure) of their sum. Repeat for three particles $x_1 = 1$, $x_2 = 2$, and $x_3 = 3$ with equal probabilities.

Solution:

The plots are shown in Figure 15.3.

15.13 Repeat Problem 15.12 with $h = h^*/2$ and with $h = 2h^*$. This shows that the bandwidth selection can have a strong effect on the pdf approximation.

Solution:

The plots are shown in Figures 15.4 and 15.5.

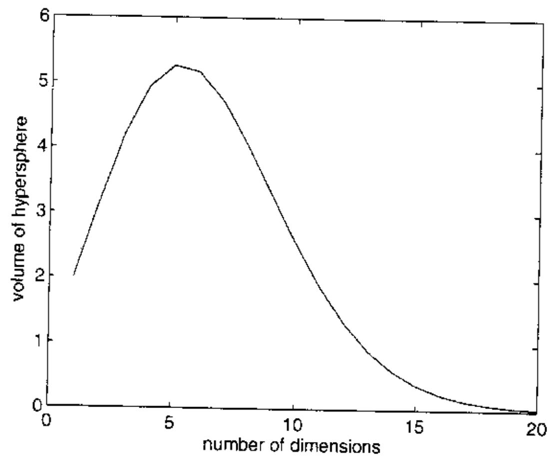

Figure 15.2 Solution to Problem 15.11

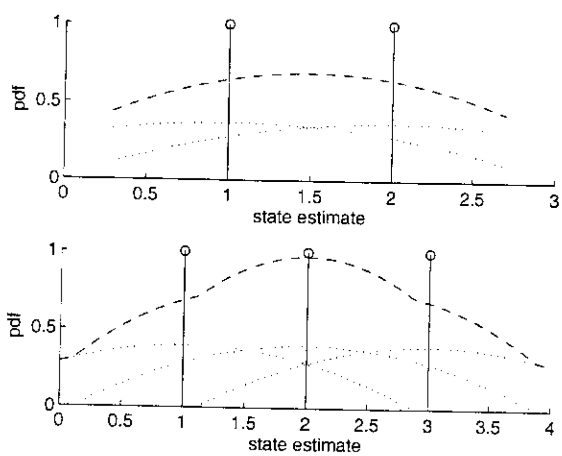

Figure 15.3 Solution to Problem 15.12. Bandwidth $h = h^*$

15.14 Kernels other than the Epanechnikov kernel can also be used for pdf approximation [Sim98, Dev01]. Some of the more popular kernels can be described in one dimension as follows.

$$
\begin{aligned}
\text{Epanechnikov: } K(x) &= \begin{cases} \frac{3}{4}(1-x^2) & |x| < 1 \\ 0 & \text{otherwise} \end{cases} \\
\text{Gaussian: } K(x) &= (2\pi)^{-1/2}\exp(-x^2/2) \\
\text{Uniform: } K(x) &= \begin{cases} \frac{1}{2} & |x| < 1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

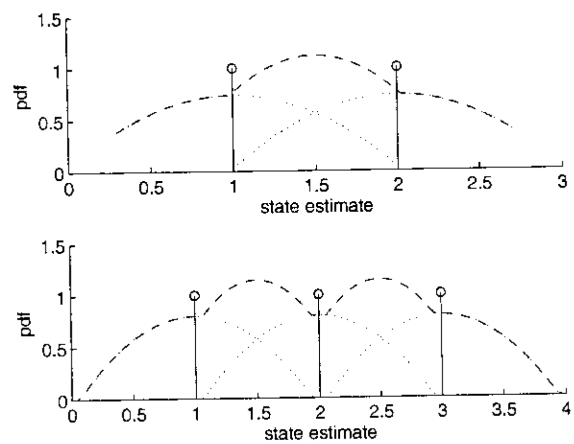

Figure 15.4 Solution to Problem 15.13. Bandwidth $h = h^*/2$

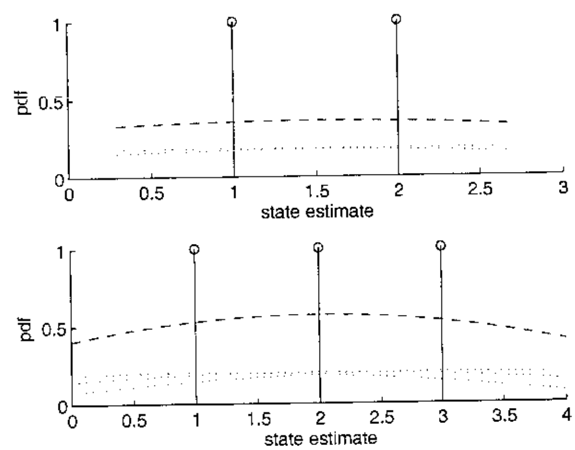

Figure 15.5 Solution to Problem 15.13. Bandwidth $h = 2h^*$

$$\text{Triangular: } K(x) = \begin{cases} 1 - x^2 & |x| < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Biweight: } K(x) = \begin{cases} \frac{15}{16}(1 - x^2)^2 & |x| < 0 \\ 0 & \text{otherwise} \end{cases}$$

Bandwidth selection is another matter, but for this problem you can simply use the optimal Epanechnikov bandwidth for all of the kernels.

a) Repeat Problem 15.12 using Gaussian kernels.

b) Repeat Problem 15.12 using uniform kernels.

c) Repeat Problem 15.12 using triangular kernels.

d) Repeat Problem 15.12 using biweight kernels.

Solution:

a). The solution using Gaussian kernels is shown in Figure 15.6.

b). The solution using uniform kernels is shown in Figure 15.7.

c). The solution using triangular kernels is shown in Figure 15.8.

d). The solution using biweight kernels is shown in Figure 15.9.

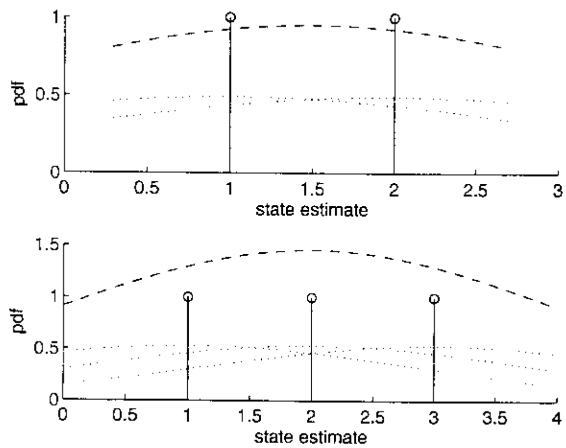

Figure 15.6 Solution to Problem 15.14 using Gaussian kernels

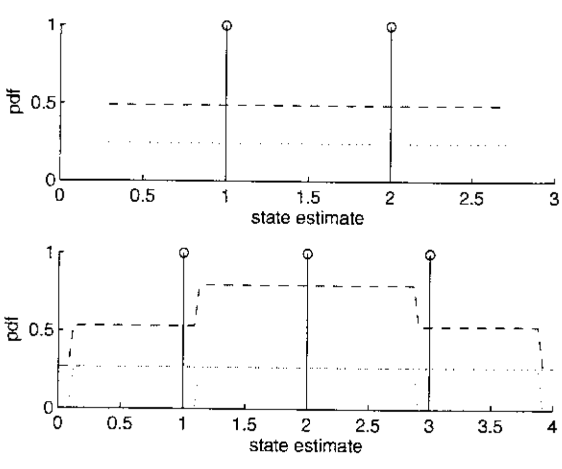

Figure 15.7 Solution to Problem 15.14 using uniform kernels

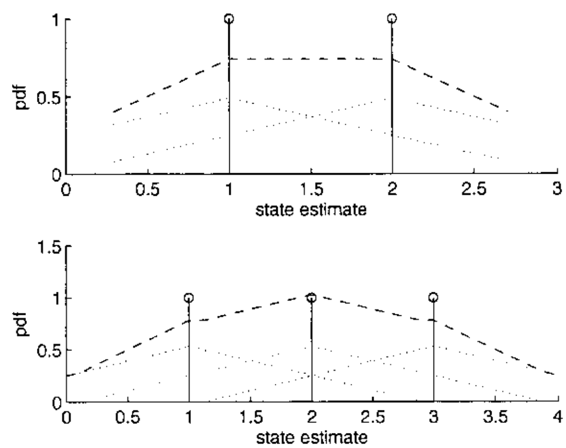

Figure 15.8 Solution to Problem 15.14 using triangular kernels

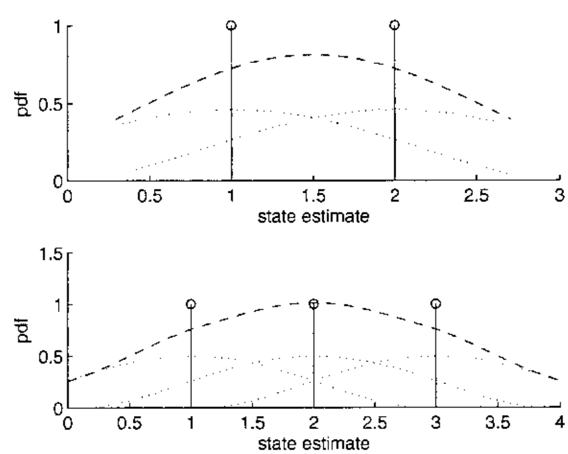

Figure 15.9 Solution to Problem 15.14 using biweight kernels

15.15 In this problem, we will explore the performance of the EKF and the particle filter for the system described in Example 15.1.

a) Run 100 simulations of the EKF and the particle filter with $N = 10$, $N = 100$, and $N = 1000$. What is the average RMS state-estimation error for each case?

b) Run 100 simulations of the EKF and the particle filter with $N = 100$ using $Q = 0.1$, $Q = 1$, and $Q = 10$. What is the average RMS state-estimation error for each case?

Solution:

a). The average RMS estimation error is 16.7 for the EKF. With the particle filter, the error is 6.1 for $N = 10$, 2.9 for $N = 100$, and 2.9 for $N = 1000$. (Your numbers may vary slightly.) This shows that increasing the number of particles improves filtering performance, but only to a certain point.

b). The average RMS estimation error is 13.8 for the EKF and 1.7 for the particle filter when $Q = 0.1$. The error is 16.7 for the EKF and 3.2 for the particle filter when $Q = 1$. The error is 25.8 for the EKF and 4.9 for the particle filter when $Q = 10$. (Your numbers may vary slightly.) As Q increases, the EKF error and particle filter error both increase at about the same rate, but the particle filter error is smaller than the EKF error.

```
function MotorProb(dt)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 1.17

% Two-state DC motor simulation - rectangular integration.
% INPUT: dt = integration step size
% Use dt = 0.05 for a good simulation, dt = 0.2 for a marginal simulation,
% and dt = 0.5 for an unstable simulation.

J = 10; % moment of inertia
F = 100; % coefficient of viscous friction
A = [ 0 1 ; 0 -F/J ];
B = [ 0 ; 1/J ];
x = [ 0 ; 0 ]; % initial state

if ~exist('dt', 'var')
    dt = 0.05;
end

tf = 5; % simulation length

xArr = [x];

for t = dt : dt : tf+dt/10
    u = 10;
    xdot = A * x + B * u;
    x = x + xdot * dt;
    xArr = [xArr x];
end

t = 0 : dt : tf;

close all;
figure;
plot(t, xArr(1,:), 'b-', t, xArr(2,:), 'r:');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('Seconds');
legend('position', 'velocity');
title(['dt = ', num2str(dt)]);
```

```
function RLC(IntFlag, dt)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 1.18

% Simulate a series RLC circuit.
% INPUT: dt = simulation step size
%        IntFlag = 0 for rectangular, 1 for trapezoidal, 2 for Runge Kutta

% The output is the voltage across the capacitor.

tf = 5; % simulation length

% Set the R, L, and C values.
R = 3;
L = 1;
C = 0.5;

% Set the system matrices; xdot = Ax + Bu, and y = Cx + Du.
A = [0 1/C; -1/L -R/L];
B = [0; 1/L];
C = [1 0];
D = [0];

x = [0 ; 0]; % initial condition

yarray = x(1);
for t = 0 : dt : tf - dt + dt/10
    u = exp(-2*t);
    if (IntFlag == 0)
        % Rectangular integration
        xdot = A * x + B * u;
        x = x + xdot * dt;
    elseif (IntFlag == 1)
        % Trapezoidal integration
        dx1 = (A * x + B * u) * dt;
        dx2 = (A * (x + dx1) + B * u) * dt;
        x = x + (dx1 + dx2) / 2;
    else
        u1 = exp(-2*(t+dt/2));
        dx1 = (A * x + B * u) * dt;
        dx2 = (A * (x + dx1 / 2) + B * u1) * dt;
        dx3 = (A * (x + dx2 / 2) + B * u1) * dt;
        dx4 = (A * (x + dx3) + B * u1) * dt;
        x = x + (dx1 + 2 * dx2 + 2 * dx3 + dx4) / 6;
    end
    y = C * x + D * u;
    yarray = [yarray y];
end
```

```
% Close all figures.
close all;
figure;
t = 0 : dt : tf;
plot(t, yarray, 'b-');
yAnalytical = 2 * (exp(-t) - exp(-2*t) - t .* exp(-2*t));
hold;
plot(t, yAnalytical, 'r:');
legend('Numerical solution', 'Analytical solution');

Err = sqrt((norm(yarray - yAnalytical))^2 / length(yarray))
```

```
function Rocket(ControlMag)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 1.19

% Hovering rocket simulation to show the effects of linearization.

% Close all figures.
close all;
figure;

% Define the step size and the simulation length.
dt = 0.01;
tf = 5;

G = 6.673e-11; % gravitational constant (m^3/kg/s^2)
M = 5.98e24; % earth mass (kg)
m0 = 1000; % initial rocket mass (kg)
R = 6.37e6; % earth radius (m)
K = 1000; % thrust proportionality constant
g = 50; % drag constant

ControlArr = [10 100 300];
for i = 1 : 3
    ControlMag = ControlArr(i);
    % Nonlinear simulation.
    x1 = 0;
    x2 = 0;
    x3 = m0;
    x1array = [];
    x2array = [];
    x3array = [];

    for t = 0 : dt : tf + dt/10
        u = G * M * x3 / K / R / R + ControlMag * abs(cos(t));
        x1dot = x2;
        x2dot = (K * u - g * x2) / x3 - G * M / (R + x1)^2;
        x3dot = -u;
        x1 = x1 + dt * x1dot;
        x2 = x2 + dt * x2dot;
        x3 = x3 + dt * x3dot;
        x1array = [x1array x1];
        x2array = [x2array x2];
        x3array = [x3array x3];
    end

    % Linearized simulation.
    x1bar = 0;
    x2bar = 0;
    x3bar = 0;
```

```
    x1arrayLin = [];
    x2arrayLin = [];
    x3arrayLin = [];

    for t = 0 : dt : tf + dt/10
        m = m0 * exp(-G * M * t / K / R / R);
        ubar = ControlMag * abs(cos(2*pi*t));
        x1bardot = x2bar;
        x2bardot = (2 * G * M / R / R / R) * x1bar - (g / m) * x2bar - (G
* M / R / R / m) * x3bar + (K / m) * ubar;
        x3bardot = -ubar;
        x1bar = x1bar + dt * x1bardot;
        x2bar = x2bar + dt * x2bardot;
        x3bar = x3bar + dt * x3bardot;
        x1arrayLin = [x1arrayLin 0 + x1bar];
        x2arrayLin = [x2arrayLin 0 + x2bar];
        x3arrayLin = [x3arrayLin m + x3bar];
    end

    % Plot the results.
    t = [0 : dt : tf];
    subplot(3,1,i);
    plot(t, x1array, '-', t, x1arrayLin, '--');
    set(gca,'FontSize',12); set(gcf,'Color','White');
    if (i == 3), xlabel('Time (seconds)'); end;
    if (i == 2), ylabel('Altitude (meters)'); end;
    if (i == 1), legend('Nonlinear', 'Linearized'); end;
    title(['\Delta u = ', num2str(ControlMag)]);
end
```

```
function Uniform(N)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 2.15

% Plot a histogram of uniformly distributed random numbers.
% INPUT: N = number of random numbers to generate

if ~exist('N', 'var')
    N = 50;
end

close all;
figure;
set(gcf,'Color','White');
rand('state', sum(100*clock)); % initialize the random number generator
for k = 1 : 3
    for i = 1 : N
        x(i) = rand;
    end
    xmean = mean(x);
    xstd = std(x);
    disp(['mean = ', num2str(xmean), ', std dev = ', num2str(xstd)]);
    subplot(3,1,k);
    hist(x,10); set(gca,'FontSize',12);
    legend(['N = ', num2str(N)]);
    N = 10 * N;
end
```

```
function Central

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 2.16

% Illustration of the Central Limit Theorem

close all;
figure; set(gcf,'Color','White');
rand('state', sum(100*clock)); % initialize the random number generator
for N = 2 : 2 : 4
    for i = 1 : 10000
        x(i) = 0;
        for j = 1 : N
            x(i) = x(i) + (rand-0.5) / N;
        end
    end
    subplot(2,1,N/2); set(gca,'FontSize',12);
    hist(x,50);
    legend(['N = ', num2str(N)]);
end
```

```
function LeastSteel

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 3.13

% Least squares curve fit for steel production data.

y = [66.6, 84.9, 88.6, 78.0, 96.8, 105.2, 93.2, 111.6, 88.3, 117.0,
115.2];
N = length(y);

close all; % close all figures
figure; % open a new figure
set(gcf,'Color','White');

% Linear curve fit.
xhat = [0; 0];
P = 100000 * eye(2);
R = 1;
for t = 1 : N
   H = [1 t];
   K = P * H' * inv(H * P * H' + R);
   xhat = xhat + K * (y(t) - H * xhat);
   P = (eye(2) - K * H) * P * (eye(2) - K * H)' + K * R * K';
end
% Compute the RMS error.
err = 0;
for t = 1 : N
   H = [1 t];
   err = err + (y(t) - H * xhat)^2;
end
err = sqrt(err / N);
disp(['Linear RMS Error = ',num2str(err)]);
H = [1 N+1];
disp(['Value predicted at next time = ', num2str(H*xhat)]);
% Plot the results.
t = 0 : N-1;
subplot(2,2,1);
plot(t, y, 'o', t, xhat(1)+xhat(2)*t);
legend('linear');
set(gca,'FontSize',12);

% Quadratic curve fit.
xhat = [0; 0; 0];
P = 100000 * eye(3);
R = 1;
for t = 1 : N
   H = [1 t t*t];
   K = P * H' * inv(H * P * H' + R);
   xhat = xhat + K * (y(t) - H * xhat);
```

```
   P = (eye(3) - K * H) * P * (eye(3) - K * H)' + K * R * K';
end
% Compute the mean square error.
err = 0;
for t = 1 : N
   H = [1 t t*t];
   err = err + (y(t) - H * xhat)^2;
end
err = sqrt(err / N);
disp(['Quadratic RMS Error = ',num2str(err)]);
H = [1 N+1 (N+1)^2];
disp(['Value predicted at next time = ', num2str(H*xhat)]);
% Plot the results.
t = 0 : N-1;
subplot(2,2,2);
plot(t, y, 'o', t, xhat(1)+xhat(2)*t+xhat(3)*t.*t);
legend('quadratic');
set(gca,'FontSize',12);

% Cubic curve fit.
xhat = [0; 0; 0; 0];
P = 100000 * eye(4);
R = 1;
for t = 1 : N
   H = [1 t t*t t*t*t];
   K = P * H' * inv(H * P * H' + R);
   xhat = xhat + K * (y(t) - H * xhat);
   P = (eye(4) - K * H) * P * (eye(4) - K * H)' + K * R * K';
end
% Compute the mean square error.
err = 0;
for t = 1 : N
   H = [1 t t*t t*t*t];
   err = err + (y(t) - H * xhat)^2;
end
err = sqrt(err / N);
disp(['Cubic RMS Error = ',num2str(err)]);
H = [1 N+1 (N+1)^2 (N+1)^3];
disp(['Value predicted at next time = ', num2str(H*xhat)]);
% Plot the results.
t = 0 : N-1;
subplot(2,2,3);
plot(t, y, 'o', t, xhat(1)+xhat(2)*t+xhat(3)*t.*t+xhat(4)*t.*t.*t);
legend('cubic');
set(gca,'FontSize',12);

% Quartic curve fit.
xhat = [0; 0; 0; 0; 0];
P = 100000 * eye(5);
R = 1;
```

```
for t = 1 : N
   H = [1 t t*t t*t*t t*t*t*t];
   K = P * H' * inv(H * P * H' + R);
   xhat = xhat + K * (y(t) - H * xhat);
   P = (eye(5) - K * H) * P * (eye(5) - K * H)' + K * R * K';
end
% Compute the mean square error.
err = 0;
for t = 1 : N
   H = [1 t t*t t*t*t t*t*t*t];
   err = err + (y(t) - H * xhat)^2;
end
err = sqrt(err / N);
disp(['Quartic RMS Error = ',num2str(err)]);
H = [1 N+1 (N+1)^2 (N+1)^3 (N+1)^4];
disp(['Value predicted at next time = ', num2str(H*xhat)]);
% Plot the results.
t = 0 : N-1;
subplot(2,2,4);
plot(t, y, 'o', t, xhat(1)+xhat(2)*t+xhat(3)*t.*t+xhat(4)*t.*t.*t+xhat(5)
*t.*t.*t.*t);
legend('quartic');
set(gca,'FontSize',12);
```

```
function [ErrW, ErrWC, ErrWN, ErrK] = Wiener1(dt)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 3.14

% Wiener filter problem.
% Inputs:
%    dt = simulation step size (typically 0.1)
% Ouputs:
%    ErrW = E(e^2) for the parameter Wiener filter
%    ErrWC = E(e^2) for the causal Wiener filter
%    ErrWN = E(e^2) for the noncausal Wiener filter
%    ErrK = E(e^2) for the time varying Kalman filter

if ~exist('dt', 'var')
    dt = 0.1;
end
tf = 100; % simulation length
Qc = 2; % continuous time process noise
Rc = 1; % continuous time measurement noise
T = 1 / (sqrt(2) - 1); % parametric Wiener filter
x = 0; % initial state
xhatW = 0; % initial parametric Wiener estimate
xhatWC = 0; % initial causal Wiener estimate
xhatK = 0; % initial Kalman estimate
K = sqrt(3) - 1; % steady state Kalman gain
P = 0; % initial Kalman filter error covariance

xArr = [x];
yArr = [x];
xhatWArr = [xhatW];
xhatWCArr = [xhatWC];
xhatWNArr = [];
xhatKArr = [xhatK];

for t = dt : dt : tf+dt/10
    % System simulation
    x = exp(-dt) * x + sqrt(Qc*dt) * randn;
    y = x + sqrt(Rc/dt) * randn;
    % Parametric Wiener filter simulation
    xhatWdot = -xhatW / T + y / T;
    xhatW = xhatW + xhatWdot * dt;
    % Causal Wiener filter simulation
    xhatWCdot = -sqrt(3) * xhatWC + (sqrt(3) - 1) * y;
    xhatWC = xhatWC + xhatWCdot * dt;
    % Noncausal Wiener filter simulation
    xhatWCdot = -sqrt(3) * xhatWC + (sqrt(3) - 1) * y;
    xhatWC = xhatWC + xhatWCdot * dt;
    % Kalman filter simulation
    K = P * inv(Rc);
```

```
    xhatKdot = -xhatK + K * (y - xhatK);
    xhatK = xhatK + xhatKdot * dt;
    Pdot = -P^2 - 2 * P + 2;
    P = P + Pdot * dt;
    % Save data in arrays
    xArr = [xArr x];
    yArr = [yArr y];
    xhatWArr = [xhatWArr xhatW];
    xhatWCArr = [xhatWCArr xhatWC];
    xhatKArr = [xhatKArr xhatK];
end

% Noncausal Wiener filter
N = length(yArr);
xhatcArr = zeros(1,N);
xhataArr = zeros(1,N);
i = 0;
xhatc = 0; % causal part of Wiener filter output
xhata = 0; % anticausal part of Wiener filter output
for t = dt : dt : tf+dt/10
    xhatadot = sqrt(3) * xhata - yArr(N-i) / sqrt(3);
    xhata = xhata - xhatadot * dt;
    xhataArr(N-i) = xhata;
    i = i + 1;
    xhatcdot = -sqrt(3) * xhatc + yArr(i) / sqrt(3);
    xhatc = xhatc + xhatcdot * dt;
    xhatcArr = [xhatcArr xhatc];
end
for i = 1 : N
    xhatWNArr = [xhatWNArr xhatcArr(i)+xhataArr(i)];
end

close all;
figure; hold on;
plot(xArr,'r'); plot(xhatWArr,'b'); plot(xhatKArr,'k'); plot
(xhatWNArr,'g');
legend('true', 'Causal Wiener', 'Kalman', 'Noncausal Wiener');
ErrW = sum((xArr-xhatWArr).^2) / length(xArr);
ErrWC = sum((xArr-xhatWCArr).^2) / length(xArr);
ErrWN = sum((xArr-xhatWNArr).^2) / length(xArr);
ErrK = sum((xArr-xhatKArr).^2) / length(xArr);
disp(['Parametric Wiener Err^2 = ', num2str(ErrW)]);
disp(['Causal Wiener Err^2 = ', num2str(ErrWC)]);
disp(['Noncausal Wiener Err^2 = ', num2str(ErrWN)]);
disp(['Kalman Err^2 = ', num2str(ErrK)]);
```

```
function Prop1

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 4.11

% Mean and variance propagation for a linear system.

tf = 5;
dt = 0.1;

f = -0.5;
qc = 1;
m = 1;

P1 = 0;
P2 = 2;

mArray = m;
P1Array = P1;
P2Array = P2;

for t = dt : dt : tf+dt/10
   m = exp(f * dt) * m;
   P1 = (1 + 2 * f * dt) * P1 + qc * dt;
   P2 = (1 + 2 * f * dt) * P2 + qc * dt;
   mArray = [mArray m];
   P1Array = [P1Array P1];
   P2Array = [P2Array P2];
end

close all;
t = 0 : dt : tf+dt/10;
figure;
set(gcf,'Color','White');
set(gca,'FontSize',12);

subplot(2,1,1);
plot(t, mArray);
legend('mean');
set(gca,'FontSize',12);

subplot(2,1,2);
plot(t, P2Array, 'r-', t, P1Array, 'b:');
legend('variance (P_0 = 2)', 'variance (P_0 = 0)');
set(gca,'FontSize',12);

xlabel('time');
```

```
function Prop2

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 4.12

% Mean and variance propagation for an RLC circuit.

tf = 5;
dt = 0.1;

A = [-2 1 ; -1 0];
B = [1 ; 1];
Qc = B * 1 * B';

m = [1 ; 2];
P = [1 0 ; 0 2];

mArray = m;
PArray(:,:,1) = P;

k = 2;
for t = dt : dt : tf+dt/10
   u = 0;
   mdot = A * m + B * u;
   m = m + mdot * dt;
   Pdot = A * P + P * A' + Qc;
   P = P + Pdot * dt;
   mArray = [mArray m];
   PArray(:,:,k) = P;
   k = k + 1;
end

close all;
t = 0 : dt : tf+dt/10;
figure;
set(gcf,'Color','White');

subplot(2,1,1);
plot(t, mArray(1,:), 'r-', t, mArray(2,:), 'b:');
legend('m(1)', 'm(2)');
set(gca,'FontSize',12);

subplot(2,1,2);
plot(t, squeeze(PArray(1,1,:)), 'r-', t, squeeze(PArray(1,2,:)), 'm--', t,
squeeze(PArray(2,2,:)), 'b:');
legend('P(1,1)', 'P(1,2)', 'P(2,2)');
set(gca,'FontSize',12);

xlabel('time');
```

```
function Prop3

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 4.13

% Show the difference between the continuous steady state covariance
% and the discretized steady state covariance for an RLC circuit.

A = [0 2 ; -1 -3];
Qc = [0 0 ; 0 1];

k = 1;
for T = 0.01 : 0.01 : 1
    F = expm(A*T);
    Q = T * Qc;
    err(k) = norm(lyap(A, Qc) - dlyap(F, Q), 'fro');
    k = k + 1;
end

T = 0.01 : 0.01 : 1;
close all;
plot(T, err);
set(gcf,'Color','White');
set(gca,'FontSize',12);
xlabel('discretization step size');
ylabel('steady state covariance error');
```

```
function Population

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 5.11

% Kalman filter for wombat population estimation.

tf = 10; % simulation length

F = [1/2 2 ; 0 1];
Q = [0 0 ; 0 10];
H = [1 0];
R = 10;
Pplus = [500 0 ; 0 200]; % initial estimation error covariance

x = [650 ; 250]; % initial state
xhat = [600 ; 200]; % initial estimate

% Initialize arrays
xArr = x;
xhatArr = xhat;
PArr = [Pplus(1,1) ; Pplus(2,2)];
KArr = [];

for k = 1 : tf
    % System simulation
    x = F * x + sqrt(Q) * randn(2,1);
    y = H * x + sqrt(R) * randn;
    % Kalman filter simulation
    Pminus = F * Pplus * F' + Q;
    K = Pminus * H' * inv(H * Pminus * H' + R);
    xhat = F * xhat;
    xhat = xhat + K * (y - H * xhat);
    Pplus = (eye(2) - K * H) * Pminus;
    % Save data for plotting
    xArr = [xArr x];
    xhatArr = [xhatArr xhat];
    PArr = [PArr [Pplus(1,1) ; Pplus(2,2)]];
    KArr = [KArr K];
end

% Plot the results
close all;
k = 0 : tf;

figure; set(gca,'FontSize',12); set(gcf,'Color','White');
plot(k, xArr(1,:), 'r:', k, xhatArr(1,:), 'b-'); xlabel('time');
legend('true population', 'estimated population');

figure; set(gca,'FontSize',12); set(gcf,'Color','White');
```

```
plot(k, xArr(2,:), 'r:', k, xhatArr(2,:), 'b-'); xlabel('time');
legend('true food supply', 'estimated food supply');

figure; set(gca,'FontSize',12); set(gcf,'Color','White');
plot(k, sqrt(PArr(1,:)), 'r:', k, sqrt(PArr(2,:)), 'b-'); xlabel('time');
legend('population est std dev', 'food supply est std dev');

k1 = 1 : tf;
figure; set(gca,'FontSize',12); set(gcf,'Color','White');
plot(k1, KArr(1,:), 'r:', k1, KArr(2,:), 'b-'); xlabel('time');
legend('K(1)', 'K(2)');

figure; set(gcf,'Color','White');
subplot(2,2,1);
plot(k, xArr(1,:), 'r:', k, xhatArr(1,:), 'b-'); set(gca,'FontSize',12);
ylabel('population');
legend('true', 'estimated');
subplot(2,2,2);
plot(k, xArr(2,:), 'r:', k, xhatArr(2,:), 'b-'); set(gca,'FontSize',12);
ylabel('food supply');
legend('true', 'estimated');
subplot(2,2,3);
plot(k, sqrt(PArr(1,:)), 'r:', k, sqrt(PArr(2,:)), 'b-'); set
(gca,'FontSize',12);
ylabel('std dev of est');
legend('population', 'food supply');
xlabel('time');
subplot(2,2,4);
plot(k1, KArr(1,:), 'r:', k1, KArr(2,:), 'b-'); xlabel('time'); set
(gca,'FontSize',12);
legend('K(1)', 'K(2)');
xlabel('time');

EstStd = std(xArr' - xhatArr');
disp(['Experimental std dev of estimation error = ', num2str(EstStd(1)),
', ', num2str(EstStd(2))]);
EstStd = sqrt(Pplus);
disp(['Theoretical std dev of estimation error = ', num2str(EstStd(1,1)),
', ', num2str(EstStd(2,2))]);
```

```
function RLC

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 5.12

% Kalman filter for RLC circuit state estimation.

T = 0.1;
F = exp(-T) * [2 2;-1 -1] + exp(-2*T) * [-1 -2;1 2];
H = [1 0];
q = 1;
Q = [0 0; 0 q];
R = 1;

x = [0 ; 0];
xhat = x;
Pplus = [0 0; 0 0];

N = 21;

Varminus = [];
Varplus = [Pplus(2,2)];
KArray = [];
xArray = [];
xhatArray = [];
yArray = [];

for k = 1 : N
    % Simulate the system and measurement
    x = F * x + [0; 1] * sqrt(q) * randn;
    y = H * x + sqrt(R) * randn;
    % Estimate the state
    Pminus = F * Pplus * F' + Q;
    K = Pminus * H' * inv(H * Pminus * H' + R);
    xhat = F * xhat;
    xhat = xhat + K * (y - H * xhat);
    Pplus = Pminus - K * H * Pminus;
    % Save data for plotting
    xArray = [xArray x];
    xhatArray = [xhatArray xhat];
    yArray = [yArray y];
    Varminus = [Varminus Pminus(2,2)];
    Varplus = [Varplus Pplus(2,2)];
    KArray = [KArray K];
end

disp(['inductor current variance = ', num2str(Pminus(2,2))]);

% Plot the results
close all;
```

```
k = 1 : N;
plot(k, yArray-xArray(1,:), 'r:');
hold;
plot(k, xhatArray(1,:)-xArray(1,:), 'b-');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('time step'); ylabel('position');
legend('measurement error', 'estimation error');

figure; hold;
for k = 1 : N-1
    plot([k-1 k], [Varplus(k) Varminus(k+1)]);
    plot([k k], [Varminus(k+1) Varplus(k+1)]);
end
set(gca,'FontSize',12); set(gcf,'Color','White'); set(gca,'Box','on');
xlabel('time step');
ylabel('estimation error variance');

Psi = [(F + Q * inv(F') * H' * inv(R) * H) (Q * inv(F')) ; (inv(F') * H' *
inv(R) * H) (inv(F'))];
Psip = Psi^2;
AB = Psip * [0 0;0 0; eye(2)];
A = AB(1:2,1:2);
B = AB(3:4,1:2);
A*inv(B)

Psip = Psip^2;
AB = Psip * [0 0;0 0; eye(2)];
A = AB(1:2,1:2);
B = AB(3:4,1:2);
A*inv(B)

Psip = Psip^2;
AB = Psip * [0 0;0 0; eye(2)];
A = AB(1:2,1:2);
B = AB(3:4,1:2);
A*inv(B)

Psip = Psip^2;
AB = Psip * [0 0;0 0; eye(2)];
A = AB(1:2,1:2);
B = AB(3:4,1:2);
A*inv(B)
```

```
function AltRLC

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 6.14

% Sequential Kalman filter for RLC circuit.

R = 100;
L = 1;
C = 1;
A = [-2/R/C 1/C ; -1/L 0];
B = [1/R/C ; 1/L];
H = eye(2); % measurement matrix
qc = 9; % variance of continuous time voltage input
Qc = B * qc * B'; % variance of continuous time process noise
R = eye(2); % covariance of discrete time measurement noise
tf = 2; % end time of simulation

T = 0.1; % discretization step size
F = expm(A*T); % state transition matrix
Q = Qc * T; % variance of discrete time process noise

x = [0 ; 0]; % initial state
[D, Lambda] = eig(Q); % eigendata of correlated process noise covariance
sqrtLambda = sqrt(Lambda);

xhatplus = x; % initial state estimate
Pplus = 0.1 * eye(2); % initial a posteriori estimation covariance
var0 = Pplus(1,1); % initial a posteriori variance of first state

% Initialize arrays
xArr = x;
xhatArr = xhatplus;
yArr = [];
varArr = [];

for t = T : T : tf
    % System simulation
    v = sqrtLambda * randn(2,1);
    w = D * v; % correlated process noise sample
    x = F * x + w; % discretized system dynamics
    y = H * x + sqrt(R) * randn(2,1); % measurement
    % Sequential Kalman filter simulation
    Pminus = F * Pplus * F' + Q;
    xhatminus = F * xhatplus;
    xhatplus = xhatminus;
    Pplus = Pminus;
    var(1,1) = Pplus(1,1);
    for i = 1 : 2
        K = Pplus * H(i,:)' / (H(i,:) * Pplus * H(i,:)' + R(i,i));
```

```
        xhatplus = xhatplus + K * (y(i) - H(i,:) * xhatplus);
        Pplus = Pplus - K * H(i,:) * Pplus;
        var(i+1,1) = Pplus(1,1);
    end
    % Save data in arrays
    xArr = [xArr x];
    xhatArr = [xhatArr xhatplus];
    yArr = [yArr y];
    varArr = [varArr var];
end

close all;

% Plot the true, estimated, and measured capacitor voltage
t = 0 : T : tf;
figure; hold on; set(gcf,'Color','White');
plot(t, xArr(1,:), 'r-', t, xhatArr(1,:), 'k:');
t = T : T : tf;
plot(t, yArr(1,:), 'b--');
set(gca,'FontSize',12);
xlabel('time'); ylabel('capacitor voltage');
legend('true', 'estimated', 'measured');
box('on');

% Plot the variance of the estimated capacitor voltage
t = T : T : tf;
figure; hold on; set(gcf,'Color','White');
plot(0, var0, 'o');
for i = 1 : 3
    plot(t, varArr(i,:), 'o');
end
tAll = [0 reshape([t ; t ; t], 1, 3*length(t))];
varAll = [var0 reshape(varArr, 1, 3*size(varArr,2))];
plot(tAll, varAll, 'k:');
set(gca,'FontSize',12);
xlabel('time'); ylabel('variance of capacitor voltage estimate');
box('on');
```

```
function Pitch

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 6.15

% Square root filter for aircraft state estimation

Mq = -0.568;
Ma = 17.98;
LaV = 1.237;
MdE = 0.175;
MdF = MdE;
LdEV = 0.001;
LdFV = LdEV;
qc = 0.001;

A = [Mq Ma; 1 -LaV];
B = [MdE MdF; -LdEV -LdFV];
Qc = [Ma; -LaV] * qc * [Ma -LaV];
H = eye(2);
R = diag([0.3 0.3]);
sqrtR = sqrt(R);

T = 0.01;
kf = 100;
F = expm(A*T);
G = (F * inv(A) - inv(A)) * B;
P = 0.01 * eye(2); % covariance of initial estimation error
S = sqrt(P);

Q = T * Qc;
[D, Lambda] = eig(Q);
sqrtLambda = sqrt(Lambda);
if norm(D*D' - eye(size(Q)), 'fro') > 100*eps
    disp('cannot find orthonormal eigenvector matrix for Q');
    return;
end

x = [0 ; 0]; % initial state
xhat = x; % initial state estimate

% Initialize arrays
xArr = x;
xhatArr = xhat;
yArr = x;
PArr(:,:,1) = P;

for k = 1 : kf
    % System simulation
    u = [0; 0];
```

```
    x = F * x + G * u;
    w = D * sqrtLambda * randn(2,1);
    x = x + w;
    y = H * x + sqrtR * randn(2,1);
    % Square root filter
    [W, T] = House1([S' * F' ; mychol(Q)]);
    S = W';
    xhat = F * xhat + G * u;
    for i = 1 : 2
        phi = S' * H(i,:)';
        a = 1 / (phi' * phi + R(i,i));
        gamma = 1 / (1 + sqrt(a * R(i,i)));
        S = S - S * a * gamma * phi * phi';
        K = a * S * phi;
        xhat = xhat + K * (y(i) - H(i,:) * xhat);
    end
    % Save data in arrays
    xArr = [xArr x];
    yArr = [yArr y];
    xhatArr = [xhatArr xhat];
    PArr(:,:,k+1) = S * S';
end

close all;
figure; set(gcf,'Color','White');
k = 0 : kf;
plot(k, xArr(1,:), 'k-', k, xhatArr(1,:), 'r--', k, yArr(1,:), 'b:');
set(gca,'FontSize',12);
xlabel('time'); ylabel('pitch rate');
legend('true', 'estimated', 'measured');

figure; set(gcf,'Color','White');
plot(k, xArr(2,:), 'k-', k, xhatArr(2,:), 'r--', k, yArr(2,:), 'b:');
set(gca,'FontSize',12);
xlabel('time'); ylabel('angle of attach');
legend('true', 'estimated', 'measured');

figure; set(gcf,'Color','White');
plot(k, squeeze(PArr(1,1,:)), 'r:', k, squeeze(PArr(2,2,:)), 'b-');
set(gca,'FontSize',12);
xlabel('time'); ylabel('estimation error covariances');
legend('pitch rate', 'angle of attack');
```

```
function [S] = mychol(P)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 6.15

% Cholesky Decomposition
% Compute S such that S*S' = P for a (2 x 2) matrix P

S(2,2) = sqrt(P(2,2));
S(1,2) = P(1,2) / S(2,2);
S(2,1) = 0;
S(1,1) = sqrt(P(1,1) - S(1,2)^2);
if ~isreal(S(1,1))
    S(1,1) = 0;
end
```

```
function ColorScalar

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 7.10

% Kalman filter with colored noise

Q = 1; % variance of process noise
Qv = 1; % variance of white noise that affects measurement

v = 0; % measurement noise
x0 = 0; % initial state
x = x0;
xhatw0 = 0;
xhatw = xhatw0; % state estimate using filter that assumes white noise
xhatc = [xhatw0 ; 0]; % state estimate using filter that assumes colored
noise
% white Kalman filter steady state solution
Pw = (-7 + sqrt(65)) / 2;
Kw = Pw / Qv;
% colored Kalman filter steady stae solution
F = [1/2 0; 0 1/2];
H = [1 1];
Pcminus = dare(F', H', [Q 0; 0 Qv], 0);
Kc = Pcminus * H' * inv(H * Pcminus * H');

% initialize arrays
ErrwArray = []; % estimation error of filter that assumes white noise
ErrcArray = []; % estimation error of filter that assumes colored noise
xhatwArray = []; % estimate of filter that assumes white noise
xhatcArray = []; % estimate of filter that assumes colored noise
xArray = []; % true state

kf = 2000;
for k = 1 : kf
    % System
    w = randn;
    x = x / 2 + sqrt(Q) * w;
    v = v / 2 + sqrt(Qv) * randn;
    y = x + v;
    % Filters
    xhatw = xhatw / 2 + Kw * (y - xhatw / 2);
    xhatc = F * xhatc;
    xhatc = xhatc + Kc * (y - H * xhatc);
    % Save data in arrays
    ErrwArray = [ErrwArray x-xhatw];
    ErrcArray = [ErrcArray x-xhatc(1)];
    xhatwArray = [xhatwArray xhatw];
    xhatcArray = [xhatcArray xhatc];
    xArray = [xArray x];
```

```
end

Err2wAve = norm(ErrwArray,2)^2 / length(ErrwArray);
disp(['white E(e^2) = ', num2str(Err2wAve)]);
Err2cAve = norm(ErrcArray,2)^2 / length(ErrcArray);
disp(['colored E(e^2) = ', num2str(Err2cAve)]);

% Analytical estimation error variance for white Kalman filter
Ee2w = (1 - Kw)^2 * Q + Kw^2 * 4 * Qv / 3;
Ee2w = Ee2w + Kw * (1 - Kw) * 2 * Kw * Qv / 3;
Ee2w = Ee2w / (1 - (1 - Kw)^2 / 4);
disp(['Analytical error variance for white Kalman filter = ', num2str
(Ee2w)]);
% Analytical estimation error variance for colored Kalman filter
Pcplus = Pcminus - Kc * H * Pcminus;
disp(['Analytical error variance for colored Kalman filter = ', num2str
(Pcplus(1,1))]);

close all;
k = 1 : kf;
plot(k, xArray, 'k-', k, xhatwArray, 'b:', k, xhatcArray, 'r--');
legend('true', 'white filter', 'colored filter');

figure;
plot(k, ErrwArray, 'k-', k, ErrcArray, 'b:')
legend('white filter', 'colored filter');
```

```
function AlphaBeta1

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 7.11

% Plot alpha and beta parameters as a function of lambda (tracking index)

lambda = logspace(-3,3,100);
alpha = -1 / 8 * (lambda.^2 + 8 * lambda - (lambda + 4) .* sqrt(lambda.^2
+ 8 * lambda));
beta = 1 / 4 * (lambda.^2 + 4 * lambda - lambda .* sqrt(lambda.^2 + 8 *
lambda));
close all;
semilogx(lambda, alpha, 'k-', lambda, beta, 'r:');
set(gca,'FontSize',12); set(gcf,'Color','White');
legend('\alpha', '\beta');
xlabel('\lambda');
axisDef = axis;
axis([1e-3 1e+3 axisDef(3) axisDef(4)]);
```

```
function AlphaBetaGamma1

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 7.12

% Plot alpha, beta, and gamma parameters as a function of lambda
% (tracking index) for the alpha-beta-gamma filter.

lambda = logspace(-3,3,100);
b = lambda / 2 - 3;
c = lambda / 2 + 3;
p = c - b.^2 / 3;
q = 2 * b.^3 / 27 - b .* c / 3 - 1;
z = ((-q + sqrt(q.^2 + 4 * p.^3 / 27)) / 2).^(1/3);
s = z - p / 3 ./ z - b / 3;
alpha = 1 - s.^2;
beta = 2 * (1 - s).^2;
gamma = 2 * lambda .* s;
close all;
semilogx(lambda, alpha, 'k-', lambda, beta, 'r:', lambda, gamma, 'b--');
set(gca,'FontSize',12); set(gcf,'Color','White');
legend('\alpha', '\beta', '\gamma');
xlabel('\lambda');
axisDef = axis;
axis([1e-3 1e+3 axisDef(3) axisDef(4)]);
```

```
function Drug

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 7.13

% Constrained Kalman filter for drug quantity estimation.

k1 = 1;
k2 = 1;
A = [-k1 0; k1 -k2];
B = [1; 0];
H = [0 1];

tf = 3; % simulation time (days)
Qc = [0 0; 0 0]; % variance of continuous time process noise
R = 1; % variance of measurement noise
x = [0.8; 0]; %initial state
xmax = 1; % maximum possible value of x(1)
xmin = 0.8; % minimum possible value of x(1)

% initialize Kalman filter
xhat = x;
P = [1 0; 0 1];
% initialize constrained Kalman filter (W = I)
xhatI = x;
D = [1 0];
% initialize truncated Kalman filter
xhatT = x;

% discretization
dt = 1/24; % step size (days)
Q = Qc * dt;
F = expm(A*dt);
G = (F * inv(A) - inv(A)) * B;

% initialize arrays
xArr = x;
xhatArr = xhat;
xhatIArr = xhatI;
xhatTArr = xhatT;

for t = dt : dt : tf+dt/10
    % system simulation
    u = 1;
    x = F * x + G * u + sqrt(Q) * randn(size(x));
    y = H * x + sqrt(R) * randn;
    % Kalman filter
    P = F * P * F' + Q;
    K = P * H' * (H * P * H' + R)^(-1);
    xhat = F * xhat + G * u;
```

```
    xhat = xhat + K * (y - H * xhat);
    P = P - K * H * P;
    % Constrained Kalman filter (W = I)
    if D * xhat > xmax
        xhatI = xhat - D' * inv(D * D') * (D * xhat - xmax);
    elseif D * xhat < xmin
        xhatI = xhat - D' * inv(D * D') * (D * xhat - xmin);
    else
        xhatI = xhat;
    end
    % Truncated Kalman filter
    xhatT = xhat;
    PT = P;
    [T, W] = eig(PT);
    rho = rhoCalc(sqrt(W)*T'*D', sqrt(D*PT*D'));
    cki = (xmin - D * xhatT) / sqrt(D*PT*D');
    dki = (xmax - D * xhatT) / sqrt(D*PT*D');
    alpha = sqrt(2/pi) / (erf(dki/sqrt(2)) - erf(cki/sqrt(2)));
    mu = alpha * (exp(-cki^2/2) - exp(-dki^2/2));
    sigma2 = alpha * (exp(-cki^2/2) * (cki - 2 * mu) - exp(-dki^2/2) *
(dki - 2 * mu)) + mu^2 + 1;
    ztilde = [mu; 0];
    ztildeCov = diag([sigma2 1]);
    xhatT = T * sqrt(W) * rho' * ztilde + xhatT;
    PT = T * sqrt(W) * rho' * ztildeCov * rho * sqrt(W) * T';
    % Save data
    xArr = [xArr x];
    xhatArr = [xhatArr xhat];
    xhatIArr = [xhatIArr xhatI];
    xhatTArr = [xhatTArr xhatT];
end

% plot data
close all;
t = 0 : dt : tf;

figure;
plot(t, xArr(1,:), 'b-', t, xhatArr(1,:), 'r:', t, xhatIArr(1,:), 'k--',
t, xhatTArr(1,:), 'm-.');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('days'); ylabel('drug mass');
legend('true', 'unconstrained', 'projected', 'truncated');

figure;
plot(t, abs(xArr(1,:)-xhatArr(1,:)), 'b-', t, abs(xArr(1,:)-xhatIArr
(1,:)), 'r:', t, abs(xArr(1,:)-xhatTArr(1,:)), 'm--');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('days'); ylabel('drug mass estimation error');
legend('unconstrained', 'projected', 'truncated');
```

```
function [rho] = rhoCalc(x, y1)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 7.13

% Compute an n x n matrix rho such that
% rho * x = [y1 ... 0]' and rho * rho' = I.
% This assumes that x is of the form W^(1/2)*T^T*phi, and
% y1 is of the form (phi^T*Q*phi)^(1/2).
% This function can be used as part of the pdf truncation algorithm
% for the inequality constrained Kalman filter.

n = length(x);
rho = zeros(n, n);
rho(1,:) = x' / y1;
I = eye(n);
for k = 2 : n
    rho(k,:) = I(k,:);
    for i = 1 : k-1
        rho(k,:) = rho(k,:) - I(k,:) * rho(i,:)' * rho(i,:);
    end
    if norm(rho(i,:)) < 100 * eps
        rho(k,:) = I(1,:);
        for i = 1 : k-1
            rho(k,:) = rho(k,:) - I(1,:) * rho(i,:)' * rho(i,:);
        end
    end
    rho(k,:) = rho(k,:) / norm(rho(k,:));
end
```

```
function ProcessNoise(dt)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 8.11

% This program shows that a simulation gives the same state variance
% regardless of the simulation step size dt, as long as dt is small
% enough to give a stable simulation, and as long as the variance of
% the process noise is adjusted correctly with changes in dt.
% INPUT: dt = simulation step size

if ~exist('dt', 'var')
    dt = 0.5;
end

Q0 = 2;
Q = Q0 * dt;
tf = 10; % final simulation time

N = 5000; % number of Monte Carlo runs to estimate E(x^2(tf))
Expx2 = 0;
for i = 1 : N
    x = 0; % initial state
    for t = dt : dt : tf + dt/10
        x = x + sqrt(Q) * randn;
    end
    Expx2 = Expx2 + x^2;
end
Expx2 = Expx2 / N;
disp(['dt = ', num2str(dt), ' : Exp(x^2) = ', num2str(Expx2)]);
```

```
function Discretize(dt)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 8.12

% Continuous time Kalman filter for a scalar system.
% INPUT: dt = integration step size

if ~exist('dt', 'var')
    dt = 0.4;
end

tf = 1000; % simulation length
Qc = 2; % continuous time process noise
Rc = 1; % continuous time measurement noise
x = 0; % initial state
xhat = 0; % initial Kalman estimate
P = 0; % initial Kalman filter error covariance

xArr = [x];
xhatArr = [xhat];
EstErr = 0;
for t = dt : dt : tf+dt/10
    % System simulation
    x = exp(-dt) * x + sqrt(Qc*dt) * randn;
    y = x + sqrt(Rc/dt) * randn;
    % Kalman filter simulation
    K = P * inv(Rc);
    xhatdot = -xhat + K * (y - xhat);
    xhat = xhat + xhatdot * dt;
    Pdot = -P^2 - 2 * P + 2;
    P = P + Pdot * dt;
    % Compute the estimation error
    EstErr = EstErr + (x - xhat)^2;
end
EstErr = EstErr / (tf / dt);
disp(['dt = ', num2str(dt), ' : Kalman Est Var = ', num2str(EstErr)]);
```

```
function ContKFGyro(SSFlag, M, MFilter)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problems 8.13 and 8.14

% Continuous time Kalman filter
% INPUTS:
%     SSFlag = steady state flag
%     M = process noise / measurement noise correlation
%     MFilter = value of M that is used by the Kalman filter

if ~exist('SSFlag', 'var')
    SSFlag = 1;
end
if ~exist('M', 'var')
    M = 0;
end
if ~exist('MFilter', 'var')
    MFilter = 0;
end

A = [0 1 ; 0 0];
C = [1 0];
q = 2;
Q = [0 0 ; 0 q];
R = 3;

dt = 0.01;
tf = 10;

x = [0 ; 0];
xhat = x;
P = [1 0 ; 0 1];

xArr = x;
xhatArr = xhat;
k = 1;
PArr(:,:,k) = P;

Pss = [sqrt(2*R^(3/2)*q^(1/2)) sqrt(R*q) ; sqrt(R*q) sqrt(2*R^(1/2)*q^
(3/2))];
Kss = Pss * C' * inv(R);

[v, d] = eig([q*dt M ; M R/dt]);

for t = dt : dt : tf + dt/10
    noise = v * sqrt(d) * randn(2,1);
    % System simulation
    xdot = A * x + sqrt(Q*dt) * randn(size(x));
    xdot = A * x + noise(1);
```

```
    x = x + xdot * dt;
    y = C * x + sqrt(R/dt) * randn;
    y = C * x + noise(2);
    % Kalman filter
    if SSFlag
        K = Kss;
        Pdot = A * P + P * A' - P * C' * inv(R) * C * P + Q;
    else
        K = (P * C' + MFilter) * inv(R);
        Pdot = A * P + P * A' - K * R * K' + Q;
    end
    xhatdot = A * xhat + K * (y - C * xhat);
    xhat = xhat + xhatdot * dt;
    P = P + Pdot * dt;
    % Save data for plotting
    xArr = [xArr x];
    xhatArr = [xhatArr xhat];
    k = k + 1;
    PArr(:,:,k) = P;
end

close all;
figure;
t = 0 : dt : tf + dt/10;
plot(t, xArr(1,:), 'r-', t, xhatArr(1,:), 'b:');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('time');
legend('true x(1)', 'estimated x(1)');

figure;
plot(t, squeeze(PArr(1,1,:)), 'r-', t, squeeze(PArr(2,2,:)), 'b:', t,
squeeze(PArr(1,2,:)), 'k--');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('time');
legend('P(1,1)', 'P(2,2)', 'P(1,2)');

disp(['Theoretical (M=0) steady state covariance = ', num2str(sqrt(2*R^
(3/2)*q^(1/2))), ', ', ...
    num2str(sqrt(R*q)), ', ', num2str(sqrt(2*R^(1/2)*q^(3/2)))]);
disp(['Numerical steady state covariance = ', num2str(P(1,1)), ', ',
num2str(P(1,2)), ...
    ', ', num2str(P(2,2))]);

EstErr = xArr - xhatArr;
PExp = EstErr * EstErr' / length(t);
disp(['Simulation error covariance = ', num2str(PExp(1,1)), ', ', num2str
(PExp(1,2)), ...
    ', ', num2str(PExp(2,2))]);
```

```
function ContEx

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 8.15

% Continuous Kalman filter for a two-state problem.

close all;
figure;
subplot(2,2,1);
CAREProblem(1,2,1,1,1,1,1,1,0,1);
subplot(2,2,2);
CAREProblem(-1,-1,1,2,4,1,1,1,0,1);
subplot(2,2,3);
CAREProblem(1,1,1,2,4,1,1,1,0,1);
subplot(2,2,4);
CAREProblem(1,1,1,2,4,1,1,0,0,0);
return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function CAREProblem(a1, a2, q11, q12, q22, r1, r2, p11, p12, p22)

disp('CARE Solution:')
disp(care([a1 0;0 a2]', [1 0;0 1]', [q11 q12;q12 q22], [r1 0;0 r2]));

PArr = [];
dt = 0.01;
tf = 4;
% Plot the time varying solution.
for t = 0 : dt : tf
    p11dot = 2 * a1 * p11 - p11^2 / r1 - p12^2 / r2 + q11;
    p12dot = (a1 + a2) * p12 - p11 * p12 / r1 - p12 * p22 / r2 + q12;
    p22dot = 2 * a2 * p22 - p12^2 / r1 - p22^2 / r2 + q22;
    p11 = p11 + p11dot * dt;
    p12 = p12 + p12dot * dt;
    p22 = p22 + p22dot * dt;
    PArr = [PArr ; p11 p12 p22];
end
t = 0 : dt : tf;
plot(t, PArr(:, 1), 'r:', t, PArr(:, 2), 'k--', t, PArr(:, 3), 'b-');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('time');
legend('P(1,1)', 'P(1,2)', 'P(2,2)');
% Compute the steady state solution.
Cond1 = (a1 ~= a2) && (q12 ~= 0);
Cond2 = (a1 == a2) && (a1 < 0) && (q12 ~= 0) && (q11*q22-q12*q12 == 0);
Cond3 = (a1 == a2) && (a1 > 0) && (q12 ~= 0) && (q11*q22-q12*q12 == 0);
if Cond1 || Cond3
    gamma1 = q11 / r1 + a1^2;
    gamma2 = q22 / r2 + a2^2;
```

```
    p12 = q12 / ( gamma1 + gamma2 + 2 * ( gamma1 * gamma2 - q12^2 / r1 /
r2 )^(1/2) )^(1/2);
    p11 = r1 * ( a1 + ( gamma1 - p12^2 / r1 / r2 )^(1/2) );
    p22 = r2 * ( a2 + ( gamma2 - p12^2 / r1 / r2 )^(1/2) );
    disp('Analytical Solution:');
    disp([p11 p12;p12 p22]);
    lambda = eig([p11 p12; p12 p22]);
    disp(['Eigenvalues of P = ', num2str(lambda(1)), ', ', num2str(lambda
(2))]);
end
if Cond2 || Cond3
    gamma3 = -a1 + ( a1^2 + q11 / r1 + q22 / r2 )^(1/2);
    p11 = q11 / gamma3;
    p22 = q22 / gamma3;
    p12 = q12 / gamma3;
    disp('Analytical Solution:');
    disp([p11 p12;p12 p22]);
    lambda = eig([p11 p12; p12 p22]);
    disp(['Eigenvalues of P = ', num2str(lambda(1)), ', ', num2str(lambda
(2))]);
end
if ~Cond1 && ~Cond2 && ~Cond3
    disp('Numerical Solution:');
    disp([p11 p12;p12 p22]);
    lambda = eig([p11 p12; p12 p22]);
    disp(['Eigenvalues of P = ', num2str(lambda(1)), ', ', num2str(lambda
(2))]);
end
```

```
function FixPt1

% Optimal State Estimation Solution Manual, by Dan Simon
% Problems 9.4, 9.5

% Fixed point smoother for a scalar problem.
% Verify the results of the written solution.

Q = 1;
R = 2 * Q;
R = 12 * Q;

P = (Q + sqrt(Q*Q+4*Q*R)) / 2;
Sigma = P;
Pi = P;
F = 1;
H = 1;
for k = 1 : 100
    L = F * P * H' * inv(H * P * H' + R);
    lambda = Sigma * H' * inv(H * P * H' + R);
    P = F * P * (F - L * H)' + Q;
    Pi = Pi - Sigma * H' * lambda';
    Sigma = Sigma * (F - L * H)';
end
```

```
function RTS2(Qc)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problems 9.15 and 9.16

% RTS smoother applied to a second order system.
% INPUT: Qc = continuous time noise covariance

if ~exist('Qc', 'var')
    Qc = 0.01;
end

w = 6; % natural frequency of the system
z = 0.16; % damping ratio of the system
A = [0 1 ; -w^2 -2*z*w]; % system matrix
H = [1 0]; % measurement matrix
Qc = [0 0 ; 0 Qc]; % process noise covariance
R = 1e-4; % measurement noise covariance

dt = 0.5; % discretization step size
Q = Qc * dt; % discrete time process noise covariance
F = expm(A * dt); % state transition matrix

x = [1 ; 1]; % initial state
xhatf = x; % initial forward state estimate
Pfplus = diag([1e-5 1e-2]); % initial forward estimation covariance
PfplusArr(:,:,1) = Pfplus;
xhatfArr(:,1) = xhatf;
xhatfminusArr = [];

kf = 10;
for k = 1 : kf
    % system simulation
    x = F * x + sqrt(Q) * randn(2,1);
    y = H * x + sqrt(R) * randn;
    % forward Kalman filter
    Pfminus = F * Pfplus * F' + Q;
    Kf = Pfminus * H' * inv(H * Pfminus * H' + R);
    xhatfminus = F * xhatf;
    xhatf = xhatfminus + Kf * (y - H * xhatfminus);
    Pfplus = Pfminus - Kf * H * Pfminus;
    % save covariances for later
    PfminusArr(:,:,k) = Pfminus;
    PfplusArr(:,:,k+1) = Pfplus;
    xhatfminusArr = [xhatfminusArr xhatfminus];
    xhatfArr = [xhatfArr xhatf];
end
% RTS equations
xhat = xhatf;
P = Pfplus;
```

```
PArr(:,:,kf+1) = P;
for k = kf-1 : -1 : 0
    Ifminus = inv(PfminusArr(:,:,k+1));
    K = PfplusArr(:,:,k+1) * F' * Ifminus;
    P = PfplusArr(:,:,k+1) - K * (PfminusArr(:,:,k+1) - P) * K';
    xhat = xhatfArr(:,k+1) + K * (xhat - xhatfminusArr(:,k+1));
    % save covariance for later
    PArr(:,:,k+1) = P;
end

% Plot stuff
close all;
k = 0 : kf;
figure;
subplot(2,1,1);
plot(k, squeeze(PfplusArr(1,1,:)), 'k-', k, squeeze(PArr(1,1,:)), 'r:');
set(gca,'FontSize',12); set(gcf,'Color','White');
ylabel('first state');
legend('forward covariance', 'smoothed covariance');
subplot(2,1,2);
plot(k, squeeze(PfplusArr(2,2,:)), 'k-', k, squeeze(PArr(2,2,:)), 'r:');
legend('forward covariance', 'smoothed covariance');
set(gca,'FontSize',12);
xlabel('time');
ylabel('second state');
```

```
function [Err0] = PopSmooth(PlotFlag)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 9.17

% RTS fixed interval smoother for population estimation.
% INPUT: PlotFlag = flag indicating whether or not to plot stuff
% OUTPUT: Err0 = estimation error of smoothed estimate at initial time

if ~exist('PlotFlag', 'var')
    PlotFlag = true;
end

kf = 10; % simulation length

F = [1/2 2 ; 0 1];
Q = [0 0 ; 0 10];
H = [1 0];
R = 10;
Pfplus = [500 0 ; 0 200]; % initial estimation error covariance

x0 = [650 ; 250]; % initial state
x = x0; % state
xhatf = [600 ; 200]; % initial estimate

% Initialize arrays
xArr = x;
xhatfminusArr = [];
xhatfArr = xhatf;
PfplusArr(:,:,1) = Pfplus;

for k = 1 : kf
    % System simulation
    x = F * x + sqrt(Q) * randn(2,1);
    y = H * x + sqrt(R) * randn;
    % Kalman filter simulation
    Pfminus = F * Pfplus * F' + Q;
    Kf = Pfminus * H' * inv(H * Pfminus * H' + R);
    xhatfminus = F * xhatf;
    xhatf = xhatfminus + Kf * (y - H * xhatfminus);
    Pfplus = Pfminus - Kf * H * Pfminus;
    % Save data for plotting
    xArr = [xArr x];
    xhatfminusArr = [xhatfminusArr xhatfminus];
    xhatfArr = [xhatfArr xhatf];
    PfminusArr(:,:,k) = Pfminus;
    PfplusArr(:,:,k+1) = Pfplus;
end

% RTS equations
```

```
xhat = xhatf;
xhatArr(:,:,kf+1) = xhat;
P = Pfplus;
PArr(:,:,kf+1) = P;
for k = kf-1 : -1 : 0
    Ifminus = inv(PfminusArr(:,:,k+1));
    K = PfplusArr(:,:,k+1) * F' * Ifminus;
    P = PfplusArr(:,:,k+1) - K * (PfminusArr(:,:,k+1) - P) * K';
    xhat = xhatfArr(:,k+1) + K * (xhat - xhatfminusArr(:,k+1));
    % save for later
    PArr(:,:,k+1) = P;
    xhatArr(:,:,k+1) = xhat;
end
Err0 = x0 - xhat;

if ~PlotFlag
    return;
end

close all;
figure; k = 0 : kf;
subplot(2,1,1);
plot(k, xArr(1,:), 'r-', k, xhatfArr(1,:), 'b:', k, xhatArr(1,:), 'k--');
set(gca,'FontSize',12); set(gcf,'Color','White');
legend('true', 'forward estimate', 'smoothed estimate');
ylabel('first state');
subplot(2,1,2);
plot(k, xArr(2,:), 'r-', k, xhatfArr(2,:), 'b:', k, xhatArr(2,:), 'k--');
set(gca,'FontSize',12); set(gcf,'Color','White');
ylabel('second state');
xlabel('time');

% Plot stuff
figure;
subplot(2,1,1);
plot(k, squeeze(PfplusArr(1,1,:)), 'k-', k, squeeze(PArr(1,1,:)), 'r:');
set(gca,'FontSize',12); set(gcf,'Color','White');
ylabel('first state');
legend('forward a posteriori covariance', 'smoothed covariance');
subplot(2,1,2);
plot(k, squeeze(PfplusArr(2,2,:)), 'k-', k, squeeze(PArr(2,2,:)), 'r:');
legend('forward a posteriori covariance', 'smoothed covariance');
set(gca,'FontSize',12);
xlabel('time');
ylabel('second state');

Cov1Improve = 100 * (PfplusArr(1,1,1) - PArr(1,1,1)) / PfplusArr(1,1,1);
Cov2Improve = 100 * (PfplusArr(2,2,1) - PArr(2,2,1)) / PfplusArr(2,2,1);
disp(['Improvement in 1st state covariance = ', num2str(Cov1Improve), '
%']);
```

```
disp(['Improvement in 2nd state covariance = ', num2str(Cov2Improve), '↙
%']);

PArr(:,:,1)
```

```
function PopSmooth1(PlotFlag)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 9.17

% Forward-backward fixed interval smoother for population estimation.
% INPUT: PlotFlag = flag indicating whether or not to plot stuff

if ~exist('PlotFlag', 'var')
    PlotFlag = true;
end

kf = 10; % simulation length

F = [1/2 2 ; 0 1];
Q = [0 0 ; 0 10];
H = [1 0];
R = 10;

x0 = [650 ; 250]; % initial state
x = x0; % state
xhatf = [600 ; 200]; % initial estimate

yArr = [];
% Simulate the system and collect the measurements.
% Use a constant commanded acceleration.
for k = 0 : kf
    x = F * x + sqrt(Q) * randn(2,1);
    y = H * x + sqrt(R) * randn;
    yArr = [yArr y];
end

% Obtain the forward estimate.
Pfplus = [500 0 ; 0 200]; % initial estimation error covariance
PfplusArr(:,:,1) = Pfplus;
xhatf = x0; % initial state estimate
for k = 1 : kf
   % Extrapolate the most recent state estimate to the present time.
   xhatf = F * xhatf;
   % Form the Innovation vector.
   Inn = yArr(k) - H * xhatf;
   % Compute the covariance of the a priori estimation error
   Pfminus = F * Pfplus * F' + Q;
   % Compute the covariance of the Innovation.
   CovInn = H * Pfminus * H' + R;
   % Form the Kalman Gain matrix.
   K = Pfminus * H' * inv(CovInn);
   % Compute the covariance of the estimation error.
   Pfplus = Pfminus - K * H * Pfminus;
   % Update the state estimate.
```

```
    xhatf = xhatf + K * Inn;
    % Save some parameters for plotting later.
    PfplusArr(:,:,k+1) = Pfplus;
end

% Obtain the backward estimate.
% The initial backward information matrix Ibminus needs to be set to a
% small nonzero matrix so that the first calculated Ibplus is invertible.
Ibminus = [1e-12 0; 0 1e-12];
PbArr(:,:,kf+1) = inv(Ibminus);
sminus = zeros(size(x)); % initial backward modified estimate
Q = Q + [0 0; 0 0];
for k = kf : -1 : 1
    Ibplus = Ibminus + H' * inv(R) * H;
    splus = sminus + H' * inv(R) * yArr(k);
    % The following line does not work unless Sw is invertible.
    %Ibminus = inv(Sw) - inv(Sw) * inv(a) * inv(Ibplus + inv(a)' * inv(Sw)
* inv(a)) * inv(a)' * inv(Sw);
    %Ibminus = inv(inv(F) * inv(Ibplus) * inv(F)' + inv(F) * Q * inv(F)');
    Ibminus = F' * inv(Q + inv(Ibplus)) * F;
    sminus = Ibminus * inv(F) * inv(Ibplus) * splus;
    % Save some parameters for plotting later.
    Pbminus = inv(Ibminus);
    PbArr(:,:,k) = Pbminus;
end

% Compute the smoothed estimation covariance.
for k = 0 : kf
    P(:,:,k+1) = inv(inv(PfplusArr(:,:,k+1)) + inv(PbArr(:,:,k+1)));
end

close all;
figure;
set(gca,'FontSize',12); set(gcf,'Color','White');
k = 0 : kf;

subplot(2,1,1);
plot(k, squeeze(PfplusArr(1,1,:)), 'r-', k, squeeze(PbArr(1,1,:)), 'b--',
k, squeeze(P(1,1,:)), 'k:');
axis([0 kf 0 max(PfplusArr(1,1,:))]);
legend('forward', 'backward', 'smoothed');
ylabel('first state');

subplot(2,1,2);
plot(k, squeeze(PfplusArr(2,2,:)), 'r-', k, squeeze(PbArr(2,2,:)), 'b--',
k, squeeze(P(2,2,:)), 'k:');
axis([0 kf 0 max(PfplusArr(2,2,:))]);
ylabel('second state');
```

```
function Tire(Fmodel)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 10.14

% Tire tread estimation.
% INPUT: Fmodel = assumed value of F matrix. This may or may not match
% the true value of the F matrix.

if ~exist('Fmodel', 'var')
    Fmodel = 0.8;
end
tf = 10; % simulation length
Ftrue = 0.8;
Q = 0.01;
H = 1;
R = 0.01;
Pplus = 0; % initial estimation error covariance
ResMeanArr = [];
ResStdArr = [];
for NumTires = 1 : 1000
    x = 1; % initial state
    xhat = 1; % initial estimate
    % Initialize arrays
    ResArr = [];
    for k = 1 : tf
        % System simulation
        x = Ftrue * x + sqrt(Q) * randn;
        y = H * x + sqrt(R) * randn;
        % Kalman filter simulation
        Pminus = Fmodel * Pplus * Fmodel' + Q;
        K = Pminus * H' * inv(H * Pminus * H' + R);
        xhatminus = Fmodel * xhat;
        xhat = xhatminus + K * (y - H * xhatminus);
        Pplus = (1 - K * H) * Pminus;
        % Save data
        ResArr = [ResArr y - xhatminus];
    end
    ResMeanArr = [ResMeanArr mean(ResArr)];
    ResStdArr = std(ResArr, 1);
end
disp(['Mean of residuals = ', num2str(mean(ResMeanArr))]);
% Compare the experimental and theoretical standard deviation of the
residual
disp(['Experimental std dev of residual = ', num2str(mean(ResStdArr))]);
disp(['Theoretical std dev of estimation error = ', num2str(sqrt(Pminus +
R))]);
```

```
function TireMulti

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 10.15

% Multiple model Kalman filtering for tire tread estimation.

global F
F = [0.8 ; 0.85 ; 0.9]; % possible values of F
NumTires = 100;
for i = 1 : NumTires
    [FhatArr(i,:), prArr(i,:,:), Fhat] = TireMultiSim(false);
end
Fhat1 = mean(FhatArr,1);
pr1(1,:) = mean(squeeze(prArr(:,1,:)),1);
pr1(2,:) = mean(squeeze(prArr(:,2,:)),1);
pr1(3,:) = mean(squeeze(prArr(:,3,:)),1);

close all
t = 0 : length(Fhat1) - 1;
figure;
subplot(2,1,1);
plot(t, Fhat1);
set(gca,'FontSize',12); set(gcf,'Color','White');
ylabel('Estimate of f');
subplot(2,1,2);
plot(t, pr1(1,:), 'b-', t, pr1(2,:), 'k--', t, pr1(3,:), 'r:');
set(gca,'FontSize',12); set(gcf,'Color','White');
ylabel('Probabilities of f');
xlabel('time step');
legend(['Probability that f = ',num2str(F(1))], ['Probability that f = ',
num2str(F(2))], ['Probability that f = ',num2str(F(3))]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [FhatArray, prArray, Fhat] = TireMultiSim(PlotFlag)

% INPUT: PlotFlag = 1 means plot results
% OUTPUT:   FhatArray = array of F estimates as a function of time
%           prArray = array of F probabilities as a function of time
%           Fhat = final estimate of F

if ~exist('PlotFlag', 'var')
    PlotFlag = true;
end

global F
kf = 10; % Length of simulation
N = size(F, 1); % number of parameter sets
pr = ones(size(F)) / N; % a priori probabilities
% Compute the initial estimate of wn
```

```
Fhat = 0;
for i = 1 : N
   Fhat = Fhat + F(i) * pr(i);
end
Q = 0.01; % discrete time process noise variance
R = 0.01; % discrete time measurement noise covariance
H = 1; % measurement matrix
x = 1; % initial state

n = length(x); % number of states
q = size(H, 1); % number of measurements

% Initialize the N state estimators
for i = 1 : N
   Pplus(i) = zeros(n);
   xhat(i) = x;
end

% Create arrays for later plotting
FhatArray = [Fhat];
prArray = [pr];
for k = 1 : kf
   % Simulate the system.
   x = F(1) * x + sqrt(Q) * randn;
   y = H * x + sqrt(R) * randn;
   % Run a separate Kalman filter for each parameter set.
   for i = 1 : N
      Pminus(i) = F(i) * Pplus(i) * F(i)' + Q;
      K = Pminus(i) * H' * inv(H * Pminus(i) * H' + R);
      xhat(i) = F(i) * xhat(i);
      r = y - H * xhat(i); % measurment residual
      S = H * Pminus(i) * H' + R; % covariance of measurement residual
      pdf(i) = exp(-r'*inv(S)*r/2) / ((2*pi)^(q/2)) / sqrt(det(S));
      xhat(i) = xhat(i) + K * (y - H * xhat(i));
      Pplus(i) = (eye(n) - K * H) * Pminus(i) * (eye(n) - K * H)' + K * R
* K';
   end
   % Compute the sum that appears in the denominator of the probability
expression.
   Prsum = 0;
   for i = 1 : N
      Prsum = Prsum + pdf(i) * pr(i);
   end
   % Update the probability of each parameter set.
   for i = 1 : N
      pr(i) = pdf(i) * pr(i) / Prsum;
   end
   % Compute the best state estimate and the best parameter estimate.
   xhatbest = 0;
   Fhat = 0;
```

```
   for i = 1 : N
      xhatbest = xhatbest + pr(i) * xhat(:,i);
      Fhat = Fhat + pr(i) * F(i);
   end
   % Save data for plotting.
   FhatArray = [FhatArray Fhat];
   prArray = [prArray pr];
end

if PlotFlag
    close all;
    t = 0 : kf;
    figure;
    plot(t, FhatArray);
    title('Estimate of f', 'FontSize', 12);
    set(gca,'FontSize',12); set(gcf,'Color','White');
    xlabel('time step');
    figure;
    plot(t, prArray(1,:), 'b-', t, prArray(2,:), 'k--', t, prArray(3,:),↙
'r:');
    title('Probabilities of f', 'FontSize', 12);
    set(gca,'FontSize',12); set(gcf,'Color','White');
    xlabel('time step');
    legend(['Probability that f = ',num2str(F(1))], ['Probability that f =↙
',num2str(F(2))], ['Probability that f = ',num2str(F(3))]);
end

return;
```

```
function RobustScalar

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 10.16

% Robust Kalman filter for a scalar system.

Q = 5;
sigma1 = 0.5;
R = 5;
sigma2 = 1;
Kmin = 0.3;
Kmax = 0.7;
K = Kmin : (Kmax - Kmin)/100 : Kmax;
P = ((1 - K).^2 * Q + K.^2 * R) ./ (2 * K - K.^2);
VardP = (sigma1^2 * (1 - K).^2 * Q^2 + sigma2^2 * K.^4 * R^2) ./ (2 * K -
K.^2);

close all;
figure;
plot(K, P+VardP, 'k:', K, VardP, 'b-', K, P, 'r--');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('K');
legend('P + E(\Delta P^2)', 'E(\Delta P^2)', 'P');

[temp, i] = min(P);
disp(['min P occurs at K = ', num2str(K(i))]);
[temp, i] = min(VardP);
disp(['min E(dP^2) occurs at K = ', num2str(K(i))]);
[temp, i] = min(P+VardP);
disp(['min P+E(dP^2) occurs at K = ', num2str(K(i))]);
```

```
function HinfRadio(kf, theta)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 11.14

% H-infinity estimators for radioactive decay problem.
% INPUTS:   kf = simulation length
%           theta = performance bound

if ~exist('kf', 'var')
    kf = 20;
end
if ~exist('theta', 'var')
    theta = 1;
end

P = 1;
Q = 1;
R = 1;
x = 0; % initial state
xhat = x; % H-infinity estimate
xArr = [x];
xhatArr = [xhat];
PArr = [P];
for k = 1 : kf
    % System simulation
    y = x + sqrt(R) * randn;
    x = x + sqrt(Q) * randn;
    % H-infinity filter simulation
    K = P * inv(1 - theta * P + P / R) / R;
    P = 1 / 4 * P * inv(1 - theta * P + P / R) + Q;
    xhat = xhat + K * (y - xhat);
    % Check for necessary conditions.
    if P <= 0
        disp('P <= 0'); return;
    elseif (1/P - theta + 1/R) <= 0
        disp('1/P - theta + 1/R <= 0'); return;
    end
    % Save data in arrays
    xArr = [xArr x];
    xhatArr = [xhatArr xhat];
    PArr = [PArr P];
end

k = 0 : kf;
close all;
figure; hold on;
subplot(2,1,1);
plot(k, xArr, 'k', 'LineWidth', 2.5); hold on;
subplot(2,1,1);
```

```
plot(k, xhatArr, 'r--');
set(gca,'FontSize',12); set(gcf,'Color','White');
legend('true state', 'H_{\infty} estimate');
xlabel('time'); ylabel('state value');
set(gca,'box','on');

subplot(2,1,2);
plot(k, PArr);
ylabel('P');

P = (1 - 4*theta + sqrt((1-4*theta)^2 + 64 * (1 - theta))) / 8 / (1 -
theta);
disp(['Theoretical steady state P = ', num2str(P)]);
```

```
function HinfVehicle(theta)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 11.15

% function HinfVehicle
% This m-file simulates a vehicle tracking problem.
% The vehicle state is estimated with a Kalman and an H-infinity filter.
% The state consists of the north and east position, and the
% north and east velocity of the vehicle.
% The measurement consists of north and east positions.
% INPUT:    theta = H-infinity performance parameter

if ~exist('theta', 'var')
    theta = 0.0005;
end
close all;
figure;
HinfVehicleSim(theta, 1, 1);
HinfVehicleSim(theta, 2, 2);
return;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function HinfVehicleSim(theta, uTrue, FigNum)

tf = 300; % final time (seconds)
T = 3; % time step (seconds)

Q = diag([4, 4, 1, 1]); % Process noise covariance (m, m, m/sec, m/sec)
Qsqrt = sqrt(Q);

R = diag([900, 900]); % Measurement noise covariance (m, m)
Rsqrt = sqrt(R);

angle = 0.9 * pi; % heading angle (measured CCW from east)

S = eye(4);
L = eye(4);
Sbar = L' * S * L;

% Define the initial state x, initial unconstrained filter estimate xhat,
% and initial constrained Kalman filter estimate xtilde.
x = [0; 0; 0; 0];
xhatK = x; % Unconstrained Kalman filter
xhatH = x; % Kalman filter with perfect measurements

% Initial estimation error covariance
PK = diag([0, 0, 0, 0]);
PH = PK;
```

```
F = [1 0 T 0 ; 0 1 0 T ; 0 0 1 0 ; 0 0 0 1]; % system matrix
B = [0; 0; T*sin(angle); T*cos(angle)]; % input matrix
H = [1 0 0 0 ; 0 1 0 0]; % measurement matrix

% Initialize arrays for saving data for plotting.
xarray = x;
xhatKarray = xhatK;
xhatHarray = xhatH;

randn('state', sum(100*clock)); % initialize random number generator

% Get the measurement at the initial time.
y = H * x + Rsqrt * randn(size(Rsqrt,1),1);

% Begin the simulation.
for t = T : T : tf+eps
    % Set the "known" input.
    uModel = 1;
    % Simulate the Kalman filter.
    temp = inv( eye(4) + H' * inv(R) * H * PK );
    KK = PK * temp * H' * inv(R);
    xhatK = F * xhatK + B * uModel + F * KK * (y - H * xhatK);
    PK = F * PK * temp * F' + Q;
    % Simulate the H-infinity filter.
    temp = eye(4) - theta * Sbar * PH + H' * inv(R) * H * PH;
    if cond(temp) > 1e10
        disp('ill conditioned H-infinity estimation problem');
        return;
    end
    temp = inv(temp);
    KH = PH * temp * H' * inv(R);
    xhatH = F * xhatH + B * uModel + F * KH * (y - H * xhatH);
    PH = F * PH * temp * F' + Q;
    lambda = eig(inv(PH) - theta * Sbar + H' * inv(R) * H);
    if min(real(lambda)) <= 0
        disp('H-infinity performance bound out of range');
        return;
    end
    % Simulate the system.
    x = F*x + B*uTrue + Qsqrt*randn(size(x));
    % Get the measurement.
    y = H * x + Rsqrt * randn(size(Rsqrt,1),1);
    % Save data in arrays.
    xhatKarray = [xhatKarray xhatK];
    xhatHarray = [xhatHarray xhatH];
    xarray = [xarray x];
end

% Compute averages.
EstErrorK = xarray - xhatKarray;
```

```
EstErrorK = sqrt(EstErrorK(1,:).^2 + EstErrorK(2,:).^2);
EstError = mean(EstErrorK);
disp(['Kalman Average Position Estimation Error = ', num2str(EstError)]);

EstErrorH = xarray - xhatHarray;
EstErrorH = sqrt(EstErrorH(1,:).^2 + EstErrorH(2,:).^2);
EstError = mean(EstErrorH);
disp(['H-infinity Average Position Estimation Error = ', num2str
(EstError)]);

% Plot data.
t = 0 : T : tf;
subplot(2,1,FigNum);
plot(t, EstErrorK, 'r:', 'LineWidth', 2.5); hold on;
plot(t, EstErrorH, 'b-', 'LineWidth', 1.5);
set(gca,'FontSize',12); set(gcf,'Color','White');
title(['True Input = ', num2str(uTrue)]);
if FigNum == 2
    xlabel('seconds');
end
ylabel('pos. est. errors');
legend('Kalman', 'H_{\infty}');

EstErrorH = xarray - xhatHarray;
stdH = std(EstErrorH, 0, 2);
```

```
function HinfVehicle1(theta)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 11.15

% Calculate closed loop Kalman and H-infinity filter eigenvalues for
% the vehicle navigation problem.

if ~exist('theta', 'var')
    theta = 0.0005;
end

T = 3; % time step (seconds)
Q = diag([4, 4, 1, 1]); % Process noise covariance (m, m, m/sec, m/sec)
R = diag([900, 900]); % Measurement noise covariance (m, m)
S = eye(4);
L = eye(4);
Sbar = L' * S * L;
F = [1 0 T 0 ; 0 1 0 T ; 0 0 1 0 ; 0 0 0 1]; % system matrix
H = [1 0 0 0 ; 0 1 0 0]; % measurement matrix
PK = dare(F', H', Q, R);

% Calculate closed loop Kalman filter eigenvalues
KK = PK * inv(eye(4) + H' * inv(R) * H * PK) * H' * inv(R);
abs(eig(F*(eye(4)-KK*H)))

PH = dare(F', eye(4), Q, (H' * R^(-1) * H - theta * Sbar)^(-1), zeros
(4,4), eye(4))
% Calculate closed loop H-infinity filter eigenvalues
temp = eye(4) - theta * Sbar * PH + H' * inv(R) * H * PH;
if cond(temp) > 1e10
    disp('ill conditioned H-infinity estimation problem');
    return;
end
KH = PH * inv(temp) * H' * inv(R);
abs(eig(F*(eye(4)-KH*H)))
```

```
function HybridNewton(theta)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 12.8

% Kalman, H-infinity, and hybrid filter computation for a
% two-state Newtonian system.

T = 1; % step size
a = 1; % acceleration noise
R = 1; % measurement noise

F = [1 T; 0 1];
H = [1 0];
Q = a^2 * [T^4/2 T^3/2; T^3/2 T^2];

P = dare(F', H', Q, R);
KK = P * H' * inv(H * P * H' + R);

P = dare(F', eye(2), Q, inv(H'*inv(R)*H+theta*eye(2)));
lambda = min(real(eig(inv(P) - theta*eye(2) + H'*inv(R)*H)));
if lambda <= 0
    disp(['theta is too big - min(lambda) = ', num2str(lambda)]);
    return;
end
KH = P * inv(eye(2) - theta * P + H' * inv(R) * H * P) * H' * inv(R);

lambda = [];
for d = 0 : 0.01 : 1
    K = d * KK + (1 - d) * KH;
    lambda = [lambda max(abs(eig(F * (eye(2) - K * H))))];
end

close all;
figure;
set(gca,'FontSize',12); set(gcf,'Color','White');
plot(0 : 0.01 : 1, lambda, 'LineWidth', 2);
xlabel('d');
ylabel('max magnitude of estimator')
```

```
function HungScalar(F, M1, alpha, N, eps, theta)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 12.9

% Robust mixed Kalman/H-infinity filter for a scalar system.

Ptilde = roots([N^2, -alpha+alpha*F^2-N^2-alpha*F^2*N^2-eps*N^2,
alpha+alpha^2*M1^2+alpha*eps]);
i = 1;
while i <= length(Ptilde)
    if ~isreal(Ptilde(i)) | (Ptilde(i) <= 0) | (alpha - N * Ptilde(i) * N
<= 0)
        Ptilde = [Ptilde(1:i-1) Ptilde(i+1:end)];
        i = i - 1;
    end
    i = i + 1;
end
if isempty(Ptilde)
    disp('Ptilde condition failed');
    return;
end
Ptilde = Ptilde(1);

R11 = 1 + alpha * M1^2;
Y = R11 + R11^2 * (alpha - Ptilde * N^2) / alpha / Ptilde / F^2 + eps;
a = theta^2 * (theta^2 - 1);
b = 1 - 2 * theta^2 + F^2 * theta^2 + theta^2 * (1 - theta^2) * Y;
c = 1 - F^2 + (2 * theta^2 - 1) * Y;
d = -Y;

P = roots([a b c d]);
i = 1;
while i <= length(P)
    if ~isreal(P(i)) | (P(i) <= 0) | (1 / theta^2 - P(i) <= 0)
        P = [P(1:i-1) P(i+1:end)];
        i = i - 1;
    end
    i = i + 1;
end
if isempty(P)
    disp('P condition failed');
    return;
end
P = P(1)

% Kalman gain calculation
KK = (sqrt(5) - 1) / 2;

% Robust filter gain calculation
```

```
T = 1 / (1 / P - theta^2);
Rtilde = T + 1;
R1 = F / (1 / Ptilde - N^2 / alpha);
F1 = F + R11 / R1;
KH = F1 * T / Rtilde;
Fhat = F1 - KH;
if (abs(Fhat) >= 1)
    disp('unstable estimator');
    return;
end

x = 0;
xhatK = 0;
xhatH = xhatK;
ErrK = [];
ErrH = ErrK;
kf = 1000;
for i = 1 : kf
    % System measurement
    y = x + randn;
    % Kalman filter
    xhatK = F * (1 - KK) * xhatK + F * KK * y;
    % Robust filter
    xhatH = Fhat * xhatH + KH * y;
    % System simulation
    x = (F + M1 * N) * x + randn;
    % Save data for plotting
    ErrK = [ErrK x-xhatK];
    ErrH = [ErrH x-xhatH];
end

disp(['std dev of est error (Kalman, robust) = ', num2str(std(ErrK)), ',
', num2str(std(ErrH))]);
```

```
function HinfConstr1(H1, G1)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 12.10

% Constrained H-infinity state estimation for a two-state system.

if ~exist('H1', 'var')
    H1 = 1;
end
if ~exist('G1', 'var')
    G1 = 0.1;
end

F = [1 1; 0 1];
%F = [1 1/2; 1/2 0];

H = [H1 0];
G = [G1 0];

D = [1 1];
Q = [1 0; 0 1];

P = [2 0; 0 1];
for k = 1 : 2000
    Sigma = inv(P * H' * H - P * G' * G + eye(2)) * P;
    temp = (eye(2) - D' * D) * F;
    P = temp * Sigma * temp' + Q;
    if k == 1000
        P
    end
end
P
eig(P)
1-G*P*G'
```

```
function SquareRoot

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.15

% Moving average filter and EKF for nonlinear scalar system.

kf = 100;
R = 1;
x = 1;
xhat1 = 2; % initial simple filter estimate
xhat2 = xhat1; % initial Kalman filter estimate
P = 1; % initial Kalman filter variance

% initialize arrays
xhat1Arr = [];
xhat2Arr = [];

for k = 1 : kf
    % Measurement
    y = sqrt(x) * (1 + randn(R));
    % Simple filter
    xhat1 = ((k - 1) * xhat1 + y^2) / k;
    % Kalman filter
    yhat = sqrt(xhat2);
    H = 1 / 2 / yhat;
    M = yhat;
    K = P * H' * inv(H * P * H' + M * R * M');
    xhat2 = xhat2 + K * (y - yhat);
    P = (1 - K * H) * P;
    % Save estimates in arrays
    xhat1Arr = [xhat1Arr xhat1];
    xhat2Arr = [xhat2Arr xhat2];
end

close all;
figure;
k = 1 : kf;
set(gca,'FontSize',12); set(gcf,'Color','White');
plot(k, xhat1Arr, 'b-', k, xhat2Arr, 'r:');
xlabel('time step'); ylabel('estimate');
legend('Simple filter', 'Kalman filter');
```

```
function Orbit

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.16

% Linearized and extended Kalman filters for orbiting satellite
% state estimation.

dt = 60; % simulation step size
tf = 180 * 60; % simulation length

G = 6.6742e-11; % gravitational constant (m^3/kg/s^2)
M = 5.98e24; % earth mass (kg)
Radius = 6.37e6; % earth radius (m)

% Note if Q = 0, PL remains 0 and the KF does not respond to measurements
Qc = 1e-6; % continuous time process noise variance for radius accel.
Q = diag([0 Qc*dt 0 0]); % discretized process noise covariance
R = diag([10000 0.01]); % measurement noise covariance
H = [1 0 0 0 ; 0 0 1 0]; % measurement matrix

x = [Radius+2e5 ; 0 ; 0 ; 0]; % initial state
x(4) = sqrt(M * G / x(1)^3);
x(4) = 1.1 * x(4);
x0 = x; % nominal state
dxhatL = zeros(size(x)); % initial estimate for linearized Kalman filter
xhatL = x0 + dxhatL;
PL = diag([0 0 0 0]); % initial est. error covariance for linearized KF
xhatE = x; % initial estimate for extended Kalman filter
PE = PL; % initial estimation error covariance for extended KF

% initialize arrays
xArr = x;
xhatLArr = xhatL;
xhatEArr = xhatE;

for t = dt : dt : tf+dt/10
    % system simulation
    xdot(1,1) = x(2);
    xdot(2,1) = x(1) * x(4)^2 - M * G / x(1) / x(1);
    xdot(3,1) = x(4);
    xdot(4,1) = -2 * x(4) * x(2) / x(1);
    xdot = xdot + sqrt(Q) * randn(4,1);
    x = x + xdot * dt;
    x(3) = mod(x(3), 2*pi);
    y = H * x + sqrt(R) * randn(2,1);
    % linearized Kalman filter
    FL = [0 1 0 0 ; 3*x0(4)^2 0 0 2*x0(1)*x0(4) ; 0 0 0 1 ; 0 -2*x0(4)/x0
(1) 0 0];
    dxhatLdot = FL * dxhatL;
```

```
    dxhatL = dxhatL + dxhatLdot * dt;
    PLdot = FL * PL + PL * FL' + Q;
    PL = PL + PLdot * dt;
    KL = PL * H' * inv(H * PL * H' + R);
    dxhatL = dxhatL + KL * (y - H * (x0 + dxhatL));
    x0(3) = x0(4) * t;
    xhatL = x0 + dxhatL;
    xhatL(3) = mod(xhatL(3), 2*pi);
    PL = PL - KL * H * PL;
    % extended Kalman filter
    FE = [0 1 0 0 ; xhatE(4)^2+2*G*M/xhatE(1)^3 0 0 2*xhatE(1)*xhatE(4) ;
0 0 0 1 ;
        2*xhatE(4)*xhatE(2)/xhatE(1)^2 -2*xhatE(4)/xhatE(1) 0 -2*xhatE(2)
/xhatE(1)];
    xhatEdot(1) = xhatE(2);
    xhatEdot(2) = xhatE(1) * xhatE(4)^2 - M * G / xhatE(1) / xhatE(1);
    xhatEdot(3) = xhatE(4);
    xhatEdot(4) = -2 * xhatE(4) * xhatE(2) / xhatE(1);
    xhatE = xhatE + xhatEdot' * dt;
    PEdot = FE * PE + PE * FE' + Q;
    PE = PE + PEdot * dt;
    KE = PE * H' * inv(H * PE * H' + R);
    xhatE = xhatE + KE * (y - H * xhatE);
    xhatE(3) = mod(xhatE(3), 2*pi);
    PE = PE - KE * H * PE;
    % save data in arrays
    xArr = [xArr x];
    xhatLArr = [xhatLArr xhatL];
    xhatEArr = [xhatEArr xhatE];
end

close all;
t = 0 : dt : tf;
t = t / 60;

% Plot true states
figure;
set(gcf,'Color','White');
subplot(2,2,1); plot(t, xArr(1,:), 'k-', t, xhatLArr(1,:), 'r:');
set(gca,'FontSize',12); ylabel('range (meters)');
subplot(2,2,2); plot(t, xArr(2,:), 'k-', t, xhatLArr(2,:), 'r:');
set(gca,'FontSize',12); ylabel('range rate (meters/s)');
subplot(2,2,3); plot(t, xArr(3,:), 'k-', t, xhatLArr(3,:), 'r:');
set(gca,'FontSize',12); xlabel('minutes'); ylabel('angle (rad)');
subplot(2,2,4); plot(t, xArr(4,:), 'k-', t, xhatLArr(4,:), 'r:');
set(gca,'FontSize',12); xlabel('minutes'); ylabel('angle rate (rad/s)');

% Plot EKF radius estimation error
figure;
plot(t, xArr(1,:)-xhatEArr(1,:), 'b-');
```

```
set(gcf,'Color','White'); set(gca,'FontSize',12);
xlabel('minutes'); ylabel('meters');

% Plot linearized Kalman filter radius estimation error
figure;
plot(t, xArr(1,:)-xhatLArr(1,:), 'b-');
set(gcf,'Color','White'); set(gca,'FontSize',12);
xlabel('minutes'); ylabel('meters');
```

```
function Motor(ControlNoise)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.17

% Hybrid extended Kalman filter simulation for two-phase step motor.
% Estimate the stator currents, and the rotor position and velocity, on
% the basis of noisy measurements of the stator currents.

% Initialize the random number generator
randn('state', 1);

T = 0.1; % measurement period

Ra = 1.9; % Winding resistance
L = 0.003; % Winding inductance
lambda = 0.1; % Motor constant
J = 0.00018; % Moment of inertia
B = 0.001; % Coefficient of viscous friction

if ~exist('ControlNoise', 'var')
    ControlNoise = 0.01; % std dev of uncertainty in control inputs
end

AccelNoise = 0.5; % std dev of shaft acceleration noise
MeasNoise = 0.1; % standard deviation of measurement noise

R = [MeasNoise^2 0; 0 MeasNoise^2]; % Measurement noise covariance
xdotNoise = [ControlNoise/L ; ControlNoise/L ; 0.5 ; 0];
Q = xdotNoise * xdotNoise'; % Process noise covariance
P = 1*eye(4); % Initial state estimation covariance

dt = 0.0005; % Integration step size
tf = 3; % Simulation length

x = [0; 0; 0; 0]; % Initial state
xhat = x; % State estimate
w = 2 * pi; % Control input frequency

% Initialize arrays for plotting at the end of the program
xArray = [];
xhatArray = [];
trPArray = [];
tArray = [];

% Begin simulation loop
for t = 0 : T : tf
    xArray = [xArray x];
    xhatArray = [xhatArray xhat];
    trPArray = [trPArray trace(P)];
```

```
    tArray = [tArray t];
    % Nonlinear simulation
    for tau = dt : dt : T+dt/10
        ua0 = sin(w*t);
        ub0 = cos(w*t);
        s4 = sin(x(4));
        c4 = cos(x(4));
        xdot = [-Ra/L*x(1) + x(3)*lambda/L*s4 + ua0/L;
            -Ra/L*x(2) - x(3)*lambda/L*c4 + ub0/L;
            -3/2*lambda/J*x(1)*s4 + 3/2*lambda/J*x(2)*c4 - B/J*x(3);
            x(3)];
        xdot = xdot + xdotNoise .* randn(4,1);
        x = x + xdot * dt;
        x(4) = mod(x(4), 2*pi);
    end
    H = [1 0 0 0; 0 1 0 0];
    z = H * x + [MeasNoise*randn; MeasNoise*randn];
    % Kalman filter time update equations
    for tau = dt : dt : T+dt/10
        s4 = sin(xhat(4));
        c4 = cos(xhat(4));
        F = [-Ra/L 0 lambda/L*s4 xhat(3)*lambda/L*c4;
            0 -Ra/L -lambda/L*c4 xhat(3)*lambda/L*s4;
            -3/2*lambda/J*s4 3/2*lambda/J*c4 -B/J -3/2*lambda/J*(xhat(1)↙
*c4+xhat(2)*s4);
            0 0 1 0];
        xhatdot = [-Ra/L*xhat(1) + xhat(3)*lambda/L*sin(xhat(4)) + ua0/L;
            -Ra/L*xhat(2) - xhat(3)*lambda/L*cos(xhat(4)) + ub0/L;
            -3/2*lambda/J*xhat(1)*sin(xhat(4)) + 3/2*lambda/J*xhat(2)*cos↙
(xhat(4)) - B/J*xhat(3);
            xhat(3)];
        xhat = xhat + xhatdot * dt;
        Pdot = F * P + P * F' + Q;
        P = P + Pdot * dt;
    end
    % Kalman filter measurement update equations
    K = P * H' * inv(H * P * H' + R);
    xhat = xhat + K * (z - H * xhat);
    xhat(4) = mod(xhat(4), 2*pi);
    P = (eye(4) - K * H) * P * (eye(4) - K * H)' + K * R * K';
end

% Plot data.
close all;
figure;
plot(tArray, xArray(1,:), tArray,xhatArray(1,:),'r:')
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('Time (Seconds)'); ylabel('Winding A Current (Amps)');
legend('True', 'Estimated');
```

```
figure;
plot(tArray, xArray(2,:), tArray,xhatArray(2,:),'r:')
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('Time (Seconds)'); ylabel('Winding B Current (Amps)');
legend('True', 'Estimated');

figure;
plot(tArray, xArray(3,:), tArray,xhatArray(3,:),'r:')
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('Time (Seconds)'); ylabel('Rotor Speed (Radians / Sec)');
legend('True', 'Estimated');

figure;
plot(tArray, xArray(4,:), tArray,xhatArray(4,:),'r:')
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('Time (Seconds)'); ylabel('Rotor Position (Radians)');
legend('True', 'Estimated');

% Compute the std dev of the estimation errors
N = size(xArray, 2);
N2 = round(N / 2);
xArray = xArray(:,N2:N);
xhatArray = xhatArray(:,N2:N);
iaEstErr = sqrt(norm(xArray(1,:)-xhatArray(1,:))^2 / size(xArray,2));
ibEstErr = sqrt(norm(xArray(2,:)-xhatArray(2,:))^2 / size(xArray,2));
wEstErr = sqrt(norm(xArray(3,:)-xhatArray(3,:))^2 / size(xArray,2));
thetaEstErr = sqrt(norm(xArray(4,:)-xhatArray(4,:))^2 / size(xArray,2));
disp(['Std Dev of Estimation Errors = ',num2str(iaEstErr),', ',num2str
(ibEstErr),', ',num2str(wEstErr),', ',num2str(thetaEstErr)]);

% Display the P version of the estimation error standard deviations
disp(['Sqrt(P) = ',num2str(sqrt(P(1,1))),', ',num2str(sqrt(P(2,2))),', ',
num2str(sqrt(P(3,3))),', ',num2str(sqrt(P(4,4)))]);
```

```
function [Est1Err, Est2Err, EstiErr] = Hen1(PlotFlag)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.18

% Discrete time 1st order and 2nd order EKF, and iterated EKF.
% 2nd order EKF algorithm is based on Henriksen's 1982 paper.

% INPUTS:
%   PlotFlag = flag saying whether or not to plot results
% OUTPUTS:
%   Est1Err = 1st order EKF estimation error (RMS)
%   Est2Err = 2nd order EKF estimation error (RMS)
%   EstiErr = iterated EKF estimation error (RMS)

if ~exist('PlotFlag', 'var')
    PlotFlag = true;
end

Q = 0.1; % process noise covariance
R = 1; % measurement noise variance

x = 1; % true state

P = 1; % initial estimation error variance
P2 = P;
Pi = P;

kf = 5; % simulation length

N = 2; % number of iterations in iterated EKF

xhat = 2; % first order EKF estimate
xhat2 = xhat; % second order EKF estimate
xhati = xhat; % iterated EKF estimate

randn('state', sum(100*clock)); % random number generator seed

xArray = x;
xhatArray = xhat;
xhat2Array = xhat2;
xhatiArray = xhat2;
Parray = P;
P2array = P2;
Piarray = Pi;

for k = 1 : kf

    % Simulate the system.
    x = x^2 + sqrt(Q) * randn;
```

```
    % Simulate the noisy measurement.
    y = x^2 + sqrt(R) * randn;

    % First order Kalman filter.
    F = 2 * xhat;
    xhat = xhat^2;
    P = F * P * F' + Q;
    H = 2 * xhat;
    K = P * H' * inv(H * P * H' + R);
    yhat = xhat^2;
    xhat = xhat + K * (y - yhat);
    P = (1 - K * H) * P;

    % Second order Kalman filter.
    % Continuous-time part of the 2nd order Kalman filter (time update).
    F2 = 2 * xhat2;
    xhat2 = xhat2^2 + (1/2) * 2 * P2;
    P2 = F2 * P2 * F2' + Q;
    H2 = 2 * xhat2;
    K2 = P2 * H2' * inv(H2 * P2 * H2' + R);
    yhat2 = xhat2^2 + (1/2) * 2 * P2;
    xhat2 = xhat2 + K2 * (y - yhat2);
    P2 = (1 - K2 * H2) * P2;

    % Iterated Kalman filter.
    % Continuous-time part of the iterated Kalman filter (time update).
    Fi = 2 * xhati;
    xhati = xhati^2;
    Pi = Fi * Pi * Fi' + Q;
    xhatminus = xhati;
    Pminus = Pi;
    for i = 1 : N
        Hi = 2 * xhati;
        Ki = Pminus * Hi' * inv(Hi * Pminus * Hi' + R);
        yhati = xhati^2;
        xhati = xhati + Ki * (y - yhati - Hi * (xhatminus - xhati));
        Pi = (1 - Ki * Hi) * Pminus;
    end

    % Save data for plotting.
    xArray = [xArray x];
    xhatArray = [xhatArray xhat];
    xhat2Array = [xhat2Array xhat2];
    xhatiArray = [xhatiArray xhati];
    Parray = [Parray diag(P)];
    P2array = [P2array diag(P2)];
    Piarray = [Piarray diag(Pi)];
end

Est1Err = sqrt((norm(xArray - xhatArray))^2/length(xArray));
```

```
Est2Err = sqrt((norm(xArray - xhat2Array))^2/length(xArray));
EstiErr = sqrt((norm(xArray - xhatiArray))^2/length(xArray));

if ~PlotFlag
    return;
end

close all;
k = 0 : kf;

figure; hold;
plot(k, xArray, 'b', k, xhatArray, 'r:', k, xhat2Array, 'k--', k,
xhatiArray, 'm-.');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('Seconds');
legend('True state', 'First order estimate', 'Second order estimate',
'Iterated estimate');

disp(['1st Order EKF RMS estimation error = ', num2str(Est1Err)]);
disp(['2nd Order EKF RMS estimation error = ', num2str(Est2Err)]);
disp(['Itereated EKF RMS estimation error = ', num2str(EstiErr)]);
```

```
function pdfApprox

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.19

% Approximate a pdf that is uniform on (-1,+1) with a sum of Gaussians.
% The Gaussian pdfs have means that are equally spaced on (-1,+1), and
% they have identical variances. Find the variance the minimizes the
% RMS error between the uniform pdf and the sum of Gaussian pdfs.

global muGauss NumPts dx

dx = 0.01;
NumPts = 2 / dx + 1;
MArray = [3 5 10]; % number of Gaussians used to estimation pdf
close all;
figure;
for k = 1 : length(MArray)
    M = MArray(k);
    muGauss(1) = -1 + 2 / (M + 1);
    for i = 2 : M
        muGauss(i) = muGauss(i-1) + 2 / (M + 1);
    end

    % Golden search code - copied from
    % http://www.cse.ucsc.edu/~hongwang/Codes/Golden_search
    a=0.01;
    b=2;
    tol=1.0e-10;
    n=0;
    g=(sqrt(5)-1)/2;
    r1=a+(b-a)*(1-g);
    f1=f(r1, M);
    r2=a+(b-a)*g;
    f2=f(r2, M);
    while (b-a) > tol,
        n=n+1;
        if f1 < f2,
            b=r2;
            r2=r1;
            f2=f1;
            r1=a+(b-a)*(1-g);
            f1=f(r1, M);
        else
            a=r1;
            r1=r2;
            f1=f2;
            r2=a+(b-a)*g;
            f2=f(r2, M);
        end
```

```
    end
    x0=(a+b)/2;
    % Plot the final solution and compute the RMS error.
    subplot(3,1,k);
    ErrRMS = f(x0, M, true)
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ErrRMS = f(sigma, M, PlotFlag)
global muGauss NumPts dx
ErrRMS = 0;
j = 1;
for x = -1 : dx : 1
    Approx(j) = 0;
    for i = 1 : M
        Approx(j) = Approx(j) + (1/M) * exp(-(x-muGauss(i))^2 / 2 /
sigma^2);
    end
    ErrRMS = ErrRMS + (Approx(j) - 1/2)^2;
    j = j + 1;
end
ErrRMS = sqrt(ErrRMS / NumPts);
if ~exist('PlotFlag', 'var')
    PlotFlag = false;
end
if PlotFlag
    set(gca,'FontSize',12); set(gcf,'Color','White');
    x = -1 : dx : 1;
    plot(x, (1/2)*ones(size(x)), 'k-', 'LineWidth', 2);
    hold on;
    plot(x, Approx, 'r:', 'LineWidth', 2);
    text(-0.5, 0.8, ['M = ', num2str(M), ' : sigma = ', num2str(sigma), ',
RMS Error = ', num2str(ErrRMS)]);
    v = axis;
    axis([v(1) v(2) 0 v(4)]);
end
end
```

```
function GaussFilter

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.20

% EKF and Gaussian sum filter for scalar system.

kf = 20; % simulation length

Q = 0;
R = 0.01;
M = 2; % number of terms in Gaussian sum filter

x = -1/2; % initial state
xhat = 0; % initial estimate

for i = 1 : M
    a(i) = 1 / M;
    xhat(i) = -1/3 + (2/3) * (i-1); % initial Gaussian sum filter est.
    Pplus(i) = 0.43^2;
end

% Initialize arrays
xArr = x;
xhatArr = xhat';

for k = 1 : kf
    % System simulation
    x = x + sqrt(Q) * randn;
    y = x^2 + sqrt(R) * randn;
    % Kalman filter simulation
    for i = 1 : M
        xhat(i) = xhat(i);
        F(i) = 1;
        Pminus(i) = F(i) * Pplus(i) * F(i)' + Q;
        H(i) = 2 * xhat(i);
        S(i) = H(i) * Pminus(i) * H(i)' + R;
        K(i) = Pminus(i) * H(i)' * inv(S(i));
        Pplus(i) = (1 - K(i) * H(i)) * Pminus(i);
        r(i) = y - xhat(i)^2;
        xhat(i) = xhat(i) + K(i) * (y - xhat(i)^2);
        beta(i) = exp(-r(i)' * inv(S(i)) * r(i) / 2) / sqrt(2 * pi)^length
(x) / sqrt(det(S(i)));
    end
    a = a .* beta / sum(a.*beta);
    % Save data for plotting
    xArr = [xArr x];
    xhatArr = [xhatArr xhat'];
end
```

```
% Plot the results
close all;
k = 0 : kf;

figure; set(gca,'FontSize',12); set(gcf,'Color','White');
subplot(2,1,1);
plot(k, xArr, 'r:', k, xhatArr(1,:), 'b-', k, xhatArr(2,:), 'k.-');
xlabel('time'); ylabel('state');
legend('true state', 'estimated state 1', 'estimated state 2');

set(gca,'FontSize',12); set(gcf,'Color','White');
for i = 1 : 2
    xmin = xhat(i) - 3 * sqrt(Pplus(i));
    xmax = xhat(i) + 3 * sqrt(Pplus(i));
    dx = (xmax - xmin) / 100;
    j = 1;
    for x = xmin : dx : xmax
        pdf(j) = exp(-(x - xhat(i))^2 / 2 / Pplus(i));
        j = j + 1;
    end
    x = xmin : dx : xmax;
    subplot(2,1,2); hold on; ylabel('pdfs');
    plot(x, pdf);
end
axis([-0.7 0.7 0 1]);
```

```
function TrackEKF

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.21

% EKF-based ground vehicle navigation

% Define the north and east coordinates of the two tracking stations.
N1 = 20;
E1 = 0;
N2 = 0;
E2 = 20;

dt = 0.1; % simulation step size
tf = 60; % simulation length

F = [1 0 dt 0 ; 0 1 0 dt ; 0 0 1 0 ; 0 0 0 1];
Q = diag([0 0 4 4]);
R = diag([1 1]);

x = [0; 0; 50; 50]; % initial state
xhatplus = x; % initial state estimate
Pplus = diag([0 0 0 0]); % initial estimation error covariance

% Initialize arrays
xArr = x;
xhatArr = xhatplus;

for t = dt : dt : tf
    % System simulation
    x = F * x + sqrt(Q) * randn(4,1);
    y(1,1) = sqrt((x(1) - N1)^2 + (x(2) - E1)^2);
    y(2,1) = sqrt((x(1) - N2)^2 + (x(2) - E2)^2);
    % EKF time update
    Pminus = F * Pplus * F' + Q;
    xhatminus = F * xhatplus;
    % EKF measurement update
    H = zeros(2,4);
    nhat = xhatminus(1);
    ehat = xhatminus(2);
    temp = sqrt((nhat - N1)^2 + (ehat - E1)^2);
    H(1,1) = (nhat - N1) / temp;
    H(1,2) = (ehat - E1) / temp;
    temp = sqrt((nhat - N2)^2 + (ehat - E2)^2);
    H(2,1) = (nhat - N2) / temp;
    H(2,2) = (ehat - E2) / temp;
    K = Pminus * H' * inv(H * Pminus * H' + R);
    yhat(1,1) = sqrt((nhat - N1)^2 + (ehat - E1)^2);
    yhat(2,1) = sqrt((nhat - N2)^2 + (ehat - E2)^2);
    xhatplus = xhatminus + K * (y - yhat);
```

```
    Pplus = (eye(4) - K * H) * Pminus;
    % Save data for plotting
    xArr = [xArr x];
    xhatArr = [xhatArr xhatplus];
end

% Plot results
close all;
k = 0 : dt : tf;

figure; set(gca,'FontSize',12); set(gcf,'Color','White');
subplot(2,2,1);
plot(k, xArr(1,:)-xhatArr(1,:), 'b-');
title('North Position Estimation Error');

subplot(2,2,2);
plot(k, xArr(2,:)-xhatArr(2,:), 'b-');
title('East Position Estimation Error');

subplot(2,2,3);
plot(k, xArr(3,:)-xhatArr(3,:), 'b-'); xlabel('time');
title('North Velocity Estimation Error');

subplot(2,2,4);
plot(k, xArr(4,:)-xhatArr(4,:), 'b-'); xlabel('time');
title('East Velocity Estimation Error');

% Compute experimental standard deviations of estimation errors.
EstStd = std(xArr' - xhatArr');
disp(['Experimental std dev of est err = ', num2str(EstStd)]);
```

```
function SysId

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.22

% EKF for system identification.

phi = 0.9; % true value of phi
Q = [1 0; 0 0]; % variance of process noise

kf = 100; % number of time steps
x = 1; % initial value of x
P = eye(2); % initial Kalman filter covariance
xhat = [0 0]'; % initial Kalman filter state estimate
xhatArr = []; % array for saving state estimate

for k = 1 : kf
    % system equations
    x = phi * x + sqrt(Q(1,1)) * randn;
    y = x;
    % EKF time update
    F = [xhat(2) xhat(1) ; 0 1];
    P = F * P * F' + Q;
    xhat(1) = xhat(2) * xhat(1);
    xhat(2) = xhat(2);
    % EKF measurement update
    H = [1 0];
    K = P * H' * inv(H * P * H');
    xhat = xhat + K * (y - xhat(1));
    P = (eye(2) - K * H) * P;
    % Save data for arrays
    xhatArr = [xhatArr xhat];
end

close all;
k = 1 : kf;
figure;
set(gca,'FontSize',12); set(gcf,'Color','White');
plot(k, phi*ones(size(k)), 'k-', k, xhatArr(2,:), 'r:');
legend('true', 'estimated');
xlabel('time step'); ylabel('phi');
```

```
function Parameter

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 13.23

% Extended Kalman filter for parameter estimation.
% Estimate the natural frequency of a second order system.

tf = 100; % simulation length
dt = 0.01; % simulation step size
wn = 2; % natural frequency
zeta = 0.1; % damping ratio
b = -2 * zeta * wn;
Q2 = 1;

Q = [1000 0; 0 Q2]; % covariance of process noise
R = [10 0; 0 10]; % covariance of measurement noise
H = [1 0 0; 0 1 0]; % measurement matrix
P = [0 0 0; 0 0 0; 0 0 20]; % covariance of estimation error

close all;
figure;
set(gca,'FontSize',12); set(gcf,'Color','White');

sigma2 = [0 1 100];
for i = 1 : length(sigma2)
    Q2 = sigma2(i); % artificial noise used for parameter estimation
    x = [0; 0; -wn*wn]; % initial state
    xhat = 2 * x; % initial state estimate

    % Initialize arrays for later plotting
    xArray = x;
    xhatArray = xhat;
    P3Array = P(3,3);

    dtPlot = tf / 100; % how often to plot output data
    tPlot = 0;

    for t = dt : dt : tf+dt
        % Simulate the system.
        w = sqrt(Q(1,1)) * randn;
        xdot = [x(2); x(3)*x(1) + b*x(2) - x(3)*w; 0];
        x = x + xdot * dt;
        z = H * x + sqrt(R) * [randn; randn];
        % Simulate the Kalman filter.
        F = [0 1 0; xhat(3) b xhat(1); 0 0 0];
        L = [0 0; -xhat(3) 0; 0 1];
        Pdot = F * P + P * F' + L * Q * L' - P * H' * inv(R) * H * P;
        P = P + Pdot * dt;
        K = P * H' * inv(R);
```

```
        xhatdot = [xhat(2); xhat(3)*xhat(1) + b*xhat(2); 0];
        xhatdot = xhatdot + K * (z - H * xhat);
        xhat = xhat + xhatdot * dt;
        if (t >= tPlot + dtPlot - 100*eps)
            % Save data for plotting.
            xArray = [xArray x];
            xhatArray = [xhatArray xhat];
            P3Array = [P3Array P(3,3)];
            tPlot = t;
        end
    end
    % Plot results
    t = 0 : dtPlot : tf;
    subplot(3,1,i);
    plot(t, xArray(3,:) - xhatArray(3,:), 'LineWidth', 2);
    set(gca,'FontSize',12); set(gcf,'Color','White');
    if (i == 3)
        xlabel('Seconds');
    elseif (i == 2)
        ylabel('w_n^2 Estimation Error');
    end
    axis([0 tf -1 5]);
    legend(['\sigma_p^2 = ', num2str(sigma2(i))]);
end
```

```
function TrackUKF(tf);

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 14.14

% UKF simulation of land vehicle tracking system.
% INPUTS:
%    tf = simulation length

if ~exist('tf', 'var')
    tf = 60;
end
dt = 0.1; % simulation step size

% Define the north and east coordinates of the two tracking stations.
N1 = 20;
E1 = 0;
N2 = 0;
E2 = 20;

F = [1 0 dt 0 ; 0 1 0 dt ; 0 0 1 0 ; 0 0 0 1];
Q = diag([0 0 4 4]);
R = diag([1 1]);

x = [0; 0; 50; 50];
xhatplus = x;
Pplus = 0.01 * eye(4); % must be positive definite for UKF

% Initialize arrays
xArr = x;
xhatArr = xhatplus;
PArr(:,:,1) = Pplus;

N = length(x);
k = 1;
for t = dt : dt : tf
    % System simulation
    x = F * x + sqrt(Q) * randn(4,1);
    y(1,1) = sqrt((x(1) - N1)^2 + (x(2) - E1)^2);
    y(2,1) = sqrt((x(1) - N2)^2 + (x(2) - E2)^2);
    y = y + sqrt(R) * randn(2,1);
    % UKF time update
    [root,p] = chol(4*Pplus);
    for i = 1 : N
        sigma(:,i) = xhatplus + root(i,:)';
        sigma(:,i+N) = xhatplus - root(i,:)';
    end
    for i = 1 : 2*N
        sigma(:,i) = F * sigma(:,i);
    end
```

```
    xhatminus = zeros(N,1);
    for i = 1 : 2*N
        xhatminus = xhatminus + sigma(:,i) / 2 / N;
    end
    Pminus = zeros(N,N);
    for i = 1 : 2*N
        Pminus = Pminus + (sigma(:,i) - xhatminus) * (sigma(:,i) -
xhatminus)' / 2 / N;
    end
    Pminus = Pminus + Q;
    % UKF measurement update
    [root,p] = chol(N*Pminus);
    for i = 1 : N
        sigma(:,i) = xhatminus + root(i,:)';
        sigma(:,i+N) = xhatminus - root(i,:)';
    end
    for i = 1 : 2*N
        nhat = sigma(1,i);
        ehat = sigma(2,i);
        yukf(1,i) = sqrt((nhat - N1)^2 + (ehat - E1)^2);
        yukf(2,i) = sqrt((nhat - N2)^2 + (ehat - E2)^2);
    end
    yhat = 0;
    for i = 1 : 2*N
        yhat = yhat + yukf(:,i) / 2 / N;
    end
    Py = zeros(2,2);
    Pxy = zeros(N,2);
    for i = 1 : 2*N
        Py = Py + (yukf(:,i) - yhat) * (yukf(:,i) - yhat)' / 2 / N;
        Pxy = Pxy + (sigma(:,i) - xhatminus) * (yukf(:,i) - yhat)' / 2 /
N;
    end
    Py = Py + R;
    Kukf = Pxy * inv(Py);
    xhatplus = xhatminus + Kukf * (y - yhat);
    Pplus = Pminus - Kukf * Py * Kukf';
    % Save data for plotting
    xArr = [xArr x];
    xhatArr = [xhatArr xhatplus];
    k = k + 1;
    PArr(:,:,k) = Pplus;
end

% Plot results
close all;
k = 0 : dt : tf;

% Plot estimation errors
figure; set(gcf,'Color','White');
```

```
subplot(2,2,1); set(gca,'FontSize',12);
plot(k, xArr(1,:)-xhatArr(1,:), 'b-', 'LineWidth', 2);
title('North Position Estimation Error');

subplot(2,2,2); set(gca,'FontSize',12);
plot(k, xArr(2,:)-xhatArr(2,:), 'b-', 'LineWidth', 2);
title('East Position Estimation Error');

subplot(2,2,3); set(gca,'FontSize',12);
plot(k, xArr(3,:)-xhatArr(3,:), 'b-', 'LineWidth', 2); xlabel('time');
title('North Velocity Estimation Error');

subplot(2,2,4); set(gca,'FontSize',12);
plot(k, xArr(4,:)-xhatArr(4,:), 'b-', 'LineWidth', 2); xlabel('time');
title('East Velocity Estimation Error');

% Plot UKF variance estimates
figure; set(gcf, 'Color', 'White'); set(gca,'FontSize',12); hold on; box
on;
plot(k, sqrt(squeeze(PArr(1,1,:))), 'r:', 'LineWidth', 2);
plot(k, sqrt(squeeze(PArr(2,2,:))), 'b--', 'LineWidth', 2);
plot(k, sqrt(squeeze(PArr(3,3,:))), 'k-', 'LineWidth', 2);
plot(k, sqrt(squeeze(PArr(4,4,:))), 'm-.', 'LineWidth', 2);
legend('North Position', 'East Position', 'North Velocity', 'East
Velocity');
xlabel('time');
ylabel('UKF Std Dev Estimate');

% Compare experimental and theoretical standard deviations of estimation
errors.
EstStd = std(xArr' - xhatArr');
disp(['Experimental std dev of est err = ', num2str(EstStd)]);
```

```
function PendulumUKF

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 14.15

% Inverted pendulum simulation - UKF and EKF estimation

dt = 0.005; % integration step size
tf = 2; % simulation time
thetainit = 0.1; % initial pendulum angle (radians)
m = 0.2; % pendulum mass (kg)
M = 1.0; % cart mass (kg)
g = 9.81; % acceleration due to gravity (m/s^2)
B = 0.1; % coefficient of viscous friction on cart wheels
l = 1; % length of pendulum (meters)
r = 0.02; % radius of pendulum mass (meters)
J = m * r * r / 2; % moment of inertia of pendulum mass (cylinder)

theta = thetainit; % theta = angle pendulum makes with vertical
thetadot = 0;
thetadotdot = 0;
d = 0; % d = horizontal displacement of cart
ddot = 0;
ddotdot = 0;

theta_array = [theta];
thetadot_array = [thetadot];
thetadotdot_array = [thetadotdot];
d_array = [d];
ddot_array = [ddot];
ddotdot_array = [ddotdot];

% Create Jacobian matrix for EKF
temp = J * (M + m) + M * m * l^2;
F22 = -m^2 * l^2 * B / (M + m) / temp - B / (M + m);
F23 = -m^2 * l^2 * g / temp;
F42 = m * l * B / temp;
F43 = m * g * l * (M + m) / temp;
F = [0 1 0 0;
    0 F22 F23 0;
    0 0 0 1;
    0 F42 F43 0];
% Measurement matrix
H = [1 0 0 0];

% Continuous time process noise covariance
Q = diag([0 0.0004 0 0.04]);
% Discrete time process noise covariance
Q = Q * dt;
% Discrete time measurement noise covariance
```

```
R = diag([0.01]);
r = size(R,1); % number of measurements

x = [0; 0; thetainit; 0]; % initial state
xhatplus = x; % initial UKF estimate
xhatKplus = x; % initial EKF estimate
% Initial estimation error covariance - must be positive definite for UKF
Pplus = 0.00001 * eye(4);
PKplus = Pplus;

% Initialize arrays
xArr = x;
xhatArr = xhatplus;
xhatKArr = xhatKplus;

N = length(x);
for t = dt : dt : tf
    % System simulation
    theta_old = theta;
    thetadot_old = thetadot;
    thetadotdot_old = thetadotdot;
    d_old = d;
    ddot_old = ddot;
    ddotdot_old = ddotdot;
    sine = sin(theta);
    cosine = cos(theta);
    % Compute the new state values.
    u = 40 * theta_old;
    theta = theta + thetadot_old * dt;
    thetadot = thetadot + thetadotdot_old * dt + sqrt(Q(3,3)) * randn *
dt;
    thetadotdot = (m*g*l*sine - m*l*cosine*(u + ...
        m*l*thetadot_old*thetadot_old*sine - B*ddot_old) / (M + m)) / ...
        (J + m*l*l - m*m*l*l*cosine*cosine / (M + m)) + sqrt(Q(4,4)) *
randn * dt;
    d = d + ddot_old * dt;
    ddot = ddot + ddotdot_old * dt + sqrt(Q(1,1)) * randn * dt;
    ddotdot = (u - m*l*thetadotdot_old*cosine + ...
        m*l*thetadot_old*thetadot_old*sine - B*ddot_old) / (M + m) + sqrt
(Q(2,2)) * randn * dt;
    % Save the state values in arrays for later plotting.
    theta_array = [theta_array theta];
    thetadot_array = [thetadot_array thetadot];
    thetadotdot_array = [thetadotdot_array thetadotdot];
    d_array = [d_array d];
    ddot_array = [ddot_array ddot];
    ddotdot_array = [ddotdot_array ddotdot];
    % Measurement simulation
    y = d + sqrt(R) * randn;
    % UKF time update
```

```
    [root,p] = chol(N*Pplus);
    for i = 1 : N
        sigma(:,i) = xhatplus + root(i,:)';
        sigma(:,i+N) = xhatplus - root(i,:)';
    end
    for i = 1 : 2*N
        % Save the old a posteriori sigma points.
        theta_old = sigma(3,i);
        thetadot_old = sigma(4,i);
        d_old = sigma(1,i);
        ddot_old = sigma(2,i);
        sine = sin(theta_old);
        cosine = cos(theta_old);
        % Compute the new sigma points (a priori).
        u = 40 * theta_old;
        sigma(3,i) = sigma(3,i) + thetadot_old * dt;
        thetadotdot_old = m * g * l * theta_old * (M + m);
        thetadotdot_old = thetadotdot_old - m * l * (u - B * ddot_old);
        thetadotdot_old = thetadotdot_old / temp;
        sigma(4,i) = sigma(4,i) + thetadotdot_old * dt;
        sigma(1,i) = sigma(1,i) + ddot_old * dt;
        ddotdot_old = (u - m * l * thetadotdot_old - B * ddot_old) / (M +↙
m);
        sigma(2,i) = sigma(2,i) + ddotdot_old * dt;
    end
    xhatminus = zeros(N,1);
    for i = 1 : 2*N
        xhatminus = xhatminus + sigma(:,i) / 2 / N;
    end
    Pminus = zeros(N,N);
    for i = 1 : 2*N
        Pminus = Pminus + (sigma(:,i) - xhatminus) * (sigma(:,i) -↙
xhatminus)' / 2 / N;
    end
    Pminus = Pminus + Q;
    % UKF measurement update
    [root,p] = chol(N*Pminus);
    for i = 1 : N
        sigma(:,i) = xhatminus + root(i,:)';
        sigma(:,i+N) = xhatminus - root(i,:)';
    end
    for i = 1 : 2*N
        nhat = sigma(1,i);
        ehat = sigma(2,i);
        yukf(1,i) = sigma(1,i);
    end
    yhat = 0;
    for i = 1 : 2*N
        yhat = yhat + yukf(:,i) / 2 / N;
    end
```

```
    Py = zeros(r,r);
    Pxy = zeros(N,r);
    for i = 1 : 2*N
        Py = Py + (yukf(:,i) - yhat) * (yukf(:,i) - yhat)' / 2 / N;
        Pxy = Pxy + (sigma(:,i) - xhatminus) * (yukf(:,i) - yhat)' / 2 /
N;
    end
    Py = Py + R;
    Kukf = Pxy * inv(Py);
    xhatplus = xhatminus + Kukf * (y - yhat);
    Pplus = Pminus - Kukf * Py * Kukf';
    % Save data for plotting
    xhatArr = [xhatArr xhatplus];
    % Extended Kalman filter simulation
    % Save the old a posteriori state estimate.
    theta_old = xhatKplus(3);
    thetadot_old = xhatKplus(4);
    d_old = xhatKplus(1);
    ddot_old = xhatKplus(2);
    sine = sin(theta_old);
    cosine = cos(theta_old);
    % EKF time update.
    % The EKF works only if "knows" u perfectly.
    u = 40 * theta_old;
    %u = 40 * theta;
    xhatK(3,1) = xhatKplus(3) + thetadot_old * dt;
    thetadotdot_old = m * g * l * theta_old * (M + m);
    thetadotdot_old = thetadotdot_old - m * l * (u - B * ddot_old);
    thetadotdot_old = thetadotdot_old / temp;
    xhatK(4,1) = xhatKplus(4) + thetadotdot_old * dt;
    xhatK(1,1) = xhatKplus(1) + ddot_old * dt;
    ddotdot_old = (u - m * l * thetadotdot_old - B * ddot_old) / (M + m);
    xhatK(2,1) = xhatKplus(2) + ddotdot_old * dt;
    PK = PKplus + (F * PKplus + PKplus * F') * dt + Q;
    % EKF measurement update.
    K = PK * H' * inv(H * PK * H' + R);
    xhatKplus = xhatK + K * (y - H * xhatK);
    PKplus = (eye(N) - K * H) * PK * (eye(N) - K * H)' + K * R * K';
    % Save date for plotting.
    xhatKArr = [xhatKArr xhatKplus];
end

% Plot results
close all;
t = 0 : dt : tf;
figure; set(gcf,'Color','White');

subplot(2,2,1); set(gca,'FontSize',12);
plot(t, 180/2/pi*theta_array, 'b-', 'LineWidth', 2); hold on;
plot(t, 180/2/pi*xhatArr(3,:), 'r:', 'LineWidth', 2);
```

```
%plot(t, 180/2/pi*xhatKArr(3,:), 'm--', 'LineWidth', 2);
ylabel('angle (deg)');

subplot(2,2,2); set(gca,'FontSize',12);
plot(t, d_array, 'b-', 'LineWidth', 2); hold on;
plot(t, xhatArr(1,:), 'r:', 'LineWidth', 2);
%plot(t, xhatKArr(1,:), 'm--', 'LineWidth', 2);
ylabel('cart pos (meters)');
legend('true', 'UKF estimate');

subplot(2,2,3); set(gca,'FontSize',12);
plot(t, thetadot_array, 'b-', 'LineWidth', 2); hold on;
plot(t, xhatArr(4,:), 'r:', 'LineWidth', 2);
%plot(t, xhatKArr(4,:), 'm--', 'LineWidth', 2);
ylabel('ang vel (deg/s)');
xlabel('time (s)');

subplot(2,2,4); set(gca,'FontSize',12);
plot(t, ddot_array, 'b-', 'LineWidth', 2); hold on;
plot(t, xhatArr(2,:), 'r:', 'LineWidth', 2);
%plot(t, xhatKArr(2,:), 'm--', 'LineWidth', 2);
ylabel('cart vel (meters/s)');
xlabel('time (s)');
```

```
function Hypersphere

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 15.11

% Plot the volume of a hypersphere as a function of dimension.

v(1) = 2;
v(2) = pi;
v(3) = 4*pi/3;
for i = 4 : 20
    v(i) = 2*pi*v(i-2)/i;
end
i = 1 : 20;
close all;
figure; set(gca,'FontSize',12); set(gcf,'Color','White');
plot(i, v);
xlabel('number of dimensions');
ylabel('volume of hypersphere');
```

```
function Kernel(BWScale, KernelType)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 15.12, 15.13, 15.14

% Epanechnikov-based pdf approximation

% INPUTS:
%     BWScale is used to scale the bandwidth from its optimal value:
%                    Bandwidth h = BWScale * h^*
%     KernelType =   1 (Epanechnikov)
%                    2 (Gaussian)
%                    3 (Uniform)
%                    4 (Triangular)
%                    5 (Biweight)

if ~exist('BWScale', 'var')
    BWScale = 1;
end
if ~exist('KernelType', 'var')
    KernelType = 1;
end

close all;
figure;

subplot(2,1,1); hold on;
EpanechSub([1 2], BWScale, KernelType);
subplot(2,1,2); hold on;
EpanechSub([1 2 3], BWScale, KernelType);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function EpanechSub(x, BWScale, KernelType)

q = ones(size(x)); % probabilities
NReg = 100; % number of probability bins for pdf approximation
n = 1; % dimension of the state vector
vol = 2; % volume of unit hypersphere in d-dimensional space
N = length(x); % number of particles in the particle filter
S = cov(x);
A = chol(S)';
A = eye(length(x));
h = (8 * vol^(-1) * (n + 4) * (2 * sqrt(pi))^n)^(1 / (n + 4)) * N^(-1 / (n
+ 4)); % bandwidth of RPF
h = BWScale * h;
% Define the domain from which we will choose a posteriori particles for
% the regularized particle filter.
xreg(1) = min(x) - std(x);
xreg(NReg) = max(x) + std(x);
dx = (xreg(NReg) - xreg(1)) / (NReg - 1);
```

```
for i = 2 : NReg - 1
    xreg(i) = xreg(i-1) + dx;
end
% Create the pdf approximation that is required for the regularized
% particle filter.
for j = 1 : NReg
    for i = 1 : N
        qreg(i,j) = 0;
        normx = norm(inv(A) * (xreg(j) - x(i)));
        if KernelType == 2
            qreg(i,j) = qreg(i,j) + q(i) * exp(-normx^2 / h^2 / 2) / h^n /↙
det(A);
        elseif normx < h
            switch KernelType
                case 1
                    qreg(i,j) = qreg(i,j) + q(i) * (n + 2) * (1 - normx^2↙
/ h^2) / 2 / vol / h^n / det(A);
                case 3
                    qreg(i,j) = qreg(i,j) + q(i) / 2 / h^n / det(A);
                case 4
                    qreg(i,j) = qreg(i,j) + q(i) * (1 - normx / h) / h^n /↙
det(A);
                case 5
                    qreg(i,j) = qreg(i,j) + q(i) * (1 - normx^2 / h^2)^2 *↙
15 / 16 / h^n / det(A);
            end
        end
    end
end
% Plot
for i = 1 : N
    plot([x(i) x(i)], [0 q(i)], 'k-');
    plot(x(i), q(i), 'ko');
    plot(xreg, qreg(i,:), 'r:');
end
plot(xreg, sum(qreg,1), 'b--');
set(gca,'FontSize',12); set(gcf,'Color','White');
xlabel('state estimate'); ylabel('pdf');
return;
```

```
function [errRMSKalman, errRMSParticle] = PartScalar(N, Q)

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 15.15

% EKF and Particle filter scalar example,
% adapted from Gordon, Salmond, and Smith paper.
% INPUTS:
%    N = number of particles
%    Q = process noise variance
% OUTPUTS:
%    errRMSKalman = Kalman filter RMS estimation error
%    errRMSParticle = Particle filter RMS estimation error

if ~exist('N', 'var')
    N = 100;
end
if ~exist('Q', 'var')
    Q = 1;
end

x = 0.1; % initial state
R = 1; % measurement noise covariance
tf = 50; % simulation length

xhat = x;
P = 2;
xhatPart = x;

% Initialize the particle filter.
for i = 1 : N
    xpart(i) = x + sqrt(P) * randn;
end

xArr = [x];
yArr = [x^2 / 20 + sqrt(R) * randn];
xhatArr = [xhat];
xhatPartArr = [xhatPart];

close all;

for k = 1 : tf
    % System simulation
    x = 0.5 * x + 25 * x / (1 + x^2) + 8 * cos(1.2*(k-1)) + sqrt(Q) *
randn;
    y = x^2 / 20 + sqrt(R) * randn;
    % Extended Kalman filter
    F = 0.5 + 25 * (1 - xhat^2) / (1 + xhat^2)^2;
    P = F * P * F' + Q;
    H = xhat / 10;
```

```
    K = P * H' * (H * P * H' + R)^(-1);
    xhat = 0.5 * xhat + 25 * xhat / (1 + xhat^2) + 8 * cos(1.2*(k-1));
    xhat = xhat + K * (y - xhat^2 / 20);
    P = (1 - K * H) * P;
    % Particle filter
    for i = 1 : N
        xpartminus(i) = 0.5 * xpart(i) + 25 * xpart(i) / (1 + xpart(i)^2)
+ 8 * cos(1.2*(k-1)) + sqrt(Q) * randn;
        ypart = xpartminus(i)^2 / 20;
        vhat = y - ypart;
        q(i) = (1 / sqrt(R) / sqrt(2*pi)) * exp(-vhat^2 / 2 / R);
    end
    % Normalize the likelihood of each a priori estimate.
    qsum = sum(q);
    if qsum < eps
        q = ones(size(q)) / N;
    else
        for i = 1 : N
            q(i) = q(i) / qsum;
        end
    end
    % Resample.
    for i = 1 : N
        u = rand; % uniform random number between 0 and 1
        qtempsum = 0;
        for j = 1 : N
            qtempsum = qtempsum + q(j);
            if qtempsum >= u
                xpart(i) = xpartminus(j);
                break;
            end
        end
    end
    % The particle filter estimate is the mean of the particles.
    xhatPart = mean(xpart);
    % Save data in arrays for later plotting
    xArr = [xArr x];
    yArr = [yArr y];
    xhatArr = [xhatArr xhat];
    xhatPartArr = [xhatPartArr xhatPart];
end
errRMSKalman = sqrt((norm(xArr - xhatArr))^2 / tf);
errRMSParticle = sqrt((norm(xArr - xhatPartArr))^2 / tf);
```

```
function PartScalarMonte

% Optimal State Estimation Solution Manual, by Dan Simon
% Problem 15.15

% This is a routine to generate Monte Carlo simulation results for the
% EKF and Particle filter scalar example that is
% adapted from the Gordon, Salmond, and Smith paper.

NMonte = 100; % number of Monte Carlo simulations

N = 100; % number of particles
Q = 0.1; % process noise variance
for i = 1 : NMonte
    [errKalman(i), errParticle(i)] = PartScalar(N, Q);
    fprintf('.');
end
fprintf('\n');
disp(['N = ', num2str(N), ', Q = ', num2str(Q), ', Ave RMS Est Err = ',
num2str(mean(errKalman)), ' (Kalman), ', num2str(mean(errParticle)), '
(Particle)']);

N = 100;
Q = 1;
for i = 1 : NMonte
    [errKalman(i), errParticle(i)] = PartScalar(N, Q);
    fprintf('.');
end
fprintf('\n');
disp(['N = ', num2str(N), ', Q = ', num2str(Q), ', Ave RMS Est Err = ',
num2str(mean(errKalman)), ' (Kalman), ', num2str(mean(errParticle)), '
(Particle)']);

N = 100;
Q = 10;
for i = 1 : NMonte
    [errKalman(i), errParticle(i)] = PartScalar(N, Q);
    fprintf('.');
end
fprintf('\n');
disp(['N = ', num2str(N), ', Q = ', num2str(Q), ', Ave RMS Est Err = ',
num2str(mean(errKalman)), ' (Kalman), ', num2str(mean(errParticle)), '
(Particle)']);

N = 100;
Q = 100;
for i = 1 : NMonte
    [errKalman(i), errParticle(i)] = PartScalar(N, Q);
    fprintf('.');
end
```

```
fprintf('\n');
disp(['N = ', num2str(N), ', Q = ', num2str(Q), ', Ave RMS Est Err = ',
num2str(mean(errKalman)), ' (Kalman), ', num2str(mean(errParticle)), '
(Particle)']);

N = 10;
Q = 1;
for i = 1 : NMonte
    [errKalman(i), errParticle(i)] = PartScalar(N, Q);
    fprintf('.');
end
fprintf('\n');
disp(['N = ', num2str(N), ', Q = ', num2str(Q), ', Ave RMS Est Err = ',
num2str(mean(errKalman)), ' (Kalman), ', num2str(mean(errParticle)), '
(Particle)']);

N = 1000;
Q = 1;
for i = 1 : NMonte
    [errKalman(i), errParticle(i)] = PartScalar(N, Q);
    fprintf('.');
end
fprintf('\n');
disp(['N = ', num2str(N), ', Q = ', num2str(Q), ', Ave RMS Est Err = ',
num2str(mean(errKalman)), ' (Kalman), ', num2str(mean(errParticle)), '
(Particle)']);
```