AAE 567 Spring 2018 Homework 3 Solutions

March 18, 2018

3.3.1 Problem 1

We will use the matrix inversion lemma 2.4.1 from chapter 2 to show that:

$$P_{\mathcal{H}}f = P_{\mathcal{M}}f + R_{f\varphi}R_{\varphi}^{-1}\varphi$$
$$E(f - P_{\mathcal{H}}f)(f - P_{\mathcal{H}}f)^* = E(f - P_{\mathcal{M}}f)(f - P_{\mathcal{M}}f)^* - R_{f\varphi}R_{\varphi}^{-1}R_{\varphi f}$$

The matrix inversion lemma states that for

$$T = \begin{bmatrix} R & M \\ N & Q \end{bmatrix}$$

we have

$$T^{-1} = \begin{bmatrix} R^{-1} + R^{-1}M\Delta^{-1}NR^{-1} & -R^{-1}M\Delta^{-1} \\ -\Delta^{-1}NR^{-1} & \Delta^{-1} \end{bmatrix}$$

with $\Delta = Q - NR^{-1}M$.

Now, as the problem suggests, consider the space $\mathcal{H}$ spanned by g and y with $h = [g\ y]^{tr}$. We then have

$$R_h = \begin{bmatrix} R_g & R_{gy} \\ R_{yg} & R_y \end{bmatrix}$$
$$R_{fh} = \begin{bmatrix} R_{fg} & R_{fy} \end{bmatrix}$$

As suggested, we observe that the Schur compement of R_h is given by $R_{\varphi} = \Delta = R_y - R_{yg}R_g^{-1}R_{gy}$. We are allowed to assume that R_{φ} and thus R_h is invertible. Thus, we have

$$P_{\mathcal{H}}f = R_{fh}R_h^{-1}h$$
$$E(f - P_{\mathcal{H}}f)(f - P_{\mathcal{H}}f)^* = R_f - R_{fh}R_h^{-1}R_{hf}$$

Since we are allowed to assume that R_h is invertible, we use the matrix inversion lemma with $T = R_h$ to see that

$$R_h^{-1} = \begin{bmatrix} R_g^{-1} + R_g^{-1} R_{gy} R_\varphi^{-1} R_{yg} R_g^{-1} & -R_g^{-1} R_{gy} R_\varphi^{-1} \\ -R_\varphi^{-1} R_{yg} R_g^{-1} & R_\varphi^{-1} \end{bmatrix}$$

The expression we get is then

$$\begin{aligned} P_{\mathcal{H}} f &= R_{fh} R_h^{-1} h \\ &= \begin{bmatrix} R_{fg} & R_{fy} \end{bmatrix} \begin{bmatrix} R_g^{-1} + R_g^{-1} R_{gy} R_\varphi^{-1} R_{yg} R_g^{-1} & -R_g^{-1} R_{gy} R_\varphi^{-1} \\ -R_\varphi^{-1} R_{yg} R_g^{-1} & R_\varphi^{-1} \end{bmatrix} \begin{bmatrix} g \\ y \end{bmatrix} \\ &= \begin{bmatrix} R_{fg} & R_{fy} \end{bmatrix} \begin{bmatrix} R_g^{-1} g + R_g^{-1} R_{gy} R_\varphi^{-1} R_{yg} R_g^{-1} g - R_g^{-1} R_{gy} R_\varphi^{-1} y \\ -R_\varphi^{-1} R_{yg} R_g^{-1} g + R_\varphi^{-1} y \end{bmatrix} \\ &= R_{fg} R_g^{-1} g + R_{fg} R_g^{-1} R_{gy} R_\varphi^{-1} R_{yg} R_g^{-1} g - R_{fg} R_g^{-1} R_{gy} R_\varphi^{-1} y \\ &\quad - R_{fy} R_\varphi^{-1} R_{yg} R_g^{-1} g + R_{fy} R_\varphi^{-1} y \\ &= P_{\mathcal{M}} f + R_{fg} R_g^{-1} R_{gy} R_\varphi^{-1} P_{\mathcal{M}} y - R_{fg} R_g^{-1} R_{gy} R_\varphi^{-1} y \\ &\quad - R_{fy} R_\varphi^{-1} P_{\mathcal{M}} y + R_{fy} R_\varphi^{-1} y \\ &= P_{\mathcal{M}} f - R_{fg} R_g^{-1} R_{gy} R_\varphi^{-1} \varphi + R_{fy} R_\varphi^{-1} \varphi \\ &= P_{\mathcal{M}} f + (R_{fy} - R_{fg} R_g^{-1} R_{gy}) R_\varphi^{-1} \varphi \end{aligned}$$

where we have made use of $\varphi = y - P_{\mathcal{M}} y$ and $P_{\mathcal{M}} y = R_{yg} R_g^{-1} g$. Now it just remains to be shown that $R_{fy} - R_{fg} R_g^{-1} R_{gy} = R_{f\varphi}$. We note that

$$\begin{aligned} R_{f\varphi} &= E(f\varphi^*) && (1) \\ &= E[f(y - P_{\mathcal{M}} y)^*] && (2) \\ &= E[fy^*] - E[f(P_{\mathcal{M}} y)^*] && (3) \\ &= R_{fy} - E[f(P_{\mathcal{M}} y)^*] && (4) \\ &= R_{fy} - E[f(R_{yg} R_g^{-1} g)^*] && (5) \\ &= R_{fy} - E[fg^*](R_{yg} R_g^{-1})^* && (6) \\ &= R_{fy} - R_{fg} R_g^{-1} R_{yg} && (7) \\ & && (8) \end{aligned}$$

This shows that $P_{\mathcal{H}} f = P_{\mathcal{M}} f + R_{f\varphi} R_\varphi^{-1} \varphi$ and completes the first half of the problem.

Now we wish to show that

$$E(f - P_{\mathcal{H}} f)(f - P_{\mathcal{H}} f)^* = E(f - P_{\mathcal{M}} f)(f - P_{\mathcal{M}} f)^* - R_{f\varphi} R_\varphi^{-1} R_{\varphi f}$$

To accomplish this we substitute in $P_{\mathcal{H}}f = P_{\mathcal{M}}f + R_{f\varphi}R_\varphi^{-1}\varphi$ to obtain

$$
\begin{aligned}
&E(f - P_{\mathcal{H}}f)(f - P_{\mathcal{H}}f)^* = E[f - (P_{\mathcal{M}}f + R_{f\varphi}R_\varphi^{-1}\varphi)][f - (P_{\mathcal{M}}f + R_{f\varphi}R_\varphi^{-1}\varphi)]^* \\
&\quad = E[(f - P_{\mathcal{M}}f) - R_{f\varphi}R_\varphi^{-1}\varphi][(f - P_{\mathcal{M}}f)^* - (R_{f\varphi}R_\varphi^{-1}\varphi)^*] \\
&\quad = E[(f - P_{\mathcal{M}}f)(f - P_{\mathcal{M}}f)^*] + E[(R_{f\varphi}R_\varphi^{-1}\varphi)(R_{f\varphi}R_\varphi^{-1}\varphi)^*] \\
&\qquad - E[(f - P_{\mathcal{M}}f)(R_{f\varphi}R_\varphi^{-1}\varphi)^*] - E[(R_{f\varphi}R_\varphi^{-1}\varphi)(f - P_{\mathcal{M}}f)^*]
\end{aligned}
$$

At this point we note that

$$E[(R_{f\varphi}R_\varphi^{-1}\varphi)(R_{f\varphi}R_\varphi^{-1}\varphi)^*] = R_{f\varphi}R_\varphi^{-1}R_{\varphi f}$$

and

$$-E[(f - P_{\mathcal{M}}f)(R_{f\varphi}R_\varphi^{-1}\varphi)^*] - E[(R_{f\varphi}R_\varphi^{-1}\varphi)(f - P_{\mathcal{M}}f)^*] = 0$$

where the second of these equations stems from the fact that $P_{\mathcal{M}}f$ is orthogonal to φ due to the fact that g is orthogonal to φ. Then, we have shown, as desired:

$$E(f - P_{\mathcal{H}}f)(f - P_{\mathcal{H}}f)^* = E(f - P_{\mathcal{M}}f)(f - P_{\mathcal{M}}f)^* - R_{f\varphi}R_\varphi^{-1}R_{\varphi f}$$

3.3.1 Problem 2

We have the state space system

$$
\begin{aligned}
x(n+1) &= Ax(n) + Bu(n) \\
y(n) &= Cx(n) + Dv(n)
\end{aligned}
$$

where $u(n)$ and $v(n)$ are independent Gaussian white noise independent of $x(0)$. $\mathcal{M}_n$ is the linear span of $\{y(k)\}_0^n$ and the optimal state estimate is $\hat{x}(n) = P_{\mathcal{M}_{n-1}}x(n)$. We wish to find the state estimate $P_{\mathcal{M}_n}x(n)$ in terms of $\hat{x}(n)$ and $y(n)$.

From Lemma 3.3.1 we have, for $\varphi(n) = y(n) - P_{\mathcal{M}_{n-1}}y(n)$.

$$P_{\mathcal{M}_n}f = P_{\mathcal{M}_{n-1}}f + R_{f\varphi(n)}R_{\varphi(n)}^{-1}\varphi(n)$$

Rewriting this specifically for our case, we have

$$
\begin{aligned}
P_{\mathcal{M}_n}x(n) &= P_{\mathcal{M}_{n-1}}x(n) + R_{x(n)\varphi(n)}R_{\varphi(n)}^{-1}\varphi(n) \\
&= \hat{x}(n) + R_{x(n)\varphi(n)}R_{\varphi(n)}^{-1}\varphi(n) \\
&= \hat{x}(n) + E(x(n)\varphi(n)^*)E(\varphi(n)\varphi(n)^*)^{-1}\varphi(n)
\end{aligned}
$$

At this point we take note of the following equalities:

$$\begin{aligned}
\varphi(n) &= y(n) - C\hat{x}(n) \\
&= C\tilde{x}(n) + Dv(n) \\
y(n) &= Cx(n) + Dv(n) \\
x(n) &= \hat{x}(n) + \tilde{x}(n) \\
E\tilde{x}(n)\tilde{x}(n)^* &= Q_n
\end{aligned}$$

Using the above equalities, we obtain

$$\begin{aligned}
E(x(n)\varphi(n)^*) &= E[(\hat{x}(n) + \tilde{x}(n))(C\tilde{x}(n) + Dv(n))^*] \\
&= E[\tilde{x}(n)(C\tilde{x}(n) + Dv(n))^*] \\
&= E[\tilde{x}(n)(C\tilde{x}(n))^*] + E[\tilde{x}(n)(Dv(n))^*] \\
&= E\tilde{x}(n)\tilde{x}(n)^*C^* \\
&= Q_nC^*
\end{aligned}$$

and

$$\begin{aligned}
E(\varphi(n)\varphi(n)^*) &= E[(C\tilde{x}(n) + Dv(n))(C\tilde{x}(n) + Dv(n))^*] \\
&= E[C\tilde{x}(n)(C\tilde{x}(n))^*] + E[Dv(n)(Dv(n))^*] \\
&= E[C\tilde{x}(n)\tilde{x}(n)^*C^*] + E[Dv(n)v(n)^*D^*] \\
&= CQ_nC^* + DD^*
\end{aligned}$$

Plugging in the expressions we have just derived, we obtain

$$P_{\mathcal{M}_n}x(n) = \hat{x}(n) + Q_nC^*(CQ_nC^* + DD^*)^{-1}(y(n) - C\hat{x}(n))$$

3.3.1 Problem 3

We have the system

$$x(n+1) = ax(n) + u(n) \text{ and } y(n) = x(n) + v(n)$$

where a is scalar and $\{u(0), v(0), v(1), x(0)\}$ are all independent mean zero variance one Guassian random variables. Let $\mathcal{M}_0 = \text{span}\{y(0)\}$ and $\mathcal{M}_1 = \text{span}\{y(0), y(1)\}$, and find $\hat{x}(0) = P_{\mathcal{M}_1}x(0)$, $E|x(0) - \hat{x}(0)|^2$, and α and β such that $\phi(1) = y(1) - P_{\mathcal{M}_1}y(1) = \alpha y(1) + \beta y(0)$

i)

We first write out

$$
\begin{aligned}
x(1) &= ax(0) + u(0) \\
y(0) &= x(0) + v(0) \\
y(1) &= ax(0) + u(0) + v(1) \\
g = \begin{bmatrix} y(0) \\ y(1) \end{bmatrix} &= \begin{bmatrix} x(0) + v(0) \\ ax(0) + u(0) + v(1) \end{bmatrix}
\end{aligned}
$$

Notice that $\widehat{x}(0) = R_{x(0)g} R_g^{-1} g$. Moreover,

$$
\begin{aligned}
R_{x(0)g} &= Ex(0) \begin{bmatrix} x(0) + v(0) & ax(0) + u(0) + v(1) \end{bmatrix} \\
&= \begin{bmatrix} Ex(0)x(0) + Ex(0)v(0) & \quad aEx(0)x(0) + Ex(0)u(0) + Ex(0)v(1) \end{bmatrix} \\
&= \begin{bmatrix} 1 & a \end{bmatrix}
\end{aligned}
$$

Moreover, we have

$$
\begin{aligned}
R_g &= E \begin{bmatrix} y(0) \\ y(1) \end{bmatrix} \begin{bmatrix} y(0) & y(1) \end{bmatrix} = \begin{bmatrix} Ey(0)y(0) & Ey(0)y(1) \\ Ey(1)y(0) & Ey(1)y(1) \end{bmatrix} \\
&= \begin{bmatrix} E\big(x(0) + v(0)\big)^2 & E\big(x(0) + v(0)\big)\big(ax(0) + u(0) + v(1)\big) \\ E\big(ax(0) + u(0) + v(1)\big)\big(x(0) + v(0)\big) & E\big(ax(0) + u(0) + v(1)\big)^2 \end{bmatrix} \\
&= \begin{bmatrix} 2 & a \\ a & a^2 + 2 \end{bmatrix}
\end{aligned}
$$

In other words,

$$
R_g = \begin{bmatrix} 2 & a \\ a & a^2 + 2 \end{bmatrix} \quad \text{and} \quad R_g^{-1} = \frac{1}{a^2 + 4} \begin{bmatrix} a^2 + 2 & -a \\ -a & 2 \end{bmatrix}
$$

This readily implies that

$$
\widehat{x}(0) = R_{x(0)g} R_g^{-1} g = \frac{1}{a^2 + 4} \begin{bmatrix} 1 & a \end{bmatrix} \begin{bmatrix} a^2 + 2 & -a \\ -a & 2 \end{bmatrix} g = \frac{1}{a^2 + 4} \begin{bmatrix} 2 & a \end{bmatrix} \begin{bmatrix} y(0) \\ y(1) \end{bmatrix}
$$

Therefore

$$
\boxed{\boxed{\widehat{x}(0) = \frac{2y(0) + ay(1)}{a^2 + 4}}}
$$

ii)

The error is given by

$$\begin{aligned}
E(x(0)-\widehat{x}(0))^2 &= R_{x(0)} - R_{x(0)g}R_g^{-1}R_{x(0)g}^* \\
&= 1 - \frac{1}{a^2+4}\begin{bmatrix}1 & a\end{bmatrix}\begin{bmatrix}a^2+2 & -a \\ -a & 2\end{bmatrix}R_{x(0)g}^* \\
&= 1 - \frac{1}{a^2+4}\begin{bmatrix}2 & a\end{bmatrix}\begin{bmatrix}1 \\ a\end{bmatrix} \\
&= 1 - \frac{2+a^2}{a^2+4} = \frac{a^2+4-2-a^2}{a^2+4} = \frac{2}{a^2+4}
\end{aligned}$$

Therefore the error

$$\boxed{E(x(0)-\widehat{x}(0))^2 = \frac{2}{a^2+4}}$$

iii)

To compute $\varphi(1) = y(1) - P_{\mathcal{M}_0}y(1)$ observe that

$$P_{\mathcal{M}_0}y(1) = R_{y(1)y(0)}R_{y(0)}^{-1}y(0) = \frac{Ey(1)y(0)}{Ey(0)^2}y(0)$$

Notice that

$$\begin{aligned}
Ey(1)y(0) &= E\big(x(0)+v(0)\big)\big(ax(0)+u(0)+v(1)\big) = a \\
Ey(0)^2 &= E\big(x(0)+v(0)\big)^2 = 2 \\
P_{\mathcal{M}_0}y(1) &= \frac{ay(0)}{2}
\end{aligned}$$

Therefore

$$\boxed{\varphi(1) = y(1) - \frac{ay(0)}{2}}$$

3.6.2 Problem 1

We are given the system

$$x(n+1) = \begin{bmatrix} 0 & 1 \\ -0.98 & 1.94 \end{bmatrix} x(n) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(n)$$
$$y(n) = \begin{bmatrix} 3 & 4 \end{bmatrix} x(n) + v(n)$$

with u and v independent Gaussian white noise processes orthogonal to $x(0) = x_0$ (which is a mean zero random variable). We are also given that the true initial value is $x_0 = \begin{bmatrix} 1 & 2 \end{bmatrix}^{tr}$. We wish to implement the steady state Kalman filter for $0 \le n \le 500$ with initial estimate $\hat{x}(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^{tr}$ and compare the estimate to the actual state.

From equation 6.6 in section 3.6.1 we have the steady state Kalman filter given by

$$\hat{\chi}(n+1) = (A - K_P C)\hat{\chi}(n) + K_P y(n)$$
$$K_P = APC^*(CPC^* + DD^*)^{-1}$$

where P is the (positive) solution to the algebraic Ricatti equation $P = APA^* + BB^* - APC^*(CPC^* + DD^*)^{-1}CPA^*$. While an initial condition is not specified, we can safely assume that we do not actually know the state of the system, so we will take the mean of the initial state distribution as our initial estimate and begin with $\hat{\chi}(0) = [0 \quad 0]^T$.

We then implement the Kalman filter as in Theorem 3.1.2. The results of the simulation are plotted in Figure 2. As can be seen, the estimates closely track the actual state components. Note that the specific shape of the curves will vary with each time the simulation is performed due to the stochastic components of the state evolution and measurement equations, but the estimates should still track the state roughly as closely as in Figure 2. The following Matlab script performs the desired computations and generates the plots:

```
%3.6.2 #1

%system matrices
A = [0 1; -0.98 1.94];
B = [0; 1];
C = [3 4];
D = 1;
```

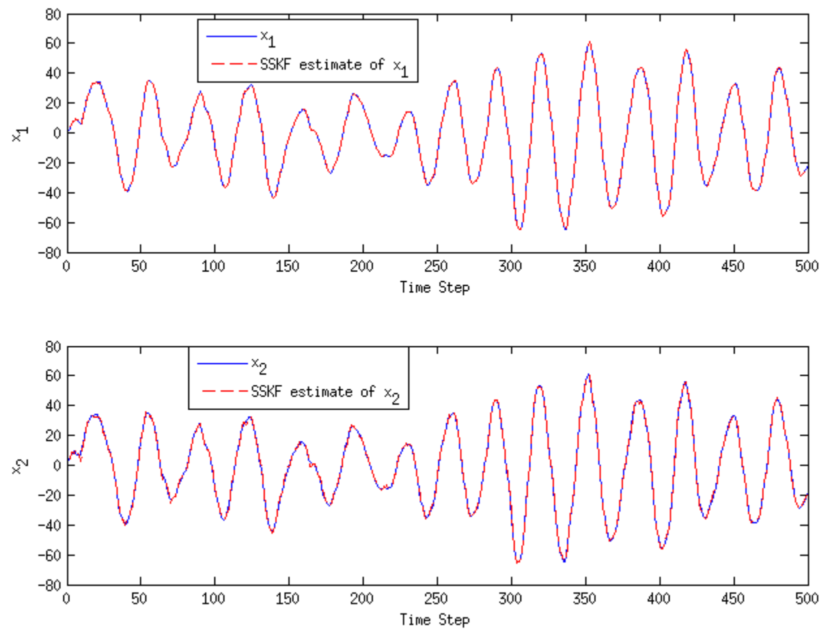

Figure 1: Plot of the actual state and SSKF estimated state for 3.6.2 Exercise 1

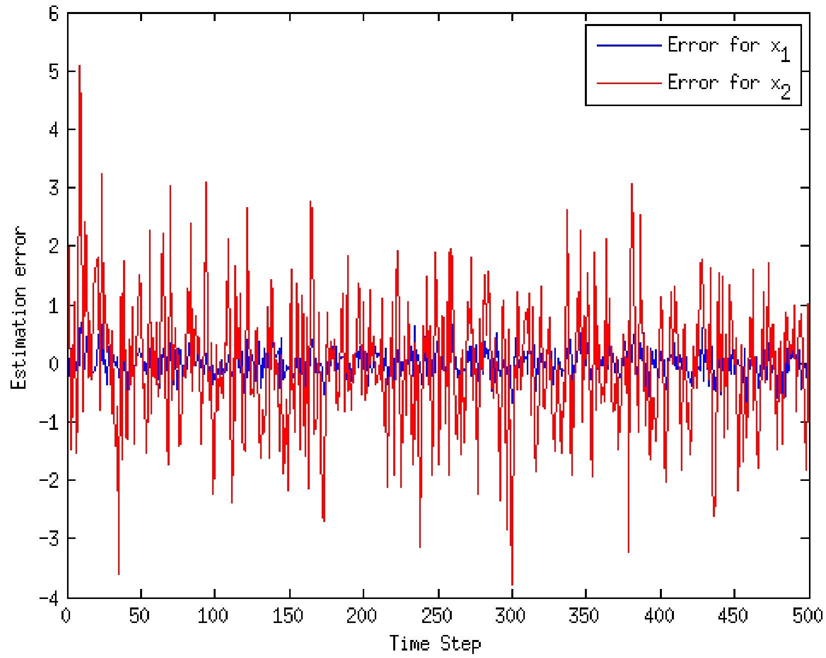

Figure 2: Plot of the error (actual state minus SSKF estimated state) for 3.6.2 Exercise 1

```
%solve DARE for P
P = dare(A',C',B*B',D*D');
K_P = A*P*C'/(C*P*C' + D*D');

%Initialize the sizes for all the arrays we'll use
x = zeros(2,501);
xhat = zeros(2,501);
y = zeros(1,501);

% set initial values. note that index i corresponds to
% time step i-1 because Matlab decided to be special and
% start indices at 1 instead of 0
x(:,1) = [1; 2];
xhat(:,1) = [0; 0];

u = normrnd(0,1,1,500);%get all noise terms
v = normrnd(0,1,1,501);
y(1) = C*x(:,1) + D*v(1);

for i = 2:501
    %update actual state and measurement
    x(:,i) = A*x(:,i-1)+B*u(i-1);
    y(:,i) = C*x(:,i)+D*v(i);

    %SS Kalman filter step
    xhat(:,i) = (A-K_P*C)*xhat(:,i-1)+K_P*y(i-1);
end

H=subplot(2,1,1);
plot(0:500,x(1,:),'b')
hold on
plot(0:500,xhat(1,:),'r--')
xlabel('Time Step')
ylabel('x_1')
legend('x_1','SSKF estimate of x_1')
H2=subplot(2,1,2);
plot(0:500,x(2,:),'b')
hold on
plot(0:500,xhat(2,:),'r--')
xlabel('Time Step')
```

```
ylabel('x_2')
legend('x_2','SSKF estimate of x_2')

figure()
plot(0:500,x(1,:)-xhat(1,:),'b')
hold on
plot(0:500,x(2,:)-xhat(2,:),'r')
xlabel('Time Step')
ylabel('Estimation error')
legend('Error for x_1','Error for x_2')
```

Compared to the Kalman filter, the steady state Kalman filter will have (on average) larger error near the initial times, with performance asymptotically approaching that of the Kalman filter as time advances.