AE353: Additional Notes on Optimal Observer a.k.a Kalman Filtering

(to be treated as an appendix to the presentation)

T. Bretl and M. Ornik

1 Optimal observer formulation

An observer's job is to estimate the state of a system given the input and the output of the system. The system we're interested in may be nonlinear, and there may be uncertainties in our model of the system (e.g., unknown parameters, disturbances, noise, etc.). Due to these uncertainties, a linear model of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

may not adequately describe the system. We therefore must add additional terms to (1) to correct for these uncertainties.

The input $u(t)$ and the output $y(t)$ in the linear system (1) are known. The only unknown variable in (1) is the state $x(t)$. If we are provided with a state trajectory $x(t)$, we can then calculate the amount by which the equations in (1) are violated, i.e.,

$$\begin{aligned} d(t) &= \dot{x}(t) - Ax(t) - Bu(t) \\ n(t) &= y(t) - Cx(t). \end{aligned} \tag{2}$$

We can rewrite these equations as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + d(t) \\ y(t) &= Cx(t) + n(t). \end{aligned} \tag{3}$$

Re-emphasizing what we've already said, $u(t)$ and $y(t)$ are known, and given $x(t)$, we can calculate $d(t)$ and $n(t)$. It is also clear that if we are given $d(t)$, $n(t)$, and the initial condition $x(t_0)$, we can calculate $x(t)$ using (3).

Now, for a given input $u(t)$ and output $y(t)$, we can ask what state trajectory $x(t)$ agrees best with the given input and output. There are many ways to quantify what "best" means. We will define "best" to mean that the errors $d(t)$ and $n(t)$ are as small as possible. In particular, we will define the state trajectory $x(t)$ that agrees best with the given input and output to be the state trajectory that minimizes

$$n(t_1)^T M_o n(t_1) + \int_{t_0}^{t_1} \left(n(t)^T Q_o n(t) + d(t)^T R_o d(t) \right) \, dt. \tag{4}$$

Therefore, to find the optimal estimate of the state trajectory, we must solve the optimal control problem

$$\begin{aligned}
\underset{x_{[t_0,t_1]},d_{[t_0,t_1]},n_{[t_0,t_1]}}{\text{minimize}} \qquad & n(t_1)^T M_o n(t_1) + \int_{t_0}^{t_1} \left(n(t)^T Q_o n(t) + d(t)^T R_o d(t) \right) \, dt \\
\text{subject to} \qquad & \dot{x}(t) = Ax(t) + Bu(t) + d(t) \\
& y(t) = Cx(t) + n(t)
\end{aligned} \tag{5}$$

The interpretation of this problem is as follows. The current time is t_1. You have taken measurements $y(t)$ over the time interval $[t_0, t_1]$. You are looking for noise $n(t)$ and disturbance $d(t)$ over this same time interval and for an estimate $x(t_1)$ of the current state that would best explain these measurements.

The matrices Q_o, R_o, and M_o are parameters that can be used to trade off noise (the difference between the measurements and what you expect them to be) with disturbance (the difference between the time derivative of the state and what you expect it to be). These matrices have to be symmetric, have to be the right size, and also have to satisfy the following conditions in order for the KF problem to have a solution:

$$Q_o \geq 0 \qquad\qquad R_o > 0 \qquad\qquad M_o \geq 0.$$

Just as with the LQR problem, this notation means is that Q_o and M_o are *positive semidefinite* and that R_o is *positive definite* (see wikipedia).

By plugging in the expression for $n(t)$ that appears in the constraint, this optimal control problem can be rewritten as

$$\begin{aligned}
\underset{x(t_1),d_{[t_0,t_1]}}{\text{minimize}} \qquad & (Cx(t_0) - y(t_0))^T M_o (Cx(t_0) - y(t_0)) \\
& \qquad + \int_{t_0}^{t_1} \left((Cx(t) - y(t))^T Q_o (Cx(t) - y(t)) + d(t)^T R_o d(t) \right) dt \\
\text{subject to} \qquad & \dot{x}(t) = Ax(t) + Bu(t) + d(t)
\end{aligned}$$

It is an optimal control problem, just like LQR—if you define

$$\begin{aligned}
f(t, x, d) &= Ax + Bu(t) + d \\
g(t, x, d) &= (Cx - y(t))^T Q_o (Cx - y(t)) + d^T R_o d(t) \\
h(t, x) &= (Cx - y(t))^T M_o (Cx - y(t))
\end{aligned}$$

then you see that this problem has the general form

$$\begin{aligned}
\underset{x(t_1),d_{[t_0,t_1]}}{\text{minimize}} \qquad & h(t_0, x(t_0)) + \int_{t_0}^{t_1} g(t, x(t), d(t)) dt \\
\text{subject to} \qquad & \frac{dx(t)}{dt} = f(t, x(t), d(t)).
\end{aligned}$$

There are four differences between this form and the one we saw when solving the LQR problem:

- The "input" in this problem is u, not d.
- The "current time" is t_1 and not t_0.
- The final state—i.e., the state at the current time—is *not* given. Indeed, the point here is to *choose* a final state $x(t_1)$ that best explains $u(t)$ and $y(t)$.
- The functions f, g, and h vary with time (because they have parameters in them—$u(t)$ and $y(t)$—that are functions of time).

Because of these four differences, the HJB equation for a problem of this form is

$$0 = -\frac{\partial v(t,x)}{\partial t} + \underset{d}{\text{minimum}} \left\{ -\frac{\partial v(t,x)}{\partial x} f(t,x,d) + g(t,x,d) \right\}, \qquad v(t_0, x) = h(t_0, x(t_0)).$$

Note the change in sign of both the first term outside the minimum and the first term inside the minimum—this is because we are effectively solving an optimal control problem in which time flows backward (from the current time t_1 to the initial time t_0, instead of from the current time t_0 to the final time t_1). It is possible to derive this form of the HJB equation in exactly the same way as it was done in the notes on LQR.

2 Solution to the problem

As usual, our first step is to find a function $v(t,x)$ that satisfies the HJB equation. Here is that equation, with the functions f, g, and h filled in:

$$\begin{aligned} 0 = -\frac{\partial v(t,x)}{\partial t} + \underset{d}{\text{minimum}} \Bigg\{ & -\frac{\partial v(t,x)}{\partial x} (Ax + Bu(t) + d) \\ & + (Cx - y(t))^T Q_o (Cx - y(t)) + d(t)^T R_o d(t) \Bigg\} \\ v(t_0, x) = (Cx(t_0) - y(t_0))^T & M_o (Cx(t_0) - y(t_0)). \end{aligned}$$

Expand the boundary condition:

$$\begin{aligned} v(t_0, x) &= (Cx(t_0) - y(t_0))^T M_o (Cx(t_0) - y(t_0)) \\ &= x(t_0)^T C^T M_o C x(t_0) - 2y(t_0)^T M_o C^T x(t_0) + y(t_0)^T M_o y(t_0) \end{aligned}$$

This function has the form

$$v(t,x) = x^T P(t) x + 2o(t)^T x + w(t)$$

for some symmetric matrix $P(t)$ and some other matrices $o(t)$ and $w(t)$ that satisfy the following boundary conditions:

$$P(t_0) = C^T M_o C \qquad o(t_0) = -CM_o y(t_0) \qquad w(t_0) = y(t_0)^T M_o y(t_0).$$

Let's "guess" that this form of v is the solution we are looking for, and see if it satisfies the HJB equation. Before proceeding, we need to compute the partial derivatives of v:

$$\frac{\partial v}{\partial t} = x^T \dot{P} x + 2\dot{o}^T x + \dot{w} \qquad\qquad \frac{\partial v}{\partial x} = 2x^T P + 2o^T$$

Here again—just as for LQR—we are applying matrix calculus. Plug these partial derivatives into HJB and we have

$$\begin{aligned}
0 &= -\left(x^T \dot{P} x + 2\dot{o}^T x + \dot{w}\right) + \underset{d}{\text{minimum}}\Big\{ -\left(2x^T P + 2o^T\right)(Ax + Bu + d) \\
&\qquad\qquad + (Cx - y)^T Q_o (Cx - y) + d^T R_o d \Big\} \\
&= -\left(x^T \dot{P} x + 2\dot{o}^T x + \dot{w}\right) + \underset{d}{\text{minimum}}\Big\{ d^T R_o d - 2(Px + o)^T d \\
&\qquad\qquad - 2(x^T P + o^T)(Ax + Bu) \\
&\qquad\qquad + (Cx - y)^T Q_o (Cx - y) \Big\}
\end{aligned} \tag{6}$$

To evaluate the minimum, we apply the first-derivative test (more matrix calculus!):

$$\begin{aligned}
0 &= \frac{\partial}{\partial d}\left(d^T R_o d - 2(Px + o)^T d - 2(x^T P + o^T)(Ax + Bu) + (Cx - y)^T Q_o (Cx - y)\right) \\
&= 2d^T R_o - 2(Px + o)^T.
\end{aligned}$$

This equation is easily solved:

$$d = R^{-1}(Px + o). \tag{7}$$

Plugging this back into (6), we have

$$\begin{aligned}
0 &= -\left(x^T \dot{P} x + 2\dot{o}^T x + \dot{w}\right) + \underset{d}{\text{minimum}}\Big\{ -\left(2x^T P + 2o^T\right)(Ax + Bu + d) \\
&\qquad\qquad + (Cx - y)^T Q_o (Cx - y) + d^T R_o d \Big\} \\
&= -(x^T \dot{P} x + 2\dot{o}^T x + \dot{w}) - (Px + o)^T R_o^{-1} (Px + o) \\
&\qquad - 2(x^T P + o^T)(Ax + Bu) + (Cx - y)^T Q_o (Cx - y) \\
&= x^T \left(-\dot{P} - P R_o^{-1} P - 2PA + C^T Q_o C\right) x \\
&\qquad + 2x^T \left(-\dot{o} - P R_o^{-1} o - PBu - C^T Q_o y - A^T o\right) \\
&\qquad + \left(-\dot{w} - o^T R_o^{-1} o - 2o^T Bu + y^T Q_o y\right) \\
&= x^T \left(-\dot{P} - P R_o^{-1} P - PA - A^T P + C^T Q_o C\right) x \\
&\qquad + 2x^T \left(-\dot{o} - P R_o^{-1} o - PBu - C^T Q_o y - A^T o\right) \\
&\qquad + \left(-\dot{w} - o^T R_o^{-1} o - 2o^T Bu + y^T Q_o y\right)
\end{aligned}$$

where the last step is because

$$x^T(N+N^T)x = 2x^TNx \text{ for any } N \text{ and } x.$$

In order for this equation to be true for any x, it must be the case that

$$\begin{aligned}
\dot{P} &= -PR_o^{-1}P - PA - A^TP + C^TQ_oC \\
\dot{o} &= -PR_o^{-1}o - PBu - C^TQ_oy - A^To \\
\dot{w} &= -o^TR_o^{-1}o - 2o^TBu + y^TQ_oy.
\end{aligned}$$

In summary, we have found that

$$v(t,x) = x^TP(t)x + 2o(t)^Tx + w(t)$$

solves the HJB equation, where P, o, and w are found by integrating the above ODEs forward in time, starting from

$$P(t_0) = C^TM_oC \qquad o(t_0) = -CM_oy(t_0) \qquad w(t_0) = y(t_0)^TM_oy(t_0).$$

The optimal choice of state estimate at time t is the choice of x that minimizes $v(t,x)$, that is, the solution to

$$\underset{x}{\text{minimize}} \qquad x^TP(t)x + 2o(t)^Tx + w(t).$$

We can find the solution to this problem by application of the first derivative test, with some matrix calculus:

$$\begin{aligned}
0 &= \frac{\partial}{\partial x}\left(x^TPx + 2o^Tx + w\right) \\
&= 2x^TP + 2o^T,
\end{aligned}$$

implying that

$$x = -P^{-1}o.$$

Let's call this solution $\widehat{x}$. Note that we can, equivalently, write

$$0 = P\widehat{x} + o.$$

Suppose we take the time derivative of this expression, plugging in what we found earlier for $\dot{P}$ and $\dot{o}$, as well as plugging in yet another version of this same expression, $o = -P\widehat{x}$:

$$\begin{aligned}
0 &= \dot{P}\widehat{x} + P\dot{\widehat{x}} + \dot{o} \\
&= \left(-PR_o^{-1}P - PA - A^TP + C^TQ_oC\right)\widehat{x} + P\dot{\widehat{x}} - PR_o^{-1}o - PBu - C^TQ_oy - A^To \\
&= -PR_o^{-1}P\widehat{x} - PA\widehat{x} - A^TP\widehat{x} + C^TQ_oC\widehat{x} + P\dot{\widehat{x}} + PR_o^{-1}P\widehat{x} - PBu - C^TQ_oy + A^TP\widehat{x} \\
&= P\dot{\widehat{x}} - PA\widehat{x} - PBu + C^TQ_o(C\widehat{x} - y) \\
&= P\left(\dot{\widehat{x}} - A\widehat{x} - Bu + P^{-1}C^TQ_o(C\widehat{x} - y)\right).
\end{aligned}$$

For this equation to hold for any P, we must have

$$\dot{\widehat{x}} = A\widehat{x} + Bu - P^{-1}C^TQ_o(C\widehat{x} - y).$$

Behold! This is our expression for an optimal observer, if we define

$$L = P^{-1}C^TQ_o.$$

Finally, suppose we take the limit as $t_0 \to -\infty$, so assume an infinite horizon. It is a fact that P tends to a steady-state value, and so L does as well. When this happens, $\dot{P} = 0$, and so the steady-state value of P is the solution to the *algebraic* equation

$$0 = -PR_o^{-1}P - PA - A^TP + C^TQ_oC.$$

It is customary to write these last two equations in a slightly different way. In particular, suppose we pre- and post-multiply both sides of this last equation by P^{-1}, and define

$$P_o = P^{-1}$$

Then, we have

$$L = P_oC^TQ_o$$

and

$$0 = P_oC^TQ_oCP_o - AP_o - P_oA^T - R_o^{-1}.$$

3 Summary

An optimal observer—a deterministic, infinite-horizon, continuous-time Kalman Filter—is given by

$$\dot{\widehat{x}} = A\widehat{x} + Bu - L(C\widehat{x} - y).$$

where

$$L = P_oC^TQ_o$$

and P_o satisfies

$$0 = P_oC^TQ_oCP_o - AP_o - P_oA^T - R_o^{-1}.$$

4 Comparison between LQR and KF (i.e., why you can use "lqr" in MATLAB to compute an optimal observer)

An optimal controller is given by

$$u = -Kx$$

where

$$K = R_c^{-1}B^TP_c \tag{8}$$

and P_c satisfies

$$0 = P_cBR_c^{-1}B^TP_c - P_cA - A^TP_c - Q_c. \tag{9}$$

An optimal observer is given by

$$\dot{\widehat{x}} = A\widehat{x} + Bu - L(C\widehat{x} - y).$$

where

$$L = P_o C^T Q_o \tag{10}$$

and P_o satisfies

$$0 = P_o C^T Q_o C P_o - A P_o - P_o A^T - R_o^{-1}. \tag{11}$$

Take the transpose of (10) and—remembering that P_o and Q_o are symmetric—we get

$$L^T = Q_o C P_o. \tag{12}$$

Take the transpose of (11) and—remembering that R_o is also symmetric—we get

$$0 = P_o C^T Q_o C P_o - P_o A^T - A P_o - R_o^{-1}. \tag{13}$$

Compare (8) and (9) with (12) and (13). They are **exactly the same** if we make the following replacements:

- replace K with L^T
- replace A with A^T
- replace B with C^T
- replace Q_c with R_o^{-1}
- replace R_c with Q_o^{-1}

This is the reason why

```
L = lqr(A', C', inv(Ro), inv(Qo))'
```

produces an optimal observer, just like

```
K = lqr(A, B, Qc, Rc)
```

produces an optimal controller. **WWWWWOOOOOWWWWW!!!!!!!**

5 Example - Kalman Filter vs Least Squares Estimation

In this section, we discuss how the Kalman filter compares to the method of estimating the state solely from the noisy sensor outputs using least squares method as discussed in lecture 17. To compare these observers, we consider a simple example of a car traveling on a straight race track. In this case, we are interested in the velocity of the crack along the track v for a given input control force u. The equation of motion can be given as

$$m\dot{v} + bv = u$$

where bv denotes the force acting on the car for a variety of reasons including wind drag. Considering our state to be v, and since the equation is already a first order ODE, the state space formulation $\dot{v} = Av + Bu$ can be given as:

$$A = -b/m; \quad B = 1/m.$$

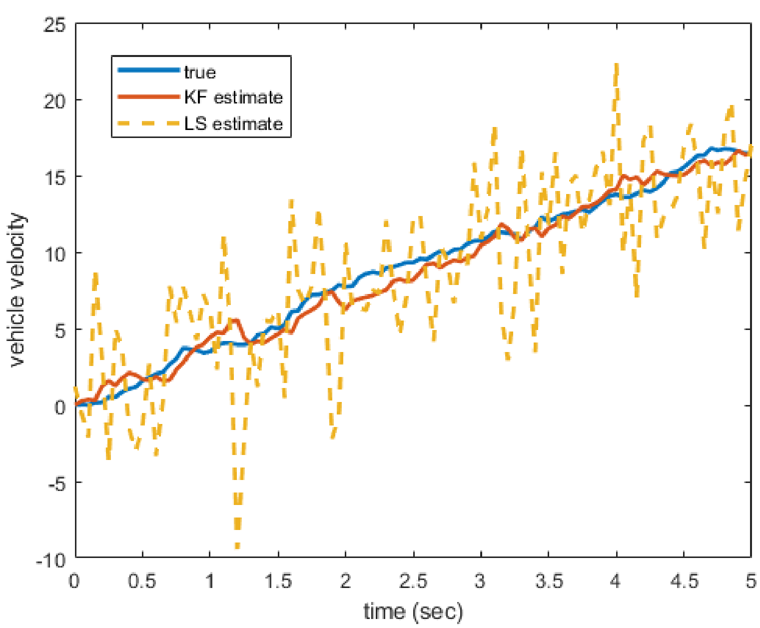
We assume that the model is imperfect and can be represented as

$$\dot{v} = Av + Bu + d$$

where d is some unknown disturbance. Because of this unknown disturbance, we cannot directly calculate the state at any time instant (which we need for state feedback $u = -Kx$) by solving the linear ODE. We instead assume that we can get imprecise measurements of the vehicle velocity from three different sensors. In that case, output is made of three measurements of velocity $y = [m_1; m_2, ; m_3]$ where $m_i = v + n_i$ for some unknown noise n_i, and hence $C = [1; 1; 1]$. The overall dynamics hence can be written as:

$$\dot{v} = Av + Bu + d$$
$$y = Cv + n$$

In the case of least squares estimate as discussed in lecture 17, we only use the three noisy measurements at a given time instant to calculate the estimate of current velocity. Let us use equal weights for each measurement, i.e., $w = [0.33, 0.33, 0.33]$. In the case of Kalman filter (optimal observer) we use both the current measurements and the state estimate and the linear model to estimate the current velocity. The following plot compares the estimates obtained using these two methods with the true velocity:



Clearly, the optimal observer performs significantly better than the least squares which only uses current measurements obtained from sensors. The calculations and the above plot was created using the following code:

```
% state space formulation and system parameters
m = 3000; b = 10;
A = -b/m;
B = 1/m;
C = [1; 1; 1];
D = 0;

% simulate and plot true data with fixed control input
u = 10000;
dt = 0.05;
t_init = 0; num_steps = 100;

% calculate true velocity and generate noisy sensor measurements
v(1) = 0;
y(1,:) = [10*randn, 6*randn, 8*randn];
t(1) = t_init;
for i = 1:num_steps
    v(i+1) = v(i) + dt*(-v(i)*b/m + u/m + 5*randn);
    y(i+1,:) = [10*randn + v(i), 6*randn + v(i), 8*randn + v(i)];
    t(i+1) = t(i) + dt;
end

% Design optimal observer
Qo = 1; Ro = 1;
L = lqr(A', C', inv(Ro), inv(Qo))';

% calculate least-squares estimate
ls_weights = [0.33, 0.33, 0.33];
for i = 1:num_steps+1
ls_est(i) = ls_weights*y(i,:)';
end

% calculate Kalman filter estimate
kf_est(1) = 0;
for i = 1:num_steps
 kf_est(i+1) = kf_est(i) + dt*(A*kf_est(i) + B*u -L*(C*kf_est(i) - y(i,:)
    '));
end

% plot true state and the two estimates
plot(t, v, t, kf_est, t, ls_est, '--', 'LineWidth',2);
legend('true velocity', 'KF estimate', 'LS estimate');
xlabel('time (sec)');
ylabel('vehicle velocity');
```