Optimization-Based Control

Richard M. Murray
Control and Dynamical Systems
California Institute of Technology

DRAFT v1.7b, 23 February 2008

Chapter 5
Kalman Filtering

In this chapter we derive the Kalman filter in continuous time (also referred to as the Kalman-Bucy filter).

Prerequisites. Readers should have basic familiarity with continuous-time stochastic systems at the level presented in Chapter 4.

5.1 Linear Quadratic Estimators

Consider a stochastic system

$$\begin{aligned}\dot{X} &= AX + Bu + FV\\ Y &= CX + W,\end{aligned}$$

where X represents that state, u is the (deterministic) input, V represents disturbances that affect the dynamics of the system and W represents measurement noise. Assume that the disturbance V and noise W are zero-mean, Gaussian white noise (but not necessarily stationary):

$$\begin{aligned} p(v) &= \frac{1}{\sqrt[n]{2\pi}\sqrt{\det R_V}} e^{-\frac{1}{2}v^T R_V^{-1} v} \qquad && E\{V(s)V^T(t)\} = R_V(t)\delta(t-s)\\ p(w) &= \frac{1}{\sqrt[n]{2\pi}\sqrt{\det R_W}} e^{-\frac{1}{2}v^T R_W^{-1} v} \qquad && E\{W(s)W^T(t)\} = R_W(t)\delta(t-s)\end{aligned}$$

We also assume that the cross correlation between V and W is zero, so that the distrubances are not correlated with the noise. Note that we use multivariable Gaussians here, with (incremental) covariance matrix $R_V \in \mathbb{R}^{m\times m}$ and $R_W \in \mathbb{R}^{p\times p}$. In the scalar case, $R_V = \sigma_V^2$ and $R_W = \sigma_W^2$.

We formulate the optimal estimation problem as finding the estimate $\hat{X}(t)$ that minimizes the mean square error $E\{(X(t)-\hat{X}(t))(X(t)-\hat{X}(t))^T\}$ given $\{Y(\tau) : 0 \leq \tau \leq t\}$. It can be shown that this is equivalent to finding the expected value of X subject to the "constraint" given by all of the previous measurements, so that $\hat{X}(t) = E\{X(t)|Y(\tau), \tau \leq t\}$. This was the way that Kalman originally formulated the problem and it can be viewed as solving a *least squares* problem: given all previous $Y(t)$, find the estimate $\hat{X}$ that satisfies the dynamics and minimizes the square error with the measured data. We omit the proof since we will work directly with the error formulation.

Theorem 5.1 (Kalman-Bucy, 1961)**.** *The optimal estimator has the form of a linear observer*

$$\dot{\hat{X}} = A\hat{X} + BU + L(Y - C\hat{X})$$

where $L(t) = P(t)C^T R_W^{-1}$ *and* $P(t) = E\{(X(t) - \hat{X}(t))(X(t) - \hat{X}(t))^T\}$ *and satisfies*

$$\begin{aligned}\dot{P} &= AP + PA^T - PC^T R_W^{-1}(t)CP + FR_V(t)F^T \\ P(0) &= E\{X(0)X^T(0)\}\end{aligned}$$

Sketch of proof. The error dynamics are given by

$$\dot{E} = (A - LC)E + \xi, \qquad \xi = FV - LW, \quad R_\xi = FR_V F^T + LR_W L^T$$

The covariance matrix $P_E = P$ for this process satisfies

$$\begin{aligned}\dot{P} &= (A - LC)P + P(A - LC)^T + FR_V F^T + LR_W L^T \\ &= AP + PA^T + FR_V F^T - LCP - PC^T L^T + LR_W L^T \\ &= AP + PA^T + FR_V F^T + (LR_W - PC^T)R_W^{-1}(LR_W + PC^T)^T \\ &\qquad - PC^T R_W CP,\end{aligned}$$

where the last line follows by completing the square. We need to find L such that $P(t)$ is as small as possible, which can be done by choosing L so that $\dot{P}$ decreases by the maximimum amount possible at each instant in time. This is accomplished by setting

$$LR_W = PC^T \qquad \Longrightarrow \qquad L = PC^T R_W^{-1}.$$

□

Note that the Kalman filter has the form of a *recursive* filter: given $P(t) = E\{E(t)E^T(t)\}$ at time t, can compute how the estimate and covariance *change*. Thus we don't need to keep track of old values of the output. Furthermore, the Kalman filter gives the estimate $\hat{X}(t)$ *and* the covariance $P_E(t)$, so you can see how well the error is converging.

If the noise is stationary (R_V, R_W constant) and *if* $\dot{P}$ is stable, then the observer gain converges to a constant and satisfies the *algebraic Riccati equation*:

$$L = PC^T R_W^{-1} \qquad AP + PA^T - PC^T R_W^{-1} CP + FR_V F^T.$$

This is the most commonly used form of the controller since it gives an explicit formula for the estimator gains that minimize the error covariance. The gain matrix for this case can solved use the `lqe` command in MATLAB.

Another property of the Kalman filter is that it extracts the maximum possible information about output data. To see this, consider the *residual random prossess*

$$R = Y - C\hat{X}$$

(this process is also called the *innovations process*). It can be shown for the Kalman filter that the correlation matrix of R is given by

$$R_R(t, s) = W(t)\delta(t - s).$$

This implies that the residuals are a white noise process and so the output error has *no* remaining dynamic information content.

5.2 Extensions of the Kalman Filter

Correlated disturbances and noise

The derivation of the Kalman filter assumes that the disturbances and noise are independent and white. Removing the assumption of independence is straightforward and simply results in a cross term ($E\{V(t)W(s)\} = R_{VW}\delta(s - t)$) being carried through all calculations.

To remove the assumption of white noise sources, we construct a filter that takes white noise as an input and produces a noise source with the appropriate correlation function (or equivalently, spectral power density function). The intuition behind this approach is that we must have an internal model of the noise and/or disturbances in order to capture the correlation between different times.

Extended Kalman filters

Consider a *nonlinear* system

$$\begin{aligned} \dot{X} &= f(X, U, V), \qquad && X \in \mathbb{R}^n, u \in \mathbb{R}^p, \\ Y &= CX + W, && Y \in \mathbb{R}^q, \end{aligned}$$

where V and W are Gaussian white noise processes with covariance matrices R_V and R_W. A nonlinear observer for the system can be constructed by using the process

$$\dot{\hat{X}} = f(\hat{X}, U, 0) + L(Y - C\hat{X})$$

If we define the error as $E = X - \hat{X}$, the error dynamics are given by

$$\begin{aligned} \dot{E} &= f(X, U, V) - f(\hat{X}, U, 0) - LC(X - \hat{X}) \\ &= F(E, \hat{X}, U, V) - LCe \end{aligned}$$

where

$$F(E, \hat{X}, U, V) = f(E + \hat{X}, U, V) - f(\hat{X}, U, 0)$$

We can now linearize around *current* estimate $\hat{X}$:

$$\begin{aligned}\hat{E} &= \frac{\partial F}{\partial E}E + \underbrace{F(0,\hat{X},U,0)}_{=0} + \underbrace{\frac{\partial F}{\partial V}V}_{\text{noise}} - \underbrace{LCe}_{\text{observer gain}} + \text{h.o.t} \\ &= \tilde{A}E + \tilde{F}V - LCE,\end{aligned}$$

where

$$\left.\begin{aligned}\tilde{A} &= \left.\frac{\partial F}{\partial e}\right|_{(0,\hat{X},U,0)} &&= \left.\frac{\partial f}{\partial X}\right|_{(\hat{X},U,0)} \\ \tilde{F} &= \left.\frac{\partial F}{\partial V}\right|_{(0,\hat{X},U,0)} &&= \left.\frac{\partial f}{\partial V}\right|_{(\hat{X},U,0)}\end{aligned}\right\} \text{Depend on current estimate } \hat{X}$$

We can now design an observer for the linearized system around the *current* estimate:

$$\begin{aligned}\dot{\hat{X}} &= f(\hat{X},U,0) + L(Y - C\hat{X}), \qquad L = PC^T R_W^{-1} \\ \dot{P} &= (\tilde{A} - LC)P + P(\tilde{A} - LC)^T + \tilde{F}R_V\tilde{F}^T + LR_W L^T \\ P(t_0) &= E\{X(t_0)X^T(t_0)\}\end{aligned}$$

This is called the (Schmidt) *extended Kalman filter* (EKF).

The intuition in the Kalman filter is that we replace the predication portion of the filter with the nonlinear modeling while using the insantaneous linearization to compute the observer gain. Although we lose optimimality, in applications the extended Kalman filter works very well and it is very versatile, as illustrated in the following example.

Example 5.1 Online parameter estimation
Consider a linear system with unknown parameters ξ

$$\begin{aligned}\dot{X} &= A(\xi)X + B(\xi)U + FV \qquad \xi \in \mathbb{R}^p \\ Y &= C(\xi)X + W\end{aligned}$$

We wish to solve the parameter identification problem: given $U(t)$ and $Y(t)$, estimate the value of the parameters ξ.

One approach to this online parameter estimation problem is to treat ξ as an unkown *state* that has zero derivative:

$$\begin{aligned}\dot{X} &= A(\xi)X + B(\xi)U + FV \\ \dot{\xi} &= 0.\end{aligned}$$

We can now write the dynamics in terms of the extended state $Z = (X, \xi)$:

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix}X\\ \xi\end{bmatrix} &= \overbrace{\begin{bmatrix}A(\xi) & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}X\\ \xi\end{bmatrix} + \begin{bmatrix}B(\xi)\\ 0\end{bmatrix}U}^{f(\left[\begin{smallmatrix}X\\ \xi\end{smallmatrix}\right],U,V)} + \begin{bmatrix}F\\ 0\end{bmatrix}V\\ Y &= \underbrace{C(\xi)X + W}_{h(\left[\begin{smallmatrix}X\\ \xi\end{smallmatrix}\right],W)}\end{aligned}$$

This system is nonlinear in the extended state Z, but we can use the extended Kalman filter to *estimate* Z. If this filter converges, then we obtain both an estimate of the original state X and an estimate of the unknown parameter $\xi \in \mathbb{R}^p$.

Remark: need various observability conditions on augmented system in order for this to work. ∇

5.3 LQG Control

Return to the original "H_2" control problem

Figure

$$\begin{aligned}\dot{X} &= AX + BU + FV\\ Y &= CX + W\end{aligned}$$

V, W Gaussian white noise with covariance R_V, R_W

Stochastic control problem: find $C(s)$ to minimize

$$J = E\left\{\int_0^\infty \left[(Y-r)^T R_V (Y-r)^T + U^T R_W U\right]\,dt\right\}$$

Assume for simplicity that the reference input $r = 0$ (otherwise, translate the state accordingly).

Theorem 5.2 (Separation principle)**.** *The optimal controller has the form*

$$\begin{aligned}\dot{\hat{X}} &= A\hat{X} + BU + L(Y - C\hat{X})\\ U &= K(\hat{X} - X_d)\end{aligned}$$

where L is the optimal observer gain ignoring the controller and K is the optimal controller gain ignoring the noise.

This is called the *separation principle* (for H_2 control).

5.4 Application to a Thrust Vectored Aircraft

To illustrate the use of the Kalman filter, we consider the problem of estimating the state for the Caltech ducted fan, described already in Section 3.4.

The following code implements an extended Kalman filter in MATLAB, by constructing a state vector that consists of the actual state, the estimated state and the elements of the covariance matrix $P(t)$:

```
% pvtol.m - nonlinear PVTOL model, with LQR and EKF
% RMM, 5 Feb 06
%
% This function has the dynamics for a nonlinear planar vertical takeoff
% and landing aircraft, with an LQR compensator and EKF estimator.
%
% state(1) x position, in meters
% state(2) y position, in meters
% state(3) theta angle, in radians
% state(4-6) velocities
% state(7-12) estimated states
% state(13-48) covariance matrix (ordered rowise)

function deriv = pvtol(t, state, flags)
global pvtol_K; % LQR gain
global pvtol_L; % LQE gain (temporary)
global pvtol_Rv; % Disturbance covariance
global pvtol_Rw; % Noise covariance
global pvtol_C; % outputs to use
global pvtol_F; % disturbance input

% System parameters
J = 0.0475; % inertia around pitch axis
m = 1.5; % mass of fan
r = 0.25; % distance to flaps
g = 10; % gravitational constant
d = 0.2;   % damping factor (estimated)

% Initialize the derivative so that it is correct size and shape
deriv = zeros(size(state));

% Extract the current state estimate
x = state(1:6);
xhat = state(7:12);

% Get the current output, with noise
y = pvtol_C*x + pvtol_C * ...
  [0.1*sin(2.1*t); 0.1*sin(3.2*t); 0; 0; 0; 0];

% Compute the disturbance forces
fd = [
  0.01*sin(0.1*t); 0.01*cos(0.027*t); 0
];

% Compute the control law
F = -pvtol_K * xhat + [0; m*g];

% A matrix at current estimated state
A = [  0     0     0     1     0     0;
```

```
       0     0     0     0     1     0;
       0     0     0     0     0     1;
       0, 0, (-F(1)*sin(xhat(3)) - F(2)*cos(xhat(3)))/m, -d, 0, 0;
       0, 0, (F(1)*cos(xhat(3)) - F(2)*sin(xhat(3)))/m, 0, -d, 0;
       0     0     0     0     0     0 ];

% Estimator dynamics (prediction)
deriv(7) = xhat(4); deriv(8) = xhat(5); deriv(9) = xhat(6);
deriv(10) = (F(1) * cos(xhat(3)) - F(2) * sin(xhat(3)) - d*xhat(4)) / m;
deriv(11) = (F(1) * sin(xhat(3)) + F(2) * cos(xhat(3)) - m*g - d*xhat(5)) / m;
deriv(12) = (F(1) * r) / J;

% Compute the covariance
P = reshape(state(13:48), 6, 6);
dP = A * P + P * A’ - P * pvtol_C’ * inv(pvtol_Rw) * pvtol_C * P + ...
  pvtol_F * pvtol_Rv * pvtol_F’;
L = P * pvtol_C’ * inv(pvtol_Rw);

% Now compute correction
xcor = L * (y - pvtol_C*xhat);
for i = 1:6, deriv(6+i) = deriv(6+i) + xcor(i); end;

% PVTOL dynamics
deriv(1) = x(4); deriv(2) = x(5); deriv(3) = x(6);
deriv(4) = (F(1)*cos(x(3)) - F(2)*sin(x(3)) - d*x(4) + fd(1)) / m;
deriv(5) = (F(1)*sin(x(3)) + F(2)*cos(x(3)) - m*g - d*x(5) + fd(2)) / m;
deriv(6) = (F(1) * r + fd(3)) / J;

% Copy in the covariance updates
for i = 1:6,
  for j = 1:6,
    deriv(6+6*i+j) = dP(i, j);
  end;
end;

% All done
return;
```

To show how this estimator can be used, consider the problem of stabilizing the system to the origin with an LQR controller that uses the estimated state. The following MATLAB code implements the controller, using the previous simulation:

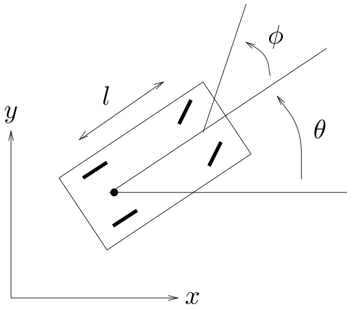
```
% kf_dfan.m - Kalman filter for the ducted fan
% RMM, 5 Feb 06

% Global variables to talk to simulation modle
global pvtol_K pvtol_L pvtol_C pvtol_Rv pvtol_Rw pvtol_F;

%%
%% Ducted fan dynamics
%%
%% These are the dynamics for the ducted fan, written in state space
%% form.
%%

% System parameters
J = 0.0475; % inertia around pitch axis
m = 1.5; % mass of fan
r = 0.25; % distance to flaps
g = 10; % gravitational constant
d = 0.2;   % damping factor (estimated)

% System matrices (entire plant: 2 input, 2 output)
A = [  0     0     0     1     0     0;
       0     0     0     0     1     0;
       0     0     0     0     0     1;
       0     0    -g   -d/m    0     0;
       0     0     0     0   -d/m    0;
       0     0     0     0     0     0 ];

B = [  0     0;
       0     0;
       0     0;
      1/m    0;
       0    1/m;
      r/J    0    ];

C = [  1     0    0      0     0     0;
       0     1    0      0     0     0 ];

D = [  0     0;   0      0];

dfsys = ss(A, B, C, D);

%%
%% State space control design
%%
%% We use an LQR design to choose the state feedback gains
%%

K = lqr(A, B, eye(size(A)), 0.01*eye(size(B'*B)));
pvtol_K = K;

%%
%% Estimator #1
%%

% Set the disturbances and initial condition
pvtol_F = eye(6);
pvtol_Rv = diag([0.0001, 0.0001, 0.0001, 0.01, 0.04, 0.0001]);

x0 = [0.1 0.2 0 0 0 0];
R11 = 0.1; R22 = 0.1; R33 = 0.001;
```

```
% Set the weighting matrices (L is computed but not used)
pvtol_C = [1 0 0 0 0 0; 0 1 0 0 0 0];
pvtol_Rw = diag([R11 R22]);
pvtol_L = lqe(A, pvtol_F, pvtol_C, pvtol_Rv, pvtol_Rw);

[t1, y1] = ode45(@pvtol, [0, 15], ...
  [x0 0*x0 reshape(x0'*x0, 1, 36)]);

subplot(321);
  plot(t1, y1(:,1), 'b-', t1, y1(:,2), 'g--');
  legend x y;
  xlabel('time');
  ylabel('States (no \theta)');
  axis([0 15 -0.3 0.3]);

subplot(323);
  plot(t1, y1(:,7) - y1(:,1), 'b-', ...
       t1, y1(:,8) - y1(:,2), 'g--', ...
       t1, y1(:,9) - y1(:,3), 'r-');
  legend xerr yerr terr;
  xlabel('time');
  ylabel('Errors (no \theta)');
  axis([0 15 -0.2 0.2]);

subplot(325);
  plot(t1, y1(:,13), 'b-', t1, y1(:,19), 'g--', t1, y1(:,25), 'r-');
  legend P11 P22 P33
  xlabel('time');
  ylabel('Covariance (w/ \theta)');
  axis([0 15 -0.2 0.2]);

%%
%% Estimator #2
%%

% Now change the output and see what happens (L computed but not used)
pvtol_C = [1 0 0 0 0 0; 0 1 0 0 0 0; 0 0 1 0 0 0];
pvtol_Rw = diag([R11 R22 R33]);
pvtol_L = lqe(A, pvtol_F, pvtol_C, pvtol_Rv, pvtol_Rw);

[t2, y2] = ode45(@pvtol, [0, 15], ...
  [x0 0*x0 reshape(x0'*x0, 1, 36)]);
subplot(322);
  plot(t2, y2(:,1), 'b-', t2, y2(:,2), 'g--');
  legend x y;
  xlabel('time');
  ylabel('States (w/ \theta)');
  axis([0 15 -0.3 0.3]);

subplot(324);
  plot(t2, y2(:,7) - y2(:,1), 'b-', ...
       t2, y2(:,8) - y2(:,2), 'g--', ...
       t2, y2(:,9) - y2(:,3), 'r-');
  legend xerr yerr terr;
  xlabel('time');
  ylabel('Errors (w/ \theta)');
  axis([0 15 -0.2 0.2]);

subplot(326);
  plot(t2, y2(:,13), 'b-', t2, y2(:,19), 'g--', t2, y2(:,25), 'r-');
  legend P11 P22 P33
  xlabel('time');
  ylabel('Covariance (w/ \theta)');
  axis([0 15 -0.2 0.2]);

print -dpdf dfan_kf.pdf
```

5.5 Further Reading

Exercises

5.1 Consider the problem of estimating the position of an autonomous mobile vehicle using a GPS receiver and an IMU (inertial measurement unit). The dynamics of the vehicle are given by



$$\begin{aligned}\dot{x} &= \cos\theta\, v\\ \dot{y} &= \sin\theta\, v\\ \dot{\theta} &= \frac{1}{\ell}\tan\phi\, v,\end{aligned}$$

We assume that the vehicle is disturbance free, but that we have noisy measurements from the GPS receiver and IMU and an initial condition error.

In this problem we will utilize the full form of the Kalman filter (including the $\dot{P}$ equation).

(a) Suppose first that we only have the GPS measurements for the xy position of the vehicle. These measurements give the position of the vehicle with approximately 1 meter accuracy. Model the GPS error as Gaussian white noise with $\sigma = 1.2$ meter in each direction and design an optimal estimator for the system. Plot the estimated states and the covariances for each state starting with an initial condition of 5 degree heading error at 10 meters/sec forward speed (i.e., choose $x(0) = (0, 0, 5\pi/180)$ and $\hat{x} = (0, 0, 0)$).

(b) An IMU can be used to measure angular rates and linear acceleration. Assume that we use a Northrop Grumman LN200 to measure the angular rate $\dot{\theta}$. Use the datasheet on the course web page to determine a model for the noise process and design a Kalman filter that fuses the GPS and IMU to determine the position of the vehicle. Plot the estimated states and the covariances for each state starting with an initial condition of 5 degree heading error at 10 meters/sec forward speed.

Note: be careful with units on this problem!