MATH4406 (Control Theory)
Part 10: Kalman Filtering and LQG

Prepared by Yoni Nazarathy, Last Updated: October 19, 2012

1 About

We have spent plenty of time in the course dealing with systems of the form:

$$\begin{array}{rcl} \dot{x}(t) &=& Ax(t)+Bu(t) \\ y(t) &=& Cx(t)+Du(t) \end{array} \quad \text{and} \quad \begin{array}{rcl} x(n+1) &=& Ax(n)+Bu(n) \\ y(n) &=& Cx(n)+Du(n) \end{array} . \tag{1}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. The focus was mostly on the continuous time version $\big(u(t), x(t), y(t)\big)$. In unit 4 we saw how to design a state feedback controller and an observer and in later units we dealt with optimal control of such systems.

We now augment our system models by adding *noise* components. To the first equation we shall add *disturbance noise* (ξ_x) and to the second equation we shall add *measurement noise* (ξ_y). This yields:

$$\begin{array}{rcl} \dot{x}(t) &=& Ax(t)+Bu(t)+\xi_x(t) \\ y(t) &=& Cx(t)+Du(t)+\xi_y(t) \end{array} \quad \text{or} \quad \begin{array}{rcl} x(n+1) &=& Ax(n)+Bu(n)+\xi_x(n) \\ y(n) &=& Cx(n)+Du(n)+\xi_y(n) \end{array} .$$

One way of modeling the noise is by assuming that $\xi(\cdot)$ is from some function class and assuming that in controlling the system we have no knowledge of what specific $\xi(\cdot)$ from that class is used. This is the method of *robust control*. Alternatively, we can think of $\xi(\cdot)$ as a *random process(es)* by associating a probability space with the model. We shall focus on the latter approach.

The level of mathematical care that is needed for handling the continuous time case is beyond our scope as it requires some basic understanding of *stochastic calculus* (e.g. Brownian motion, Stochastic Differential Equations, Ito's formula etc...). We shall thus focus on the discrete time case which is simpler in that the random processes (es) are discrete time sequences of *random variables*. Luckily, the methods that we shall survey (Kalman filtering and Linear Quadratic Gaussian (LQG) optimal control) are often applied in practice in discrete time on digital computers. So understanding the discrete time case is both pedagogically simpler and often of greater practical interest.

In treating $\xi_x(n)$ and $\xi_y(n)$ as discrete time random processes we shall assume they are each i.i.d. (independent and identically distributed) sequences of zero-mean Gaussian (normal) random vectors with covariance matrices Σ_x and Σ_y respectively (we review this below). In many physical situations this is actually a practical assumption:

- Having the noise of one time slot independent of the disturbance at other time slots is the practical situation (especially for short lived disturbances). (This is the first 'i' of i.i.d.).
- Having noise of a constant statistical law makes sense for time invariant systems. (This is the second 'i' of i.i.d.).
- Having noise that have a mean of 0 implies there is no general direction of the disturbance.
- Having noise that follows the Gaussian distribution is sensible if the noise is a summation of many small factors. In this case the *central limit theorem* implies that the noise distribution is Gaussian.

Note 1: We are not restricting individual coordinates of $\xi(n)$ (at any time n) to be independent.

Note 2: Note that even though the noise terms are i.i.d., $x(\cdot)$ is no longer an i.i.d. process (it will be in the pathological case in which $A = 0$ and $B = 0$).

Note 3: In many situations the variance (covariance matrix) of ξ can be modeled from "first principles" just as the (A, B, C, D) model is. This is the case of noise is due to well understood electromagnetic effects as well as due to rounding errors appearing in digital control.

What will we do with the stochastic model?

1. **State estimation (Kalman filtering):** For the deterministic system, we saw the Luenberger observer:

   $$\hat{x}(n+1) = A\hat{x}(n) + Bu(n) + K\big(y(n) - \hat{y}(n)\big).$$

   The Kalman filter is used to do essentially the same thing, yet now taking into control the fact that now $x(\cdot)$ is a random process.

2. **Optimal control (LQG):** For the deterministic system we saw how to design a state feedback control such that,

   $$\sum_{k=0}^{\infty} x'(k)Qx(k) + u'(k)Ru(k),$$

   is minimized (if $Q \geq 0$ and $R > 0$). Now with random noise, $x(\cdot)$ is a random process. Further if we use a state feedback control then $u(\cdot)$ is random process. We are thus interested in finding a control law that minimizes,

   $$\mathbb{E}\left[\sum_{k=0}^{\infty} x'(k)Qx(k) + u'(k)Ru(k)\right]$$

We will have time to touch LQG only briefly and focus mostly on the Kalman filter.

A practical note: The celebrated Kalman filter is implemented in a variety of engineering applications dealing with tracking, positioning and sensing. It is a good thing to know about outside the scope of control also.

2 Gaussian Random Vectors

We briefly review/summarize Gaussian random vectors. We begin with Gaussian scalars: A random variable, X is said to have a Gaussian (normal) distribution with a mean of μ and a variance of $\sigma^2 > 0$, denoted, $X \sim N(\mu, \sigma^2)$ if,

$$\mathbb{P}\Big(a \leq X \leq b\Big) = \int_a^b \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} dx.$$

We have,

$$\mathbb{E}\left[X\right] = \int_{-\infty}^{\infty} x \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} dx = \mu.$$

Further,

$$Var(X) = \int_{-\infty}^{\infty} (x-\mu)^2 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} dx = \sigma^2.$$

We now consider the random vector $X = [X_1, \ldots, X_n]'$. Assume momentarily that each of the random variables X_i follows some arbitrary distribution. The expectation of X is the vector,

$$\mathbb{E}\left[X\right] = \begin{bmatrix} \mathbb{E}\left[X_1\right] \\ \vdots \\ \mathbb{E}\left[X_n\right] \end{bmatrix}.$$

Similarly, if X is a matrix of random variables than $\mathbb{E}\left[X\right]$ is the matrix of the expectations of the individual entries.

This leads to the important (for our context) definition of the *covariance matrix* of a random vector X.

$$Cov(X) = \mathbb{E}\left[\big(X - \mathbb{E}\left[X\right]\big)\big(X - \mathbb{E}\left[X\right]\big)'\right].$$

Note that $Cov(X)$ is an $n \times n$ symmetric matrix with individual elements:

$$\Big(Cov(X)\Big)_{i,j} = Cov(X_i, X_j).$$

Reminder: For two scalar random variables Z and W,

$$Cov(Z, W) = \mathbb{E}\left[\big(Z - \mathbb{E}\left[Z\right]\big)\big(W - \mathbb{E}\left[W\right]\big)\right] = \mathbb{E}\left[ZW\right] - \mathbb{E}\left[Z\right]\mathbb{E}\left[W\right].$$

Notes: (i) If $Z = W$ then $Cov(Z, W) = Var(Z)$. (ii) If one (or both) of the random variables are zero mean then the covariance is simply $\mathbb{E}\,[ZW]$. (iii) If the random variables are independent then the covariance is 0.

The covariance matrix thus records the variance of the random variables on the diagonal and the covariances on the off-diagonal entries.

Exercise 2.1 *Assume you are given an n dimensional random variable X and an $m \times n$ matrix A. Define $Y = AX$. What is the mean vector of Y in terms of that of X? What is the covariance matrix of Y in terms of the covariance matrix of X?*

We are now in a position to define (one of several equivalent definitions) Gaussian random vectors: We say the random vector X is Gaussian with mean vector μ and covariance matrix Σ, denoted $X \sim N(\mu, \Sigma)$ if,

$$\mathbb{P}\Big(a_1 \leq X_1 \leq b_1, \ldots, a_n \leq X_n \leq b_n\Big) = \int_{a_1}^{b_1} \ldots \int_{a_n}^{b_n} \phi(x_1, \ldots, x_n) dx_1 \ldots dx_n,$$

with the density function being,

$$\phi(x_1, \ldots, x_n) = \frac{1}{(2\pi)^{n/2} det(\Sigma)^{1/2}} e^{-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)}.$$

It can be calculated that in this case,

$$\mathbb{E}\,[X] = \mu, \qquad Cov(X) = \Sigma.$$

Further, the marginal distribution of each of the X_i's is normal.

Notes: (i) Distributions of Gaussian random vectors are characterized by their mean vector and covariance matrix. (ii) If two coordinates are non-correlated (covariance 0) then they are independent. (iii) Linear transformations of Gaussian random vectors yield random vectors that still follow the Gaussian distribution with mean and covariance as given by Exercise 2.1.

The final property that we shall overview for Gaussian random vectors deals with *conditional distributions*. Partition $X \sim \mathcal{N}(\mu, \Sigma)$ into X_a and X_b and have,

$$\mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}. \qquad \Sigma = \begin{bmatrix} \Sigma_a & \Sigma_{ab} \\ \Sigma'_{ab} & \Sigma_b \end{bmatrix}.$$

We have that the distribution of X_a conditional on $X_b = x_b$ is

$$\mathcal{N}\Big(\mu_a + \Sigma_{ab}\Sigma_b^{-1}(x_b - \mu_b), \quad \Sigma_a - \Sigma_{ab}\Sigma_b^{-1}\Sigma'_{ab}\Big). \tag{2}$$

This is useful for estimating X_a based on *measurements* of X_b. A sensible estimate in this case is, $\mu_a + \Sigma_{ab}\Sigma_b^{-1}(x_b - \mu_b)$. As a "sanity check" of this formula observe that if X_a and X_b are independent then $\Sigma_{ab} = 0$ and thus the estimate is simply μ_a.

3 Minimum Mean Square Estimation

Consider now the general situation in which you observe the value of a random vector $X_b = x_b$ and would like to use it to estimate the value of X_a. Here we model X_a and X_b as two random vectors (measurable functions) on the same probability space and hope that they are somewhat dependent (i.e. knowledge of X_b can give us some information on X_a). We are thus looking for a function $f(\cdot)$ such that $f(X_b)$ is a "good" estimate on X_a. There are all kinds of definitions of "good" – here is perhaps the most popular one:

$$\min_h \mathbb{E}\left[||X_a - h(X_b)||^2\right], \tag{3}$$

where $||\cdot||$ is the Euclidean norm and the minimization is over all $h(\cdot)$ in some function class whose definition we leave vague for the purpose of this informal discussion. Note that the expectation is with respect to both X_a and X_b. Does this criterion make sense? Yes, of course! Further, it turns out to be very tractable in certain cases since it turns out that the $h(\cdot)$ that minimizes (3) is:

$$h^*(x_b) = \mathbb{E}\left[X_a \mid X_b = x_b\right]. \tag{4}$$

The above is read as the "conditional expectation of the random vector X_a, given the observed value x_b". Does the best estimator $h^*(\cdot)$ make sense? Yes of course!

Brief reminder: If two random vectors X_a and X_b are distributed say with a density $f_{ab}(x_a, x_b)$, then the conditional density of X_a given $X_b = x_b$ is:

$$f_{a|b}(x_a|x_b) = \frac{f_{ab}(x_a, x_b)}{f_b(x_b)},$$

where the denominator is the marginal density of X_b, namely (assuming X_a is k-dimensional):

$$f_b(x_b) = \int_{x_a \in \mathbb{R}^k} f_{ab}(x_a, x_b) dx_a.$$

I.e. to get the marginal density of X_b you need to "integrate out" all of the values that X_a may get. And to get the conditional distribution of X_a given the information that X_b takes a specific values x_b, you need to "rescale" the joint density by the marginal of X_b. Try to draw this in two dimensions.

Now the conditional expectation (for a given value of X_b) that appears in (4) is simply evaluated as follows:

$$\mathbb{E}\left[X_a \mid X_b = x_b\right] = \int_{x_a \in \mathbb{R}^k} x_a \; f_{a|b}(x_a|x_b) dx_a.$$

Further note that the expression $\mathbb{E}\left[X_a \mid X_b\right]$ (where we do not specify a given values for X_b) is actually a random variable that is a function of the random variable X_b, where the function is:

$$g(X_b) = \int_{x_a \in \mathbb{R}^k} x_a \; f_{a|b}(x_a|X_b) dx_a.$$

Hence the conditional expectation $\mathbb{E}\left[X_a \mid X_b\right]$ is actually a random variable in itself. And we may thus attempt to take its expectation. It turns out that in this case:

$$\mathbb{E}\left[g(X_b)\right] = \mathbb{E}\left[\mathbb{E}\left[X_a \mid X_b\right]\right] = \mathbb{E}\left[X_a\right]. \tag{5}$$

Note: The above "brief reminder" about conditional expecation is very informal as technical details are missing. Yet this is enough for our needs.

Here is now (an informal) proof of (4):

Proof First use the conditional expectation formula similar to (5):

$$\mathbb{E}\left[||X_a - h(X_b)||^2\right] = \mathbb{E}\left[\mathbb{E}\left[||X_a - h(X_b)||^2 \;\mid\; X_b\right]\right] = \int \mathbb{E}\left[||X_a - h(X_b)||^2 \;\mid\; X_b(\omega)\right] dP_{X_b}(\omega). \tag{6}$$

The last expression represents the outer expectation as a Lebesgue integral with respect to the probability measure associated with the random variable X_b. This is not needed to understand the proof, but is here for additional clarity on the meaning of expectation.

Note that the internal conditional expectation (conditional on X_b) is a function, $\tilde{g}(\cdot)$ of the random variable X_b. Let's investigate this function in a bit greater detail. Assume that the estimator $h(X_b)$ takes on the value z (i.e. assume that in the probability sample space associated with the random variable X_b, we get and ω such that $h(X_b(\omega)) = z$). Then,

$$\mathbb{E}\left[||X_a - z||^2 \;\mid\; X_b\right] = \mathbb{E}\left[||X_a||^2 \;\mid\; X_b\right] - 2z'\mathbb{E}\left[X_a \;\mid\; X_b\right] + ||z||^2$$

Taking derivative with respect to z (note that z is generally a vector) and equating to 0 implies that the above is minimized by $z = \mathbb{E}\left[X_a|X_b\right]$. I.e. the integrand in (6) is minimized by setting,

$$h(X_b) = \mathbb{E}\left[X_a|X_b\right].$$

Thus the integral (the outer expectation) is also minimized by this choice of $h(\cdot)$ and thus the (3) is minimized by (4). □

Evaluating (4) for arbitrarily distributed X_a and X_b can be a complicated (not explicitly solvable) task. Yet for Gaussian random vectors we are blessed with a clean result. Indeed as we saw in the case of Gaussian random vectors that this conditional expectation has the closed (linear) form. So if you believe (4), in the case of Gaussian random vectors,

$$h^*(x_b) = \mu_a + \Sigma_{ab}\Sigma_b^{-1}(x_b - \mu_b).$$

We thus see that for Gaussian random vectors, the optimal estimator $h^*(\cdot)$ is an linear (affine to be precise) function of x_b. It is thus tempting to restrict the function class of $h^*(\cdot)$ in (3) to,

$$h(x_b) = Gx_b + g,$$

where G and g are a matrix and a vector of the appropriate dimension. The pair (G, g) that minimizes (3) is sometimes called the LMMSE estimator (Linear Minimum Mean Square Error estimator).

Exercise 3.1 *What are G and g in the case of Gaussian random variables?*

Exercise 3.2 *Prove the following proposition by taking derivatives w.r.t. to G and g.*

Proposition 3.3 *Let (X_a, X_b) be random vectors with means μ_a and μ_b respectively and with a covariance matrix (of $(X_a, X_b)'$) being:*

$$\begin{bmatrix} \Sigma_a & \Sigma_{ab} \\ \Sigma'_{ab} & \Sigma_b \end{bmatrix}.$$

Then LMMSE estimator of X_a given $X_b = x_b$ is:

$$h^*(x_b) = \mu_a + \Sigma_{ab}\Sigma_b^{-1}(x_b - \mu_b).$$

Further the covariance matrix of the error vector $X_a - h^*(X_b)$ is given by:

$$\mathbb{E}\left[\big(X_a - h^*(X_b)\big)\big(X_a - h^*(X_b)\big)'\right] = \Sigma_a - \Sigma_{a,b}\Sigma_b^{-1}\Sigma'_{a,b}.$$

In the case of non-Gaussian random variables, restricting to an affine estimator based on G and g is often a compromise:

Exercise 3.4 *Let X_b have a uniform distribution on the interval $[-1, 1]$ and let $X_a = X_b^2$. Find the best affine estimator of X_a in terms of X_b and compare its performance (using the objective (3)) to the best estimator (4).*

Repeat for the case of,

$$f_{a,b}(x_a, x_b) = \begin{cases} 2e^{-(x_a+x_b)} & 0 \le x_b \le x_a < \infty, \\ 0 & \textit{elsewhere.} \end{cases}$$

4 The Kalman Filtering Problem "Solved" by LMMSE

Our goal is to have a state estimate, $\hat{x}(n)$ for a given (A, B, C, D) + noise system:

$$\begin{array}{rcl} x(n+1) & = & Ax(n) + Bu(n) + \xi_x(n) \\ y(n) & = & Cx(n) + Du(n) + \xi_y(n) \end{array}.$$

More specifically we assume we have controlled this system over times $k = 0, \ldots, N-1$ by setting inputs $u(0), \ldots, u(N-1)$ (which we know) and have measured outputs $y(0), \ldots, y(N-1)$. Note that we treat $x(0)$ as a random variable also where we assume we know its mean and covariance.

We will now show that his problem can be posed as estimating X_a based on measurement of X_b (as presented in the previous section) where,

$$X_a = (x(0)', x(1)', \ldots, x(N)')', \qquad X_b = (y(0)', y(1)', \ldots, y(N)')',$$

and the inputs $u(0), \ldots, u(N-1)$ are known values.

By iterating the system, we get:

$$\begin{array}{rcl} x(1) & = & Ax(0) + Bu(0) + \xi_x(0), \\ x(2) & = & A^2x(0) + ABu(0) + Bu(1) + A\xi_x(0) + \xi_x(1), \\ x(3) & = & A^3x(0) + A^2Bu(0) + ABu(1) + Bu(2) + A^2\xi_x(0) + A\xi_x(1) + \xi_x(2), \\ & \vdots & \\ x(N) & = & A^Nx(0) + \sum_{k=0}^{N-1} A^{N-1-k}Bu(k) + \sum_{k=0}^{N-1} A^{N-1-k}\xi_x(k). \end{array}$$

Plugging the above in the output equations, we get,

$$\begin{array}{rcl} y(0) & = & Cx(0) + Du(0) + \xi_y(0), \\ y(1) & = & CAx(0) + CBu(0) + C\xi_x(0) + Du(1) + \xi_y(1) \\ y(2) & = & CA^2x(0) + CABu(0) + CBu(1) + CA\xi_x(0) + C\xi_x(1) + Du(2) + \xi_y(2) \\ & \vdots & \\ y(N) & = & CA^Nx(0) + \sum_{k=0}^{N-1}(CA^{N-1-k}B)u(k) + Du(N) + \sum_{k=0}^{N-1} CA^{N-1-k}\xi_x(k) + \xi_y(N) \end{array}$$

It is thus a simple matter to write out constant matrices $\tilde{A}, \tilde{C}$ and well as functions of

the known input, $\tilde{b}(u), \tilde{d}(u)$, such that:

$$X_a = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N) \end{bmatrix} = \tilde{A} \begin{bmatrix} x(0) \\ \xi_x(0) \\ \vdots \\ \xi_x(N-1) \end{bmatrix} + \tilde{b}\big(u(0), \ldots, u(N-1)\big),$$

$$X_b = \begin{bmatrix} y(0) \\ \vdots \\ y(N) \end{bmatrix} = \tilde{C} \begin{bmatrix} x(0) \\ \xi_x(0) \\ \vdots \\ \xi_x(N-1) \end{bmatrix} + \begin{bmatrix} \xi_y(0) \\ \vdots \\ \xi_y(N) \end{bmatrix} + \tilde{d}\big(u(0), \ldots, u(N)\big)$$

Exercise 4.1 *Specify $\tilde{A}, \tilde{C}$ as well as $\tilde{b}(u), \tilde{d}(u)$ explicitly.*

It is now useful to consider the combined random vector,

$$\zeta = \begin{bmatrix} x(0) \\ \xi_x(1) \\ \vdots \\ \xi_x(N-1) \\ \xi_y(0) \\ \vdots \\ \xi_y(N) \end{bmatrix}.$$

We may now rewrite the equations for X_a and X_b as follows:

$$\begin{bmatrix} X_a \\ X_b \end{bmatrix} = \tilde{F}\zeta + f\big(u(0), \ldots, u(N)\big).$$

Exercise 4.2 *Specify $\tilde{F}$ as well as $\tilde{f}(u)$ explicitly.*

We further have,

$$\Sigma_\zeta := Cov(\zeta) = \begin{bmatrix} \Sigma_{x(0)} & 0 & 0 & 0 \\ 0 & \begin{bmatrix} \Sigma_x & & 0 \\ & \ddots & \\ 0 & & \Sigma_x \end{bmatrix} & \begin{bmatrix} \Sigma_{xy} & & 0 \\ & \ddots & \\ 0 & & \Sigma_{xy} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} \Sigma'_{xy} & & 0 \\ & \ddots & \\ 0 & & \Sigma'_{xy} \end{bmatrix} & \begin{bmatrix} \Sigma_y & & 0 \\ & \ddots & \\ 0 & & \Sigma_y \end{bmatrix} & 0 \\ 0 & 0 & 0 & \Sigma_y \end{bmatrix}.$$

Here the $\Sigma_{x(0)}$ is an assumed covariance matrix for $x(0)$. The other Σ elements are the covariances of the noise vectors: Σ_x is the covariance matrix of the disturbance. Σ_y is the covariance matrix of the measurement noise. And $\Sigma_{x,y}$ is the cross-covariance between disturbance and measurements (this is often assumed 0).

Thus,

$$Cov\Big(\left[\begin{array}{c} X_a \\ X_b \end{array}\right]\Big) = \tilde{F}\Sigma_\zeta \tilde{F}' :=: \left[\begin{array}{cc} \Sigma_a & \Sigma_{ab} \\ \Sigma'_{ab} & \Sigma_b \end{array}\right].$$

Observe also that,

$$\begin{aligned} \mu_a = \mathbb{E}\left[X_a\right] &= [\mathbb{E}\left[x(0)\right]' \;\; 0' \;\; \ldots \;\; 0']' + \tilde{b}\big(u(0), \ldots, u(N-1)\big), \\ \mu_b = \mathbb{E}\left[X_b\right] &= \tilde{d}\big(u(0), \ldots, u(N)\big). \end{aligned}$$

We now have all of the needed ingredients of Proposition 3.3 to calculate the LMMSE of X_a based on X_a. I.e. take,

$$h^*(x_b) = \mu_a + \Sigma_{ab}\Sigma_b^{-1}(x_b - \mu_b).$$

and then the predictor at for the state at time n is:

$$\hat{x}(n) = \Big[h^*(x_b)\Big]_{(nN+1,\ldots,nN+n)}.$$

While this is very nice, it is not efficient from a control theory perspective since getting an estimate for X_a requires computation of the order of $O((nN)^3)$. It would be much better to have some sort of recursive solution that yields $\hat{x}(N)$ at each step. This is the celebrated Kalman filtering algorithm which we present in the next section.

Exercise 4.3 *Consider the scalar system:*

$$\begin{aligned} x(n+1) &= 2x(n) + u(n) + \xi_x(n) \\ y(n) &= x(n) + \xi_y(n) \end{aligned}$$

Where $\xi_x(n)$ and $\xi_y(n)$ are both of unit variance and assumed uncorrelated.

Assume $x(0)$ is such that $\mathbb{E}\left[x(0)\right] = 0$ and $Var\big(x(0)\big) = 0$. Assume a control input of $u(n) = 1$ was applied to the system over the times $n = 0, 1, 2$. And the measured output was, $\big(y(0), y(1), y(2), y(3)\big) = (y_0, y_1, y_2, y_3)$.

Use the derived LMMSE in this section to obtain an estimator for $x(3)$ in terms of (y_0, y_1, y_2, y_3).

5 The Kalman Filtering Algorithm

For simplicity in this section, we assume $B = 0$ and $D = 0$ and thus our system is

$$\begin{aligned} x(n+1) &= Ax(n) + \xi_x(n) \\ y(n) &= Cx(n) + \xi_y(n) \end{aligned}.$$

The more general case (with inputs) easily follows and is left as an exercise. We shall also assume for simplicity that $\Sigma_{xy} = 0$. This assumption can also be relaxed.

In general the Kalman filtering algorithm is based on (deterministic) sequence $K(0), K(1), \ldots$ that is used as follows:

$$\hat{x}(n+1) = A\hat{x}(n) + K(n)\big(y(n+1) - CA\hat{x}(n)\big). \tag{7}$$

In this sense it is like a Luenberger observer yet where the matrices K generally depend on time (even in the case presented here where A and C are constant). As an aid for calculating $K(n)$ we have,

$$S(n) := Cov\big(x(n+1) - \hat{x}(n+1) \;\big|\; x(n), x(n-1), \ldots, x(0)\big),$$

with $S(n)$ following the following recursion:

$$S(n+1) = A\Big(S(n) - S(n)C'\big(CS(n)C' + \Sigma_y\big)^{-1}CS(n)\Big)A' + \Sigma_x.$$

Now $S(n)$ is used to obtain $K(n)$ as follows:

$$K(n) = S(n)C'\big(CS(n)C' + \Sigma_y\big)^{-1}.$$

Note that in many applications we may also use the *steady state Kalman filter* in which we take $S(n)$ as the fixed unique positive definite S solving equation:

$$S = A\Big(S - SC'\big(CSC' + \Sigma_y\big)^{-1}CS\Big)A' + \Sigma_x.$$

This then yields a constant K in (7).

It is obvious that the Kalman filter and (even more) the steady state Kalman filter are computationally efficient compared to the method described in the previous section.

Exercise 5.1 *Consider the scalar system,*

$$\begin{aligned} x(n+1) &= \frac{4}{5}x(n) + \xi_x(n), \\ y(n) &= x(n) + \xi_y(n). \end{aligned}$$

Take, $Var\big(\xi_x(n)\big) = 9/25$ *and* $Var\big(\xi_y(n)\big) = 1$. *Find the form of the predictor* $\hat{x}_n(y)$. *Find the steady state predictor.*

We have the following:

Theorem 5.2 *The sequence defined in* (7) *is the LMMSE estimator of* $x(n)$.

Note that the proof below is based on the the fact the noise terms are Gaussian. In this case the LMMSE is also the optimal MSE estimator. A more general proof based on the *orthogonality principle*, based on the representation of square integrable random vectors as elements of a Hilbert space is also known but is not discussed here. In that case Gaussian assumptions are not required and (7) is still the LMMSE (yet not necessarily the best MSE estimator).

Proof

Denote $Y(n) = \Big(y(0), y(1), \ldots, y(n)\Big)$ and set,

$$\hat{x}^-(n) := \mathbb{E}\left[x(n)|Y(n-1)\right], \qquad \hat{x}(n) := \mathbb{E}\left[x(n)|Y(n)\right].$$

Observe by (4) that $\hat{x}(n)$ is the optimal MSE estimator of $x(n)$ and thus also the LMMSE estimator since $x(\cdot)$ is Gaussian. Denote the respective conditional covariance matrices:

$$\begin{aligned} P(n) &:= \mathbb{E}\left[\big(x(n)-\hat{x}(n)\big)\big(x(n)-\hat{x}(n)\big)' \mid Y(n)\right], \\ P^-(n) &:= \mathbb{E}\left[\big(x(n)-\hat{x}^-(n)\big)\big(x(n)-\hat{x}^-(n)\big)' \mid Y(n-1)\right]. \end{aligned}$$

Further for $n=0$ set, $P^-(0) := \Sigma_{x(0)}$ and $\hat{x}^-(0) := \mathbb{E}\left[x(0)\right]$. Observe that in addition to $y(\cdot)$ and $x(\cdot)$, the sequences $\hat{x}^-(\cdot)$ and $\hat{x}(\cdot)$ are also jointly Gaussian since they are all generated by linear combinations of the "primitives" of the process, $\xi_x(\cdot), \xi_y(\cdot)$ and $x(0)$ and also since $\hat{x}^-(\cdot)$ and $\hat{x}(\cdot)$ follow from the formula for the conditional expecation in (2).

The key step is to observe that when conditioning on $Y(n-1)$, the distribution of $[x(n)', y(n)']'$ is,

$$\mathcal{N}\Big(\begin{bmatrix} \hat{x}^-(n) \\ C\hat{x}^-(n) \end{bmatrix}, \begin{bmatrix} P^-(n) & P^-(n)C' \\ CP^-(n) & CP^-(n)C'+\Sigma_y \end{bmatrix}\Big). \tag{8}$$

Noting that,

$$\hat{x}(n) = \mathbb{E}\left[x(n) \mid Y(n)\right] = \mathbb{E}\left[x(n) \mid y(n), Y(n-1)\right],$$

we apply the mean and covariance formulas of(2) based on (8) with everything preconditioned on $Y(n-1)$ to get:

$$\hat{x}(n) = \hat{x}^-(n) + P^-(n)C'\big(CP^-(n)C'+\Sigma_y\big)^{-1}\big(y(n)-C\hat{x}^-(n)\big), \tag{9}$$

$$P(n) = P^-(n) - P^-(n)C'\big(CP^-(n)C'+\Sigma_y\big)^{-1}CP^-(n). \tag{10}$$

Now observe that,

$$\hat{x}^-(n+1) = \mathbb{E}\left[x(n+1)|Y(n)\right] = \mathbb{E}\left[Ax(n)+\xi_x(n)|Y(n)\right] = A\mathbb{E}\left[x(n)|Y(n)\right] = A\hat{x}(n),$$

and thus substitution in (9) for time $n+1$ yields,

$$\hat{x}(n+1) = A\hat{x}(n) + P^-(n+1)C'\big(CP^-(n+1)C' + \Sigma_y\big)^{-1}\big(y(n+1) - CA\hat{x}(n)\big).$$

Further,

$$P^-(n+1) = Cov\big(x(n+1) \mid Y(n)\big) = Cov\big(Ax(n) + \xi_x(n) \mid Y(n)\big) = AP(n)A' + \Sigma_x.$$

Substitution of (10) in the above yields

$$P^-(n+1) = A\Big(P^-(n) - P^-(n)C'\big(CP^-(n)C' + \Sigma_y\big)^{-1}CP^-(n)\Big)A' + \Sigma_x.$$

Now denote $S(n) := P^-(n+1)$ to obtain the desired equations:

$$\begin{aligned} \hat{x}(n+1) &= A\hat{x}(n) + K(n)\big(y(n+1) - CA\hat{x}(n)\big) \\ K(n) &= S(n)C'\big(CS(n)C' + \Sigma_y\big)^{-1} \\ S(n+1) &= A\Big(S(n) - S(n)C'\big(CS(n)C' + \Sigma_y\big)^{-1}CS(n)\Big)A' + \Sigma_x. \end{aligned}$$

□

Exercise 5.3 *What is the Kalman filter for the case of $B \neq 0$ and $D \neq 0$. Describe any needed changes in the proof above.*

6 Brief Overview of LQG

We only touch LQG briefly and informally. Consider the system,

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) + \xi_x(n) \\ y(n) &= Cx(n) + Du(n) + \xi_y(n), \end{aligned}$$

and assume our goal is to find an optimal output feedback law: $u^*(y)$, such that the following is minimized:

$$\mathbb{E}\,\big[\sum_{k=0}^{N} x(n)'Qx(n) + u(n)'Ru(n)\big],$$

with N either finite or infinite and $Q \geq 0$, $R > 0$. Assume further that (A, B) is controllable and (A, C) is observable.

This generalization of the linear quadratic regulator (LQR) problem studied in previous units, is often refereed to as the LQG problem (Linear quadratic Gaussian). Note

that the LQR formulation that we studied ignored the output y and assumed state-feedback.

It turns out that solution of the LQG problem by means of dynamic programming (yet with a stochastic element) is essentially equivalent to dynamic programming solution of LQR. The basic ingredient is once again *Bellman's principle of optimality*, yet this time presented in a stochastic (Markovian) setting:

In somewhat greater generality, consider systems of the form:

$$x(n+1) = f\Big(x(n), u\big(x(n)\big), \xi(n)\Big), \qquad n = 0, 1, \ldots, N-1,$$

where $f(\cdot)$ is some function and ξ is an i.i.d. sequence. For any prescribed $u(\cdot)$ such a system is a Markov chain (informally a stochastic process whose next step only depends on the current state and some noise component and not on the past). The basic setting of *stochastic dynamic programming* (a.k.a. Markov decision processes) is to find a $u_n^*(x), n = 0, 1, \ldots, N-1$ such that,

$$\mathbb{E}\left[g_N\big(x(N)\big) + \sum_{k=0}^{N-1} g_k\big(x(k), u_k\big(x(k)\big), \xi(k)\big)\right],$$

is minimized. Here $g_k(\cdot), k = 1, \ldots, N-1$ is the cost per stage and $g_N(\cdot)$ is the terminal cost. Note also the slight change of notation, where we put the time index as a subscript of u.

Principle of optimality (stochastic version): Let $u^* = \big(u_0^*(\cdot), \ldots, u_{N-1}^*(\cdot)\big)$ be an optimal policy. Assume that in the stochastic process resulting from $u^*(\cdot)$ it is possible to reach a given state at time n . Consider now the *subproblem* whereby the process is in state $x(n)$ at time n and wish to minimize:

$$\mathbb{E}\left[g_N\big(x(N)\big) + \sum_{k=n}^{N-1} g_k\big(x(k), u_k\big(x(k)\big), \xi(k)\big)\right],$$

then the truncated policy $\big(u_n^*(\cdot), u_{n+1}^*(\cdot), \ldots, u_{N-1}^*(\cdot)\big)$ is optimal for this subproblem. $\square$

By application of the principle of optimality in similar spirit to as is done for the solution of discrete time LQR, we get a solution to the LQG problem that parallels that of the LQR problem, yet takes the noise into account in the following beautiful manner:

1. The Kalman filtering solution yields an estimator of $\hat{x}(\cdot)$.

2. The deterministic LQR solution (assuming known x) is applied to $\hat{x}$.

In view of the brevity of this section, we omit details, yet mention that this is a stochastic manifestation of the *separation principle* presented in Unit 4, where the observer and feedback control law can be designed separately and then combined. Non-linear (deterministic and stochastic) systems usually do not exhibit this clean property – and are a current active area of research.