# MODULE 6

# PREDICTION, APPLICATIONS, AND MORE BASICS

# CHAPTER 6: PREDICTION, APPLICATIONS, AND MORE BASICS ON DISCRETE KALMAN FILTERING

## PREDICTION

- Recall our formulation for the discrete Kalman filter:

→ Process Model:

$$X_{k+1} = \Phi_k X_k + W_k \quad (5.5.1)$$

$$Z_k = H_k X_k + V_k \quad (5.5.2)$$

→ Initialize $\hat{X}_0^-$ and $P_0^-$

→ Iterate:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (5.5.17)$$

$$\hat{X}_k = \hat{X}_k^- + K_k (Z_k - H_k \hat{X}_k^-) \quad (5.5.8) \quad \text{(Filter)}$$

$$P_k = (I - K_k H_k) P_k^- \quad (5.5.22)$$

$$\hat{X}_{k+1}^- = \Phi_k \hat{X}_k \quad (5.5.23)$$

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k \quad (5.5.25) \quad \Big\} \text{(Predict)}$$

$\Rightarrow \hat{X}_{k+1}^-$ is a <u>one-step</u> prediction with error covariance matrix $P_{k+1}^-$.

- An additional set of equations similar to the prediction equations (5.5.23), (5.5.25) can be added to the Kalman filter to perform N-step prediction.

NOTE: The one-step prediction is still needed to iterate the filter.

- N-step prediction equations:

$$\hat{X}(k+N|k) = \phi(k+N,k)\,\hat{x}(k|k) \qquad (6.1.1)$$

$$P(k+N|k) = \phi(k+N,k)\,P(k|k)\,\phi^T(k+N,k) + Q(k+N,k) \qquad (6.1.2)$$

→ $\hat{x}(k+N|k)$ is the estimate of the state vector at time $k+N$ given $k$ observations.

→ $P(k+N|k)$ is the error covariance matrix associated with $\hat{x}(k+N|k)$.

→ $Q(k+N,k) = E[W_{N,k}\,W_{N,k}^T]$

where $W_{N,k}$ is the white noise contribution to the system state vector accumulated over $N$ timesteps, beginning at step $k$.

EX:

$N=3, \quad k=0, \quad k+N=3$

$$X_1 = \phi_0 X_0 + w_0$$

$$X_2 = \phi_1 X_1 + w_1$$

$$= \phi_1[\phi_0 X_0 + w_0] + w_1$$

$$= \phi_1\phi_0 X_0 + \phi_1 w_0 + w_1$$

$$X_3 = \phi_2 X_2 + w_2$$

$$= \phi_2[\phi_1\phi_0 X_0 + \phi_1 w_0 + w_1] + w_2$$

$$= \underbrace{\phi_2\phi_1\phi_0 X_0}_{\phi(3,0)} + \underbrace{\phi_2\phi_1 w_0 + \phi_2 w_1 + w_2}_{w_{3,0}}$$

- There are two main types of prediction problems that can be studied;

TYPE 1: Fixed prediction interval.

- In this case, $N$ is fixed and the Kalman filter iterates on $k$ as before.
- At each step, a new observation $z_k$ is processed to derive predictions $\hat{x}(k+1|k)$ and $\hat{x}(k+N|k)$. The "smoothed" or "filtered" estimate $\hat{x}(k|k)$ is also produced.
- $P(k+N|k)$ indicates the reliability of the prediction $\hat{x}(k+N|k)$.

TYPE 2: Fixed timestep-

- k is fixed. No new observations are processed.
- N is iterated according to $N = 1, 2, 3, \ldots$ to produce predictions with longer and longer prediction intervals.
- The behavior of $P(k+N|k)$ as N advances indicates how rapidly the quality of the predictions degrades as the interval grows.

Example of a "Type 2" Application:

- The timing on the ranging signals transmitted by the GPS satellites is dithered to intentionally introduce an error in civilian receiver applications.
- The degradation is called "SA" for "selective availability".
- The error limits the horizontal accuracy of civilian receivers to about 100m.
- It is possible for a fixed earth station to receive the satellite ranging signals and use its known fixed position to compute a correction.

- The correction can then be transmitted as an auxilliary signal.

- Using the satellite ranging signals along with an auxilliary correction signal to compute a more accurate position estimate is known as "differential GPS".

- Range and range rate corrections are normally transmitted, but only in relatively coarse timesteps.

- Thus, the differential receiver must extrapolate the position between update times when the differential signal is actually received.

- When a differentially corrected position $z_k$ is observed, the receiver must then compute predictions

$$\hat{x}(k+1|k),\ \hat{x}(k+2|k),\ \hat{x}(k+3|k),\ \ldots$$

until the next corrected position can be observed.

- A simplified version of this problem is discussed in example 6.1 on page 244 of the book.

# Alternative Formulation of the Kalman Filter

- If the state vector is not too large, there is an alternative formulation of the Kalman filter that can be used.
- This form has a simpler expression for the kalman gain sequence $K_k$, but involves matrix inversions.
- From before,

$$P_k = (I - K_k H_k) P_k^- \quad (5.5.22)$$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (5.5.17)$$

- Substituting (5.5.17) into (5.5.22), we obtain

$$P_k = \left[ I - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k \right] P_k^-$$

$$= P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- \quad (6.2.3)$$

- Now,

$$P_k \left[ (P_k^-)^{-1} + H_k^T R_k^{-1} H_k \right]$$

$$= \underbrace{\left[ P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- \right]}_{(6.2.3)} \left[ (P_k^-)^{-1} + H_k^T R_k^{-1} H_k \right]$$

$\longrightarrow$

$$\ldots = P_k^- (P_k^-)^{-1} + P_k^- H_k^T R_k^{-1} H_k$$

$$- P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- (P_k^-)^{-1}$$

$$- P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- H_k^T R_k^{-1} H_k$$

$$= I + P_k^- H_k^T \Big[ R_k^{-1} - (H_k P_k^- H_k^T + R_k)^{-1} I$$

$$- (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- H_k^T R_k^{-1} \Big] H_k$$

$$= I - P_k^- H_k^T \Big[ (H_k P_k^- H_k^T + R_k)^{-1} - R_k^{-1} + (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- H_k^T R_k^{-1} \Big] H_k$$

$$= I - P_k^- H_k^T \Big[ (H_k P_k^- H_k^T + R_k)^{-1} \big( I + H_k P_k^- H_k^T R_k^{-1} \big) - R_k^{-1} \Big] H_k$$

$$= I - P_k^- H_k^T \Big[ (H_k P_k^- H_k^T + R_k)^{-1} \big( R_k R_k^{-1} + H_k P_k^- H_k^T R_k^{-1} \big) - R_k^{-1} \Big] H_k$$

$$= I - P_k^- H_k^T \Big[ (H_k P_k^- H_k^T + R_k)^{-1} (H_k P_k^- H_k^T + R_k) R_k^{-1} - R_k^{-1} \Big] H_k$$

$$= I - P_k^- H_k^T \underbrace{\Big[ R_k^{-1} - R_k^{-1} \Big]}_{\text{zero}} H_k$$

$$= \underline{\underline{I}}.$$

$\Longrightarrow$ Thus, if $P_k$, $P_k^-$, and $R_k$ are invertible, then

$$P_k^{-1} = (P_k^-)^{-1} + H_k^T R_k^{-1} H_k \qquad (6.2.4)$$

- Equation (6.2.4) can be inverted to obtain $P_k$ from $P_k^-$.

- Again from before,

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (5.5.17)$$

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

$$= P_k P_k^{-1} P_k^- H_k^T R_k^{-1} R_k (H_k P_k^- H_k^T + R_k)^{-1}$$

$$= P_k P_k^{-1} P_k^- H_k^T R_k^{-1} (H_k P_k^- H_k^T R_k^{-1} + R_k R_k^{-1})^{-1}$$

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

- Plugging in (6.2.4) for $P_k^{-1}$, we obtain

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

$$= P_k [(P_k^-)^{-1} P_k^- + H_k^T R_k^{-1} H_k P_k^-] H_k^T R_k^{-1} (H_k P_k^- H_k^T R_k^{-1} + I)^{-1}$$

$$= P_k [I + H_k^T R_k^{-1} H_k P_k^-] H_k^T R_k^{-1} (H_k P_k^- H_k^T R_k^{-1} + I)^{-1}$$

$$= P_k [(H_k^T R_k^{-1})(H_k^T R_k^{-1})^{-1} + H_k^T R_k^{-1} H_k P_k^-] H_k^T R_k^{-1} (H_k P_k^- H_k^T R_k^{-1} + I)^{-1}$$

$$= P_k H_k^T R_k^{-1} [(H_k^T R_k^{-1})^{-1} + H_k P_k^-] H_k^T R_k^{-1} (H_k P_k^- H_k^T R_k^{-1} + I)^{-1}$$

$$= P_k H_k^T R_k^{-1} [I + H_k P_k^- H_k^T R_k^{-1}] (I + H_k P_k^- H_k^T R_k^{-1})^{-1}$$

$$= P_k H_k^T R_k^{-1} \quad (6.2.5)$$

- Equations (6.2.4) and (6.2.5) give the alternative formulation:

$$P_k^{-1} = (P_k^-)^{-1} + H_k^T R_k^{-1} H_k \qquad (6.2.4) = (6.2.6)$$

$$K_k = P_k H_k^T R_k^{-1} \qquad (6.2.5) = (6.2.7)$$

- With this formulation,
  - → $P_k$ is found first using (6.2.6).
  - → $K_k$ is then found using (6.2.7).
- The iteration for the Kalman filter is now:

1. Initialize $(P_0^-)^{-1}$ and $\hat{x}_0^-$.
2. $k = 0$
3. $P_k^{-1} = (P_k^-)^{-1} + H_k^T R_k^{-1} H_k$ (6.2.6)
4. $K_k = P_k H_k^T R_k^{-1}$ (6.2.7)
5. $\hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-)$ (5.5.8)
6. $\hat{x}_{k+1}^- = \Phi_k \hat{x}_k$ (5.5.23)
7. $P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k$ (5.5.25)
8. $k = k+1$

- This alternative formulation is especially useful if the initial state vector error covariance matrix has very large entries so that, for example,

$$P_0^- = \text{diag}(\infty, \infty, \ldots, \infty)$$

→ With the usual formulation,

$$K_0 = P_0^- H_0^T (H_0 P_0^- H_0^T + R_0)^{-1}$$

is indeterminate in this case.

→ with the alternative formulation, however, we have

$$P_0^{-1} = (P_0^-)^{-1} + H_0^T R_0^{-1} H_0 = H_0^T R_0^{-1} H_0,$$

which can be used to compute $K_0$ using (6.2.7).

- An example of this is given in Example 6.2 on page 248 of the book.

The final result is no surprise; it is exactly the result one would expect from elementary statistics. The main point of this example is this: The alternative algorithm provides a means of starting the Kalman filter with "infinite uncertainty" if the physical situation under consideration so dictates. ■

## 6.3 PROCESSING THE MEASUREMENT VECTOR ONE COMPONENT AT A TIME

We now have two different Kalman filter algorithms as summarized in Figs. 5.8 and 6.3. They are, of course, algebraically equivalent and produce identical estimates (with perfect arithmetic). The choice as to which should be used in a particular application is a matter of computational convenience. Both algorithms involve matrix inverse operations, and these may lead to difficulties. When using the alternative algorithm of Fig. 6.3, there is no reasonable way to avoid two $(n \times n)$ matrix inversions with each recursive cycle. If the dimension of the state vector $n$ is large, this is, at best, awkward computationally. On the other hand, the matrix inverse that appears in the regular algorithm given in Fig. 5.8 is the same order as the measurement vector. Since this is often less than the order of the state vector, it is usually the preferred algorithm. Furthermore, if the measurement errors at time $t_k$ are uncorrelated, the inverse operation can be eliminated entirely by processing the scalar measurements one at a time.* This will now be shown.

We begin with the expression for the updated error covariance, Eq. (6.2.6):

$$\mathbf{P}_k^{-1} = (\mathbf{P}_k^-)^{-1} + \left[\mathbf{H}_k^{aT} \middle| \mathbf{H}_k^{bT} \middle| \cdots\right] \left[\begin{array}{c|c|c} (\mathbf{R}_k^a)^{-1} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & (\mathbf{R}_k^b)^{-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \ddots \end{array}\right] \left[\begin{array}{c} \mathbf{H}_k^a \\ \hline \mathbf{H}_k^b \\ \hline \vdots \end{array}\right] \tag{6.3.1}$$

The second term in Eq. (6.3.1) is intentionally written in partitioned form and $\mathbf{R}_k$ is assumed to be at least block diagonal. Physically, this means that the measurements available at $t_k$ can be grouped together such that the measurement errors among the $a$, $b$, . . . blocks are uncorrelated. This is often the case when redundant measurements come from different instruments. We next expand the partitions of Eq. (6.3.1) to get

* Processing the vector measurements one component at a time is sometimes referred to as sequential processing. The authors, however, have found that there is some confusion in the meaning of the words sequential and recursive. For this reason, processing the measurements sequentially one at a time *at a given point in time* will be referred to here simply as one-at-a-time processing. The term recursive will be reserved to mean the step-by-step evolution of the measurement processing with time.

$$\mathbf{P}_k^{-1} = \underbrace{(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^{aT}(\mathbf{R}_k^a)^{-1}\mathbf{H}_k^a}_{\substack{\mathbf{P}_k^{-1} \text{ after assimilating} \\ \text{block } a \text{ measurements}}} + \underbrace{\mathbf{H}_k^{bT}(\mathbf{R}_k^b)^{-1}\mathbf{H}_k^b}_{\substack{\mathbf{P}_k^{-1} \text{ after assimilating both} \\ \text{block } a \text{ and } b \text{ measurements}}} + \underbrace{\cdots}_{\text{and so forth}} \tag{6.3.2}$$

Note that the sum of the first two terms is just the $\mathbf{P}_k^{-1}$ one would obtain after assimilating the "block $a$" measurement just as if no further measurements were available. The Kalman gain associated with this block of measurements may now be used to update the state estimate accordingly. Now think of making a trivial projection ahead through zero time. The a posteriori $\mathbf{P}$ then becomes the a priori $\mathbf{P}$ for the next step. When this is added to the $b$ term of Eq. (6.3.2), we have the updated $\mathbf{P}_k^{-1}$ after assimilating the second block of data. This can now be repeated until all blocks are processed. The final estimate and associated error is then the same as would be obtained if all the measurements at $t_k$ had been processed simultaneously. Thus, the designer has some flexibility in the design of the system software. The available measurements at any particular time may be processed either in blocks, one block at a time, or all at once, as best suits the situation at hand. One-at-a-time measurement processing is illustrated in the timing diagram of Fig. 6.4. Note that once we have established the validity of one-at-a-time processing, it makes no difference whether we use the "usual" update formula given in Chapter 5 (see Fig. 5.8) or the alternative formula given in Section 6.2. The end results are the same (within the limits of computational arithmetic).

The concept of processing the measurements one block at a time leads to an interesting physical interpretation of $\mathbf{P}$ inverse. With reference to Eq. (6.3.2), think of $(\mathbf{P}_k^-)^{-1}$ as a measure of the information content of the a priori estimate, that is, before the new measurement information is assimilated into the estimate. For simplicity, begin with $(\mathbf{P}_k^-)^{-1} = 0$. This corresponds to infinite uncertainty, or zero information. Then, as each measurement block is processed, we add an amount $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ to the previous information, until finally the total information is the sum indicated by Eq. (6.3.2). The term "add" is appropriate here because $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is always positive definite. For the heuristic reasons just noted, $\mathbf{P}$ inverse is often referred to as the *information matrix*. This concept is developed further in Chapter 9 in the discussion of decentralized filters (Section 9.6).

One-at-a-time processing is also useful from a system organization viewpoint. Often the system must have the flexibility to accommodate a variety of measurement combinations at each update point. By block processing, the system may be programmed to cycle through all possible measurement blocks one at a time, processing those that are available and skipping those that are not. Simultaneous processing requires a somewhat more complicated system organization whereby the system must be able to form appropriate $\mathbf{H}_k$ and $\mathbf{R}_k$ matrices for all possible combinations of measurements, and it must be prepared to do the corresponding matrix operations with various dimensionality.

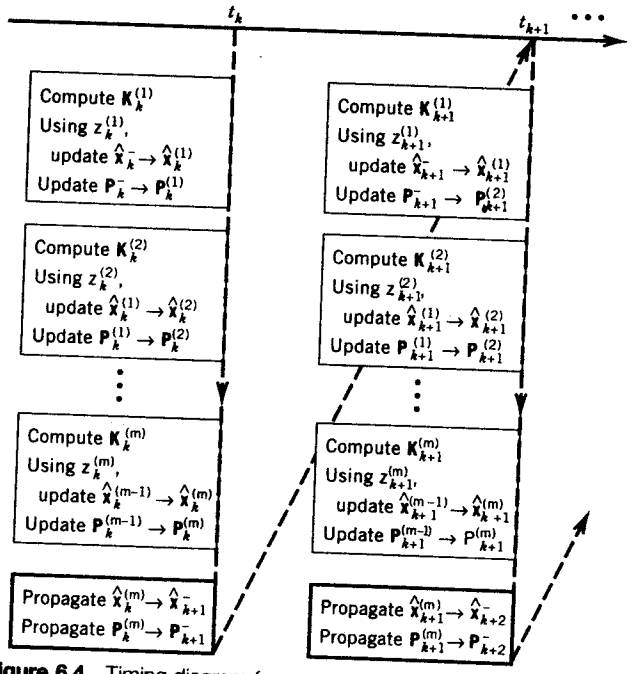


**Figure 6.4** Timing diagram for one-at-a-time measurement processing ($m$ = number of elements in the measurement vector $\mathbf{z}_k$).

There are bound to be some applications where the measurement errors are all mutually correlated. The $\mathbf{R}_k$ matrix is then "full." If this is the case, and one-at-a-time processing is desirable, linear combinations of the measurements may be formed in such a way as to form a new set of measurements whose errors are uncorrelated. One technique for accomplishing this is known as the Gram–Schmidt orthogonalization procedure (4). This procedure is straightforward and an exercise is included to demonstrate its application to the problem of decoupling the measurement errors (see Problem 6.2). Note that the intuitive procedure demonstrated in Problem 6.2 is closely related to Cholesky factorization. This is discussed in detail in Section 5.4.

## 6.4 POWER SYSTEM RELAYING APPLICATION

New applications of Kalman filtering keep appearing regularly, and many of these are now outside the original application area of navigation. One such application has to do with power system relaying (9, 10). When a fault (short) occurs on a three-phase transmission line, it is desirable to sense the problem promptly and take appropriate relaying action to protect the remainder of the system. The hierarchy of decisions that must be made as to which relays should trip (and where) is relatively complicated. It suffices to say here that it is desir-

# Avoiding Divergence Problems with the Kalman Filter

- Under certain circumstances, the recursive Kalman filtering equations can sometimes diverge.
- When this happens, the gains, the P matrix, or both can diverge.
- There are three main factors that can cause divergence: numerical roundoff errors, modeling errors, and lack of observability of one or more state variables.

→ "lack of observability" means that the observations contain no information about one or more elements of the state vector.

EX:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u_k$$

$$z_k = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k + v_k$$

→ In this case, the observations contain no information about the state variable $x_3$.

→ Thus, as $k \to \infty$, our uncertainty about $x_3$ can grow without bound. This causes $P_k$ to diverge as $k \to \infty$.

# Roundoff Errors

- Numerical roundoff errors can accumulate and cause divergence problems as the number of steps in k becomes large.

- If
  1. All state variables are observable
  2. The input noise $w_k$ drives all state variables

  → Then the Kalman filter has a certain "degree of natural stability".

  → A steady state solution for $P_k$ normally exists, even if the system dynamics are nonstationary.

  → As long as the iterative solution for $P_k$ remains positive definite (i.e., symmetric), it will tend to return to the steady state solution after a perturbation.

  NOTE: The P-update equation

$$P_k = (I - K_k H_k) P_k^- \quad (5.5.22)$$

  is not guaranteed to remain positive definite when finite precision arithmetic is used.

- Divergence problems due to roundoff errors must generally be handled on a case by case basis.

- The following may help:

1. Use high-precision arithmetic whenever possible.

2. If the observations are sparse, so that "k" may advance many steps between the arrival of consecutive observations, then it is advisable to age the P matrix using the N-step update

$$P(k+N|k) = \Phi(k+N,k)\,P(k|k)\,\Phi^T(k+N|k) + Q(k+N,k) \qquad (6.1.2)$$

instead of iterating the usual one-step update

$$P_k = (I - K_k H_k)\,P_k^- \qquad (5.5.22)$$

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k \qquad (5.5.25)$$

3. Avoid situations where one or more state variables are not driven by $w_k$, i.e., avoid models that have deterministic state variables.

   - A small error can cause the P matrix to lose symmetry and diverge.

   A) Reformulate the state model to remove deterministic state variables

   B) Inject noise into the states by adding small positive quantities to the elements of the major diagonal of $Q_k$.

Note: B) above makes the Kalman filter suboptimal (because the system model is incorrect), but this is preferable to divergence.

4. Force the $P_k$ and $P_k^-$ matrices to be symmetric at each iteration of "k". One way to do this is to propagate only the upper or lower diagonal factorization of $P_k$ and $P_k^-$.

5. Avoid large uncertainty in $P_k^-$ when a highly precise observation $z_k$ will force $P_k$ to be small (e.g., when $V_k$ has a very small covariance matrix).

→ In this case, the $P_k$ update equation

$$P_k = (I - K_k H_k) P_k^- \quad (5.5.22)$$

will approach the indeterminate form $0 \times \infty$.

→ One solution is to reduce the magnitudes of the entries of $P_k^-$. This makes the Kalman filter suboptimal, but is again preferable to divergence.

NOTE: For problems 4. and 5. above, it may also help to use an alternative P update equation such as

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (5.5.18)$$

which has natural symmetry and is more stable numerically than (5.5.22).

## Modeling Errors

- The Kalman filter is generally quite sensitive to modeling errors.
  - → Try to avoid them!
  - → Any instruments used in measuring the observations $z_k$ should be kept well calibrated.
  - → Almost all physical quantities drift at least a little bit; thus, beware of constant valued "deterministic" state variables. If they cannot be avoided, consider injecting noise as in 3.B) above.

## Observability Problems

- Sometimes the measurements $z_k$ do not provide enough information to estimate all of the state variables.
- One or more state variables may be hidden from view and not observable ("internal" state variables).
- If the hidden variables are unstable, the P matrix will diverge even if there are no roundoff errors.

→

- Diagonal entries in the P matrix corresponding to the unobservable states may diverge even if the process is stable,
- The only real solution to this problem is to add additional measurements to the $z_k$ vector that will provide information about the hidden state variables.

## Relationship to Deterministic Least Squares Fitting

- Consider a set of m linear equations in $x = [x_1\ x_2\ \cdots\ x_n]^T$ :

$$Mx = b$$

$M$: $m \times n$
$x$: $n \times 1$
$b$: $n \times 1$.

- Suppose $m > n$, so that the system is overdetermined.
- Assume also that the system is inconsistent.
- The problem is to find an optimal solution $x_{opt}$ for $x$ that will minimize the error $\varepsilon$ given by

$$Mx_{opt} - b = \varepsilon.$$

for the "classical" least squares formulation.

- We generalize by considering the "weighted" squared-error

$$\mathcal{E}_w^T \mathcal{E}_w = (Mx_{opt} - b) W (Mx_{opt} - b)^T.$$

→ This is called the "weighted least squares" problem.

→ The classical least squares problem is obtained by taking $W = I$.

- Differentiating $\mathcal{E}_w^T \mathcal{E}_w$ with respect to $x_{opt}$ and setting the result equal to zero, we obtain

$$2(M^T W M) x_{opt} - (b^T W M)^T - M^T W b = 0.$$

- The deterministic weighted least squares solution is given by

$$x_{opt} = [(M^T W M)^{-1} M^T W] b.$$

- The Kalman filter solution proceeds as follows:

  - $x$ is assumed to be a random constant.
  - The system dynamical equation is therefore

$$\dot{x} = 0.$$

  - The discrete state space model is

$$x_{k+1} = I x_k + 0 w_k$$

$$z_k = H_k x_k + v_k$$

where $z_k \leftrightarrow b$ and $H_k \leftrightarrow M$.

- Given an initial estimate $\hat{x}_0^-$ and associated error covariance matrix $P_0^-$, the a posteriori estimate $\hat{x}_0$ will correspond to the deterministic weighted least squares solution, provided;

1. We take $\hat{x}_0^- = 0$
2. We take $P_0^- = \text{diag}(\infty)$.
3. We take $R_0^- = W$.

- Using the alternative Kalman filter formulation, we then have

$$P_0^{-1} = (P_0^-)^{-1} + H_0^T R_0^{-1} H_0$$

$$= H_0^T R_0^{-1} H_0$$

$$K_0 = P_0 H_0^T R_0^{-1}$$

$$= (H_0^T R_0^{-1} H_0)^{-1} H_0^T R_0^{-1}$$

$$\hat{x}_0 = \hat{x}_0^- + K_0 (z_0 - H_0 \hat{x}_0^-)$$

$$= K_0 z_0$$

$$= \left[(H_0^T R_0^{-1} H_0)^{-1} H_0^T R_0^{-1}\right] z_0$$

$$= \left[(M^T W M)^{-1} M^T W\right] b \quad \checkmark$$

→ In this case, the weighted least squares solution agrees with the Kalman filter solution when we have no a priori knowledge of $x_0$ and no knowledge of the noise statistics.

# Stability

- If we assume that the Kalman filter has converged to a steady state, then $z$-transform analysis can be used to assess stability in the usual sense.

- The estimated state vector update equation is

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-)$$

- Substituting $\hat{x}_k^- = \phi_{k-1} \hat{x}_{k-1}$, we obtain

$$\hat{x}_k = \phi_{k-1} \hat{x}_{k-1} + K_k (z_k - H_k \phi_{k-1} \hat{x}_{k-1})$$

$$= (\phi_{k-1} - K_k H_k \phi_{k-1}) \hat{x}_{k-1} + K_k z_k$$

- Taking the $z$-transform on both sides,

$$\hat{X}_k(z) = (\phi_{k-1} - K_k H_k \phi_{k-1}) z^{-1} \hat{X}_k(z) + K_k Z_k(z)$$

$$[zI - (\phi_{k-1} - K_k H_k \phi_{k-1})] \hat{X}_k(z) = z K_k Z_k(z)$$

- The vector transfer function is then

$$\hat{X}_k(z) Z_k^{-1}(z) = z K_k [zI - (\phi_{k-1} - K_k H_k \phi_{k-1})]^{-1}$$

output ↑ input ↑

- Thus, the Kalman filter at steady state is stable in the usual sense if all the roots of the characteristic polynomial

$$\det\left[zI - (\Phi_{k-1} - K_k H_k \Phi_{k-1})\right] = 0$$

lie <u>inside</u> the unit circle of the z-plane.

for a moment and reflect on just what we have (and do not have) with this thing we call a Kalman filter.

1. The Kalman filter is intended to be used for estimating *random* processes. Any application in a nonrandom setting must be viewed with caution (see Example 6.6).
2. The Kalman filter is model-dependent. This is to say that we assume that we know a priori the model parameters. These, in turn, come from the second-order "statistics" of the various processes involved in the application at hand. Therefore, in its most primitive form, the Kalman filter is not adaptive or self-learning.
3. The Kalman filter is a linear estimator. When all the processes involved are Gaussian, the filter is optimal in the minimum-mean-square-error sense within a class of *all* estimators, linear and nonlinear. [See Meditch (2) for a good discussion of optimality.]
4. Various Kalman filter recursive algorithms exist. The "usual" algorithm was given in Chapter 5, an alternative one was presented in Chapter 6, and a third one (*U-D* factorization) will be presented in Chapter 9. All of these yield identical results (assuming perfect arithmetic).
5. Under certain special circumstances, the Kalman filter yields the same result obtained from deterministic least squares (see Section 6.8).
6. Kalman filtering is especially useful as an analysis tool in off-line error analysis studies. The optimal filter error covariance equation can be propagated recursively without actual (or simulated) measurement data. This is also true for the suboptimal filter with some restrictions (see Section 6.7).

With these brief comments in mind, we are now ready to proceed to variations on the original discrete Kalman filter. It is worth mentioning that the discrete filter came first historically (1960). The continuous version and other variations followed the discrete filter.

## PROBLEMS

**6.1** The process of landing on an aircraft carrier is a highly complex operation primarily because the carrier deck is constantly in motion with a certain degree of randomness that is attributable to wind and sea conditions. In particular, one motion called *heaving* changes the vertical displacement of the carrier deck. Accurate prediction of the heave motion even 10 to 15 sec into the future will significantly enhance the success of the landing operation.

In a paper published in 1983, Sidar and Doolin (21) suggested using Kalman filter methods to predict the motion of the carrier deck. On the basis of empirical data, they developed a power spectral density (PSD) for the heave motion, and then they worked out an optimal predictor based on this spectral model. The functional form for the PSD to be used here comes from the Sidar–Doolin paper, but the amplitude factor has been changed for convenience. Also, the measurement noise variance $\mathbf{R}_k$ and the sampling interval used here are hypothetical. Thus, there is no claim that the results of this problem represent an exact real-life situation.