Discrete Kalman Filter

M. Sami Fadali
Professor EE
University of Nevada
Outline

- What is the Discrete Kalman Filter (DKF)?
- Derivation of DKF.
- Implementation of DKF.
- Example
What is the (DKF)?

- Algorithm for the optimal recursive estimation of the state of a system.

- Needs:
  - Initial state estimate and error covariance.
  - Noisy measurements with known properties.
  - System state-space model.
Derivation of DKF

Process and measurement models

\[ x_{k+1} = \phi_k x_k + w_k \]
\[ z_k = H_k x_k + v_k \]

\( x_k \) = \( n \times 1 \) state vector at \( t_k \)
\( \phi_k \) = \( n \times n \) state-transition matrix at \( t_k \)
\( z_k \) = \( m \times 1 \) measurement vector at \( t_k \)
\( H_k \) = \( m \times n \) measurement matrix matrix at \( t_k \)
Noise

\( w_k = n \times 1 \) zero-mean white Gaussian process noise vector at \( t_k \)

\[
E\{w_k w_i^T\} = \begin{cases} 
Q_k, & i = k \\
0, & i \neq k 
\end{cases}
\]

\( v_k = m \times 1 \) zero-mean white Gaussian measurement noise vector at \( t_k \)

\[
E\{v_k v_i^T\} = \begin{cases} 
R_k, & i = k \\
0, & i \neq k 
\end{cases}
\]

\[
E\{w_k v_i^T\} = 0
\]
Notation

\( ^\wedge = \) estimate

\( \sim = \) perturbation.

\( \hat{x}_k^- \) = a priori estimate of \( x_k \) (before the measurement at \( t_k \)).

\( \hat{x}_k^+ \) = a posteriori estimate of \( x_k \) (after the measurement at \( t_k \)).
Measurement Noise

- Measurement noise $\nu_k$ is white.

$$z_k = H_k x_k + \nu_k$$

- A priori estimate at time $k$ (i.e. $\hat{x}_k^-$) uncorrelated with measurement noise at time $k$.

$$\hat{x}_k^- = f(z_i, i = 0, 1, ..., k - 1, ICS)$$

$$E\{(x_k - \hat{x}_k^-)\nu_k^T\} = [0]$$
Estimators

- **A priori estimate**
  \[ \hat{x}^-_k = f(z_i, i = 0,1, ..., k - 1, ICs) \]

- **A posteriori estimate**
  \[ \hat{x}^+_k = K_{x,k} \hat{x}^-_k + K_k z_k \]

- **A priori estimation error**
  \[ e^-_k = \hat{x}^-_k - x_k \]

- **A posteriori estimation error**
  \[
  e^+_k = \hat{x}^+_k - x_k = K_{x,k} \hat{x}^-_k + K_k (H_k x_k + v_k) - x_k \\
  = K_{x,k} \hat{x}^-_k + (K_k H_k - I_n) x_k + K_k v_k
  \]
Unbiased Linear Estimator

A posteriori estimate
\[ \hat{x}_k^+ = K_{x,k} \hat{x}_k^- + K_k z_k \]

Error
\[ \hat{x}_k^+ - x_k = K_{x,k} \hat{x}_k^- + (K_k H_k - I_n) x_k + K_k \nu_k \]

Expectation must be zero for unbiased
\[ E\{\hat{x}_k^+ - x_k\} = K_{x,k} E\{\hat{x}_k^-\} - (I_n - K_k H_k) E\{x_k\} = 0 \]

Assume unbiased a priori estimate
\[ E\{\hat{x}_k^-\} = E\{x_k\} \]

For unbiased estimator
\[ K_{x,k} = I_n - K_k H_k \]
\[ \hat{x}_k^+ = (I_n - K_k H_k) \hat{x}_k^- + K_k z_k = \hat{x}_k^- + K_k [z_k - H_k \hat{x}_k^-] \]
Error Covariance Matrices

- Assume unbiased estimates: \( E\{e_k\} = 0 \)

- A priori error \( e_k^- = x_k - \hat{x}_k^- \)

- A priori error covariance matrix \( P_k^- = E\{e_k^- e_k^- T\} \)

- A posteriori error

\[
  e_k^+ = x_k - \hat{x}_k^+ = (I_n - K_k H_k) e_k^- - K_k v_k
\]

- A posteriori error covariance matrix

\[
  P_k^+ = E\{e_k^+ e_k^+ T\} = \left[ E\{e_{ki}^+ e_{kj}^+ \} \right]
\]
Derivation of DKF

- Recursively correct estimate.

\[
\hat{x}_k^+ = (I_n - K_k H_k)\hat{x}_k^- + K_k z_k
\]
\[
= (I_n - K_k H_k)\hat{x}_k^- + K_k (H_k x_k + \nu_k)
\]
\[
= \hat{x}_k^- + K_k H_k (x_k - \hat{x}_k^-) + K_k \nu_k
\]
\[
e_k^+ = x_k - \hat{x}_k^+ = (I_n - K_k H_k) (x_k - \hat{x}_k^-) - K_k \nu_k
\]
\[
= (I_n - K_k H_k) e_k^- - K_k \nu_k
\]

- Measurement noise orthogonal to a priori error:

\[
E\{(x_k - \hat{x}_k^-)\nu_k^T\} = E\{e_k^- \nu_k^T\} = [0]
\]
Error Covariance Matrix

\( \mathbf{e}_k^+ = (I_n - K_k H_k)\mathbf{e}_k^- - K_k \mathbf{v}_k, \quad E\{\mathbf{e}_k^- \mathbf{v}_k^T\} = [0] \)

\[
P_k^+ = E\left\{\mathbf{e}_k^+ \mathbf{e}_k^{+T}\right\} = E\left\{(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)^T\right\}
\]

\[
= (I_n - K_k H_k)E\{\mathbf{e}_k^- \mathbf{e}_k^{-T}\}(I_n - K_k H_k)^T + K_k E\{\mathbf{v}_k \mathbf{v}_k^T\}K_k^T
\]

- Substitute for the expectations

\[
P_k^+ = (I_n - K_k H_k)P_k^- (I_n - K_k H_k)^T + K_k R_k K_k^T
\]

- Expand

\[
P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k (H_k P_k^- H_k^T + R_k) K_k^T
\]
Minimum Mean-square Error

Choose gain $K_k$ (blending factor) to minimize mean square error $E \left\{ e_k^+ e_k^+ \right\}$

$$E \left\{ e_k^+ e_k^+ \right\} = \text{tr} \left( E \left\{ e_k^+ e_k^+^T \right\} \right) = \text{tr} \left( \left[ E \left\{ e_k^+ e_k^+ \right\} \right] \right)$$

$$\sum_{i=0}^{n} E \left\{ (e_{ki}^+)^2 \right\} = \text{tr} \left[ E \left\{ e_k^+ e_k^+^T \right\} \right] = \text{tr}[P_k^+]$$

Minimize over all possible choices of $K$

$$\partial \sum_{i=0}^{n} E \left\{ (e_{ki}^+)^2 \right\} / \partial K_k = \partial \text{tr}[P_k^+] / \partial K_k$$
Derivative of Trace

For any scalar $s$

$$\frac{d}{dA} s = \begin{bmatrix} \frac{d}{dA} s \\ \frac{d}{dA} a_{ij} \end{bmatrix}$$

$$\frac{d}{dA} \text{tr}[AB] = B^T \text{ (A, B square)}$$

$$\frac{d}{dA} \text{tr}[ACA^T] = 2AC \text{ (C symmetric)}$$
Minimization

\[ P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k \left( H_k P_k^- H_k^T + R_k \right) K_k^T \]

- Use \( \text{tr}[X] = \text{tr}[X^T] \) (same trace for two terms)
- Apply trace formulas, use \( \text{tr}[Q] = \text{tr}[Q^T] \)

\[
\frac{\partial \text{tr}[P_k^+]}{\partial K_k} = -2P_k^- H_k^T + 2K_k \left( H_k P_k^- H_k^T + R_k \right) = [0]
\]

- Solve for the Kalman Gain

\[
K_k = P_k^- H_k^T \left( H_k P_k^- H_k^T + R_k \right)^{-1}
\]
Error Covariance Matrix Forms

- **Joseph form**

\[
P_k^+ = (I_n - K_k H_k) P_k^- (I_n - K_k H_k)^T + K_k R_k K_k^T
\]

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

\[
= P_k^- - K_k (H_k P_k^- H_k^T + R_k) K_k^T
\]

\[
= (I - K_k H_k) P_k^-
\]

- Four expressions for the error covariance.

- **Numerical Computation:** behave differently.
Joseph Form

\[ P^+_k = (I_n - K_k H_k) P^-_k (I_n - K_k H_k)^T + K_k R_k K_k^T \]

- Expand to obtain the other forms.
- Best numerical computation properties.
- Use Joseph form to reduce numerical errors.
Derivation of Other Forms

- Derived earlier

\[ P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k \left( H_k P_k^- H_k^T + R_k \right) K_k^T \]

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

- Three equal terms

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

\[ = K_k \left( H_k P_k^- H_k^T + R_k \right) \left( H_k P_k^- H_k^T + R_k \right)^{-1} H_k P_k^- \]
Derivation (Cont.)

\[ P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k \left( H_k P_k^- H_k^T + R_k \right) K_k^T \]

- Cancel two equal terms (two forms)

\[ P_k^+ = P_k^- - K_k \left( H_k P_k^- H_k^T + R_k \right) K_k^T \]
\[ = P_k^- - K_k H_k P_k^- \]

- Common factor

\[ P_k^+ = \left( I_n - K_k H_k \right) P_k^- \]
Fundamental Thm. of Estimation Theory

- Minimum mean-square error estimator:
  \[ \hat{x}_k = E\{x_k | z^*_k\} \]
  \[ z^*_k = col\{z_i, i = 0,1, ..., k\}. \]

- \( z^*_k \) = all measurements up to and including time \( k \)

- State estimate depends on all the measurements.
Proof

\[ MSE = E\left\{ [x_k - \hat{x}_k]^T [x_k - \hat{x}_k] | z_k^* \right\} \]

- Expand and complete squares.

\[ MSE = E\left\{ x_k^T x_k | z_k^* \right\} \]
\[ + [\hat{x}_k - E\{x_k | z_k^*\}]^T [\hat{x}(k) - E\{x_k | z_k^*\}] \]
\[ - E\{x_k^T | z_k^*\} E\{x_k | z_k^*\} \]

- Choose \( \hat{x}_k \) to minimize mean-square error.
- Minimized if \( \hat{x}_k = E\{x_k | z_k^*\} \)
A Priori Estimate

- Predictor

\[ x_{k+1} = \phi_k x_k + w_k \]

\[ \hat{x}_{k+1}^- = E\{x_{k+1} | z_k^*\} = \phi_k E\{x_k | z_k^*\} + 0 = \phi_k \hat{x}_k^+ \]

\[ \hat{x}_{k+1}^- = \phi_k \hat{x}_k^+ \]

- Estimation error

\[ e_{k+1}^- = x_{k+1} - \hat{x}_{k+1}^- \]

\[ = (\phi_k x_k + w_k) - \phi_k \hat{x}_k^+ \]

\[ e_{k+1}^- = \phi_k e_k^+ + w_k \]
A Priori Error Covariance Matrix

- Error
  \[ e_{k+1}^- = \phi_k e_k^+ + w_k \]

- Sum of two orthogonal terms \( E\{e_k^+ w_k^T\} = [0] \)

- Error covariance matrix
  \[
  P_{k+1}^- = E \left\{ e_{k+1}^- (e_{k+1}^-)^T \right\} \\
  = \phi_k E \left\{ e_k^+ (e_k^+)^T \right\} \phi_k^T + E\{w_k w_k^T\} \\
  P_{k+1}^- = \phi_k P_k^+ \phi_k^T + Q_k
  \]
"DKF Loop"

Enter initial state estimate and its error covariance
\( \hat{x}_0^-, P_0^- \)

Compute Kalman Gain
\[
K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}
\]

Project Ahead:
\[
\hat{x}_{k+1}^- = \phi_k \hat{x}_k^+
\]
\[
P_{k+1}^- = \phi_k P_k^+ \phi_k^T + Q_k
\]

Compute error covariance
\[
P_k^+ = (I_n - K_k H_k) P_k^-
\]

Measurements
\( \{z_0, z_1, \ldots \} \)

Update estimate with measurement \( z_k \)
\[
\hat{x}_k^+ = \hat{x}_k^- + K_k [z_k - H_k \hat{x}_k^-]
\]

State Estimates
\( \{ \hat{x}_0^+, \hat{x}_1^+, \ldots \} \)
Example: Wiener Process

- Scalar example.
- Discretize the CT system.
- St. dev. of measurement error $= 0.5 \Rightarrow R = 0.25$

$$G(s) = \frac{1}{s} \iff g(t) = 1$$
Example: Discretization

\[ \dot{x}(t) = u(t) \]
\[ x_{k+1} = x_k + w_k, \quad \phi_k = 1 \]
\[ z_k = x_k + v_k, \quad H_k = 1 \]

\[ Q_k = E\{w_k^2\} = E \left\{ \left( \int_0^{\Delta t} 1. u(\xi)d\xi \right) \left( \int_0^{\Delta t} 1. u(\eta)d\eta \right) \right\} \]
\[ = \int_0^1 \int_0^1 E\{u(\xi)u(\eta)\}d\xi \ d\eta \]
\[ = \int_0^1 \int_0^1 \delta(\xi - \eta)d\xi \ d\eta = 1 \]
\[ \hat{x}_0^- = 0, \quad P_0^- = 0, \quad R = 1/4 \]
Kalman Loop: $k = 0$

- Calculate the gain
  \[ K_0 = P_0^- H_0^T (H_0 P_0^- H_0^T + R_0)^{-1} = 0/(0 + 1/4) = 0 \]

- Update the estimate
  \[ \hat{x}_0^+ = \hat{x}_0^- + K_0 (z_0 - H \hat{x}_0^-) = 0 + 0(z_0 - 0) = 0 \]

- Update the error
  \[ P_0^+ = (I - K_0 H)P_0^- = (1 - 0)(0) = 0 \]

- Project ahead
  \[ \hat{x}_1^- = \phi \hat{x}_0^+ = 1.0 = 0 \]
  \[ P_1^- = \phi P_0^+ \phi^T + Q = 1. (0). 1 + 1 = 1 \]
Kalman Loop: $k = 1$

- **Calculate the gain**
  \[ K_1 = P_1^- H_1^T (H_1 P_1^- H_1^T + R_1)^{-1} = 1/(1 + 1/4)^{-1} = 4/5 \]

- **Update the estimate**
  \[ \hat{x}_1^+ = \hat{x}_1^- + K_1 (z_1 - H \hat{x}_1^-) = 0 + \frac{4}{5} (z_1 - 0) = \frac{4}{5} z_1 \]

- **Update the error**
  \[ P_1^+ = (I - K_1 H) P_1^- = (1 - 4/5)(1) = 1/5 \]

- **Project ahead**
  \[ \hat{x}_2^- = \phi \hat{x}_1^+ = 1. (4/5)z_1 = (4/5)z_1 \]
  \[ P_2^- = \phi P_1^+ \phi^T + Q = 1. (1/5). 1 + 1 = 6/5 \]
% Across Measurements:
K = P*H'/(H*P*H’+R);
xhatp = xhat + K*(z-H*xhat);
P = (eye(n)-K*H)*P;
P = (P+P’)/2;

% Between Measurements:
xhat = phi*xhat;
P = phi*P*phi’+Q;
Example: Gauss-Markov Process

\[ R_{XX}(\tau) = \sigma^2 e^{-\beta |\tau|}, \quad S_{XX}(s) = \frac{2\sigma^2 \beta}{-s^2 + \beta^2} \]

\[ G(s) = S_{XX}^+(s) = \frac{\sqrt{2\sigma^2 \beta}}{s + \beta}, \quad g(t) = \sqrt{2\sigma^2 \beta} e^{-\beta t} \]

\[ \dot{x} = -\beta x + \sqrt{2\sigma^2 \beta} u, \quad y = x \]
Example: Discretization

\[ \phi = e^{-\beta \Delta t}, \quad H = W = 1, \quad G = \sqrt{2\sigma^2 \beta} \]

\[ Q(k) = \int_0^{\Delta t} e^{F\xi} GWG^T e^{F^T\xi} d\xi \]

\[ = \int_0^{\Delta t} \left( e^{-\beta \Delta t \sqrt{2\sigma^2 \beta}} \right)^2 (1) d\xi \]

\[ = \sigma^2 \left( 1 - e^{-2\beta \Delta t} \right) \]

\[ x_{k+1} = e^{-\beta \Delta t} x_k + w_k \]

\[ z_k = x_k + v_k \]

\[ R = 1 \]
Initial Conditions

- Process mean and variance
  \[ R_{xx}(\tau) = \sigma^2 e^{-\beta |\tau|} \]
  \[ \lim_{\tau \to \infty} R_{xx}(\tau) = 0 \Rightarrow E\{x(t)\} = 0 \]
  \[ R_{xx}(0) = E\{x^2(t)\} = \sigma^2 = \text{var}\{x(t)\} \]

- Use process mean and variance to initialize
  \[ \hat{x}_0^- = 0, \quad P_0^- = \sigma^2 \]
Simulation Results

- Use unity variance $\sigma^2 (P_0^{-} = 1m^2)$ and unity time constant, $\beta = 1$.
- Use a sampling period $\Delta t = 0.02$ s
- Steady-state Kalman gain $K = 0.165$
- Steady-state error variance $P = 0.16m^2$,
- RMS Error $\sqrt{P} = 0.4m$
- Close to steady state after 20 steps.
- Suboptimal filter: Use $K = 0.165$ for a simple implementation.
Simulation Results

Figure 4.5
© John Wiley & Sons, Inc. All rights reserved.
Discrete Lyapunov Equation

\[ P_{k+1}^- = \phi_k P_k^+ \phi_k^T + Q_k \]

- **Substitute for** \( P^+ \)
  \[ P_{k+1}^- = \phi_k (I_n - K_k H_k) P_k^- (I_n - K_k H_k)^T \phi_k^T \]
  \[ + \phi_k K_k R_k K_k^T \phi_k^T + Q_k \]

- **Lyapunov Equation:** \( P_{k+1}^- = \bar{A}_k P_k^- \bar{A}_k^T + \bar{Q}_k \)
  \[ \bar{A}_k = \phi_k (I_n - K_k H_k) \]
  \[ \bar{Q}_k = \phi_k K_k R_k K_k^T \phi_k^T + Q_k \]

- **Applies for any gain** \( K \) (not just the optimal Kalman gain \( K \))
Solution of Lyapunov Eqn.

\[ P_{k+1}^- = \bar{A}_k P_k^- \bar{A}_k^T + \bar{Q}_k \]

\[ P_k^- = \Phi(k, 0) P_0^- \Phi^T(k, 0) \]

\[ + \sum_{i=0}^{k-1} \Phi(k, i + 1) \bar{Q}_i \Phi^T(k, i + 1) \]

\[ \Phi(k, i) = \begin{cases} \bar{A}_{k-1} \bar{A}_{k-2} ... \bar{A}_i \\ \bar{A}^{k-i}, \bar{A} \text{ constant} \end{cases} \]

\[ \Phi(k, k) = I_n \]

- Proof by induction.
Discrete Riccati Equation

\[ P_{k+1}^- = \phi_k P_k^+ \phi_k^T + Q_k \]

- For the Kalman (optimal) gain (slide 14)
  \[ P_k^+ = P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- \]

- Riccati Equation
  \[ P_{k+1}^- = \phi_k \left[ P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- \right] \phi_k^T + Q_k \]
Wiener Example: Lyapunov Equation

\[ \phi_k = 1, \quad H = 1, \quad Q = 1, \quad R = 0.25 \]

\[ P_{k+1}^- = \bar{A}_k P_k^- \bar{A}_k^T + \bar{Q}_k \]

\[ \bar{A}_k = \phi_k (I_n - K_k H_k) = 1(1 - K_k \times 1) = 1 - K_k \]

\[ \bar{Q}_k = \phi_k K_k R_k K_k^T \phi_k^T + Q_k = 1. (K_k^2 / 4). 1 + 1 \]

\[ = K_k^2 / 4 + 1 \]

\[ P_{k+1}^- = \bar{A}_k P_k^- \bar{A}_k^T + \bar{Q}_k \]

\[ = (1 - K_k)^2 P_k^- + K_k^2 / 4 + 1 \]
Wiener Example: Riccati Equation

\[ \phi_k = 1, \quad H = 1, \quad Q = 1, \quad R = 0.25 \]

\[ P_{k+1}^- = \phi_k \left[ P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- \right] \phi_k^T + Q_k \]

\[ = 1 \left[ P_k^- - P_k^- \cdot 1 \cdot (1 \cdot P_k^- \cdot 1 + 0.25)^{-1} \cdot 1 \cdot P_k^- \right] \cdot 1 + 1 \]

\[ = \left[ 1 - P_k^- / (P_k^- + 0.25) \right] P_k^- + 1 \]

\[ P_{k+1}^- = 1.25 - 0.25 / (4P_k^- + 1) \]
Deterministic Inputs

\[ \dot{x} = Fx + Bu_d + Gu, \quad x_0 \]

Linear system: use superposition.

a. Add zero-state deterministic response

\[ \hat{x}_{k+1}^- = \phi_k \hat{x}_k^+ + \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \tau)B(\tau)u_d d\tau \]

b. (i) Subtract deterministic output from \( z_k \)
(ii) Compute \( x_{k,d} \) separately and add to the KF estimate to obtain the state estimate.

\[ \hat{x}_{k,tot} = \hat{x}_k + x_{k,d} \]
Real-time Implementation

- **Data Latency:** delay between data time and current time due to sensor, computation, and information delivery.

- **Processor Loading:** how much is the processor used?
  - throughput (bits/s) analysis
  - dedicated vs. shared processor
  - specialized exploitation to reduce computation e.g. exploit sparse matrices.
Round-off Errors

- Use high precision off line.
- Choose step carefully.
- Propagate $n(n + 1)/2$ unique terms of symmetric $P$ matrix only.
- Use array algorithms that propagate the square root of the matrix $P$ (see Kailath, covered later).
- Use suboptimal filter (fix $K$).
The Separation Principle

- Linear system with state estimator feedback.
- Design controller and state estimator separately.
- True for observer or Kalman filter.
Conclusion

- Popular recursive algorithm.
- Minimize mean-square error.
- Real-time implementation.
- Estimator feedback.