Outline

- Linearized KF Linearize about a nominal trajectory.
- Extended KF Linearize about the estimated trajectory.
- Example.

Linearized KF

- Linearize about a nominal trajectory.
- Assume that the actual trajectory is approximately known.
- Nonlinear process and/or measurement model.

Extended Kalman Filter

- Linearize about the estimated trajectory \( \hat{x}_k^+ \) instead of nominal.
- Proof is similar to linearized KF.
- Can include higher order terms but the performance is not necessarily better.
Derivation of LKF/EKF

Process and measurement models
\[
\begin{align*}
\dot{x}(t) &= f(x(t), u_d(t), t) + u(t) \\
z(t) &= h(x(t), t) + v(t) \\
x(t) &= n \times 1 \text{ state vector} \\
u_d(t) &= p \times 1 \text{ reference input vector.} \\
u(t) &= p \times 1 \text{ noise vector.} \\
z(t) &= z \times 1 \text{ measurement vector} \\
f, h &= \text{vectors of known continuous functions.}
\end{align*}
\]

Linearized KF: Nominal Trajectory

\[
x^*(t) = n \times 1 \text{ nominal (reference) trajectory}
\]
- Approximately equal to the actual trajectory.
- Corresponds to the deterministic input \( u_d(t) \)

\[
\begin{align*}
\dot{x}^*(t) &= f(x^*(t), u_d(t), t) \\
x(t) &= x^*(t) + \Delta x(t) \\
\dot{x}^*(t) + \Delta \dot{x}(t) &= f(x^*(t) + \Delta x(t), u_d(t), t) + u(t)
\end{align*}
\]

Trajectories

Actual Trajectory \( x \)
Nominal Trajectory \( x^* \)

Linearization: State Equation

\[
\begin{align*}
\dot{x}^*(t) + \Delta \dot{x}(t) &= f(x^* + \Delta x, u_d(t)) + u(t) \\
&\approx f(x^*, u_d(t), t) + \left( \frac{\partial f(x, u_d, t)}{\partial x} \right)_{(x^*, u_d)} \Delta x + u(t)
\end{align*}
\]

- Cancel nominal terms.

\[
\begin{align*}
\Delta \dot{x}(t) &\approx \left( \frac{\partial f(x, u_d, t)}{\partial x} \right)_{(x^*, u_d)} \Delta x + u(t) = F \Delta x + u(t)
\end{align*}
\]
- Discretize for KF: \( \Delta x_{k+1} = \phi \Delta x_k + w_k \)
Discretization

\[ \dot{x}(t) = f(x(t), u_d(t), t) \]
- Euler forward approximation
  \[ \dot{x}(t) \approx \frac{x(t_{k+1}) - x(t_k)}{\Delta t} = f(x(t_k), u_d(t_k), t_k) \]
  \[ x(t_{k+1}) = x(t_k) + f(x(t_k), u_d(t_k), t_k) \Delta t \]
- Simple but can cause numerical instability
- Better approximations available (e.g. RK)

Linearization: Output Equation

\[ z(t) = h(x^* + \Delta x, t) + v(t) \]
\[ \approx h(x^*, t) + \frac{\partial h(x,t)}{\partial x} \bigg|_{x^*} \Delta x + v(t) \]
\[ \Delta z(t) = z(t) - h(x^*, t) \]
\[ \approx \frac{\partial h(x,t)}{\partial x} \bigg|_{x^*} \Delta x + v(t) = H\Delta x + v(t) \]
- Discretize the system then design a DKF.

Linearized (Extended) KF: Predictor

- \( \hat{x}^-_{k+1} = \) solution of nonlinear equation \( \dot{x}(t) = f(x(t), u_d(t), t) \) at \( t_{k+1} \) with IC \( x_k^* (\hat{x}_k^+) \)
- Often use an approximate solution.
- Use \( x_k^* (\hat{x}_k^+) \) for the linearized (extended) Kalman filter.
- Error Covariance Matrix (discretize)
  \[ P^-_{k+1} = \phi_k P^+_k \phi_k^T + Q_k \]

Which linearization is better?

- Linearization: suboptimal filters
- Estimate of the true trajectory better in a statistical sense (on the average).
- Extended KF better in a statistical sense.
- Linearized KF may perform better.
- Choice depends on the application.
linearized/Extended KF: Corrector

- Minor changes from the DKF
  \[ \Delta \hat{x}_k^+ = \Delta \hat{x}_k^- + K_k [\Delta z_k - H_k \Delta \hat{x}_k^-] \]
  \[ x_k^+ + \Delta x_k^+ = x_k^- + \Delta x_k^+ + K_k [z_k - h(x_k^-, t_k) - H_k \Delta \hat{x}_k^-] \]
  \[ \hat{x}_k^+ = \hat{x}_k^- + K_k [z_k - h(x_k^-, t_k) - H_k \Delta \hat{x}_k^-] \]
  \[ = \hat{x}_k^- + K_k [z_k - \hat{z}_k^-] \]

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

EKF Loop

- Enter initial state estimate and its error covariance \( \hat{x}_0, P_0^- \)
- Compute Kalman Gain
  \[ K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \]
- Linearize
  \[ F = \frac{\partial f}{\partial x}|_{x_k, u_k, \phi_k, H_k} \]
- Project Ahead:
  \[ \hat{x}_{k+1} = f(\hat{x}_k^+, \Delta t) + x_k \]
  \[ P_{k+1}^- = \phi_k P_k^+ \phi_k^T + Q_k \]
- Update estimate with measurement \( z_k \)
  \[ \hat{x}_k^+ = \hat{x}_k^- + K_k [z_k - h(x_k^-, t_k) - H_k \Delta \hat{x}_k^-] \]
- Compute error covariance
  \[ P_k^+ = (I_n - K_k H_k) P_k^- (I_n - K_k H_k)^T + K_k R_k K_k^T \]

Error Covariance (EKF)

- Discrete-time model
  \[ x(t_{k+1}) = f(x(t_k), u_d(t_k), t_k) + w(k) \]
  \[ \hat{x}_{k+1} = E\{x(t_{k+1})|z_{1:k}\} \approx f(\hat{x}_k, u_d(t_k), t_k) \]
  \[ \epsilon_{k+1} \approx \Delta f(x, u_d, t) \]
  \[ e_k^+ = \phi_k e_k^- + w(k) \]
  \[ P_{k+1}^- = E\{e_{k+1} e_{k+1}^T\} = \phi_k P_k^+ \phi_k^T + Q_k \]
Hybrid EKF

- Continuous-time system with discrete Kalman filter.
- Sample output.
- Predictor
  - Use continuous dynamics to derive the update for the error covariance.
  - Use an approximation to update the state and the error covariance.

Covariance Equation

- From the theory of the continuous time Kalman filter
  \[
  \dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t) - P(t)H^T(t)R^{-1}H(t)P(t)
  \]
- Between measurements: no knowledge makes \( R \to \infty \)
  \[
  \dot{P}(t) = F(t)P(t) + P(t)F^T(t) + G(t)Q(t)G^T(t)
  \]
  \[
  F_k = F(t_k), \quad G_k = G(t_k), \quad Q_k = Q(t_k)
  \]

Hybrid KF Loop

1. Enter initial state estimate and its error covariance \( \hat{x}_0, P_0 \)
2. Compute Kalman Gain
   \[
   K_k = P_k^{-1}H_k^T(H_kP_k^{-1}H_k^T + R_k)^{-1}
   \]
3. Linearize
   \[
   F = \left[ \frac{\partial f}{\partial x} \right]_{\hat{x}_{k+1}} \phi_k, H_k
   \]
4. Compute error covariance
   \[
   P_{k+1}^e = (I_n - K_kH_k)P_k^e + [F_kP_k^e + P_k^eF_k^T + G_kQ_kG_k^T]\Delta t
   \]
5. Measuremments \( \{ z_0, z_1, \ldots \} \)
6. Update estimate with measurement \( z_k \)
   \[
   \hat{x}_k = \hat{x}_k + K_k[z_k - h(x_k, t_k) - H_k\Delta \hat{x}_k]
   \]
7. Project Ahead:
   \[
   \hat{x}_{k+1} = f(\hat{x}_k)\Delta t + x_k
   \]
8. Compute error covariance
   \[
   P_{k+1}^e = (I_n - K_kH_k)P_k^e
   \]
9. State Estimates \( \{ \hat{x}_0, \hat{x}_1, \ldots \} \)

Example: Space Vehicle

- Vehicle near earth.
- Assume
  - Circular orbit.
  - Planar motion and measurements.
- Estimate vehicle position.
Coordinates of Space Vehicle

\[ \gamma = \text{angle between the earth's horizon and the local vertical} \]
\[ \alpha = \text{angle between the local vertical and a known reference line} \]
\[ R_e(r) = \text{radius of earth (vehicular orbit)} \]
\[ (r, \theta) = \text{polar coordinates of vehicular motion}. \]

Equations of Motion of a Particle

\[ [\mathbf{r}_1] = \begin{bmatrix} \cos(\theta) & \sin(\theta) \end{bmatrix} [\mathbf{i}] \]
\[ [\mathbf{\theta}_1] = \begin{bmatrix} -\sin(\theta) & \cos(\theta) \end{bmatrix} [\mathbf{i}] \]
\[ [\mathbf{i}] = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \end{bmatrix} [\mathbf{r}_1] \]
\[ [\mathbf{j}] = \begin{bmatrix} \sin(\theta) & \cos(\theta) \end{bmatrix} [\mathbf{\theta}_1] \]

\[ \mathbf{r}_1, \mathbf{\theta}_1 \] are the two unit vectors in the directions of the two body-fixed axes.

Velocity

\[ [\mathbf{r}_1] = \begin{bmatrix} \cos(\theta) & \sin(\theta) \end{bmatrix} [\mathbf{i}] \]
\[ \text{Changes in } r \text{ do not change the two unit vectors (zero coefficients of } \dot{r} \text{).} \]
\[ [\dot{\mathbf{r}}_1] = \begin{bmatrix} \partial \mathbf{r}_1 / \partial r & \partial \mathbf{r}_1 / \partial \theta \end{bmatrix} [\dot{\mathbf{r}}] \]
\[ = \dot{\theta} \begin{bmatrix} -\sin(\theta) & \cos(\theta) \end{bmatrix} [\mathbf{i}] = \dot{\theta} [\mathbf{\theta}_1] \]
\[ \mathbf{v} = \frac{d\mathbf{r}_1}{dt} = \dot{\mathbf{r}}_1 + r \frac{d\mathbf{r}_1}{dt} = [\mathbf{r}_1] [\mathbf{r}_1] \]

Acceleration

\[ \mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d}{dt} \{[\dot{\mathbf{r}}_1] [\mathbf{r}_1][\mathbf{\theta}_1] \} \]
\[ = \{[\ddot{\mathbf{r}}_1] [\mathbf{r}_1] [\mathbf{\theta}_1] + [\dot{\mathbf{r}}_1] \partial \mathbf{\theta}_1 / \partial \mathbf{r}_1 \} \]
\[ = \{[\ddot{\mathbf{r}}_1 - r \dot{\theta}^2] [\mathbf{r}_1] + [\dot{\mathbf{r}}_1] \dot{\theta} \} \]

Used to write equation of motion
Equations of Motion

Use polar coordinates (planar motion)
\[ \ddot{r} - r \dot{\theta}^2 + g \left( \frac{R_e}{r} \right)^2 = u_r(t) \]
\[ 2 \dot{r} \dot{\theta} + r \ddot{\theta} = u_\theta(t) \]

\( R_e \) = radius of earth.

\((u_r, u_\theta)\) = small random functions in \((r, \theta)\) directions = white noise

Nominal Trajectory

- Constant angular velocity \( \dot{\theta} = \omega_0 \)
  \[ \theta^* = \omega_0 t \]
- Fixed nominal radius \( r^* = R_0 \)
- Zero random perturbation

\[ \ddot{r} - r \dot{\theta}^2 + g \left( \frac{R_e}{r} \right)^2 = 0, \quad r \dot{\theta} + 2 \dot{r} \dot{\theta} = 0 \]
\[ -R_0 \omega_0^2 + g \left( \frac{R_e}{R_0} \right)^2 = 0 \Rightarrow \omega_0 = \sqrt{\frac{gR_e^2}{R_3^3}} = \sqrt{\frac{K}{R_0^3}} \]

Nonlinear State Equations

\[ x = [x_1 \ x_2 \ x_3 \ x_4]^T = [r \ \dot{r} \ \theta \ \dot{\theta}]^T \]
\[ \ddot{r} - r \dot{\theta}^2 + g \left( \frac{R_e}{r} \right)^2 = u_r(t), \quad 2 \dot{r} \dot{\theta} + r \ddot{\theta} = u_\theta(t) \]
\[ \dot{x}_1 = x_2 \]
\[ \dot{x}_2 = x_1 x_4^2 - \frac{K}{x_1^2} + u_r(t) \]
\[ \dot{x}_3 = x_4 \]
\[ \dot{x}_4 = -2x_2 x_4 / x_1 + u_\theta(t) / x_1 \]

Linearization

\[ \Delta \dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & 0 & 0 & \frac{\partial f_2}{\partial x_4} \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & 0 & \frac{\partial f_4}{\partial x_4} \end{bmatrix} \Delta x + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1/x_1^* \end{bmatrix} u \]

\[ f_2 = x_1 x_4^2 - K / x_1^2 + u_r(t) \]
\[ f_4 = -2x_2 x_4 / x_1 + u_\theta(t) / x_1 \]
\[ \partial(u_\theta/x_1) / \partial u_\theta = 1/x_1 \]
**Linearized State Equation**

- Substitute the nominal values

\[
x_1^* = r^* = R_0, \ x_2^* = 0, \ x_3^* = \theta^* = \omega_0 t, \ x_4^* = \omega_0
\]

\[
\Delta \dot{x} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
\frac{2K}{R_0^3} & 0 & 0 & 2R_0 \omega_0 \\
0 & 0 & 0 & 1 \\
0 & -\frac{2\omega_0}{R_0} & 0 & 0
\end{bmatrix} \Delta x + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u
\]

**Measurement Model**

\[
z = [z_1 \ z_2]^T = [\gamma \ \alpha]^T
\]

\[
= \begin{bmatrix} \sin^{-1} \left( \frac{R_e}{r} \right) & \alpha_0 - \theta \end{bmatrix}^T = \begin{bmatrix} \sin^{-1} \left( \frac{R_e}{x_1} \right) & \alpha_0 - x_3 \end{bmatrix}^T
\]

**Linearization**

- \(x_1^* = r^* = R_0\)

\[
z^* = h(x^*) = [\gamma^* \ \alpha^*]^T = \left[ \sin^{-1} \left( \frac{R_e}{x_1^*} \right) \ \alpha_0 - x_3^* \right]^T
\]

\[
\begin{bmatrix} \Delta z_1 \\ \Delta z_2 \end{bmatrix} = \begin{bmatrix} R_e \\ -\frac{R_e}{R_0 \sqrt{R_0^2 - R_e^2}} \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \Delta x + \nu
\]

**Extended KF: Corrector**

\[
\hat{x}_k^+ = \hat{x}_k^- + K_k [z_k - h(x_k^*) - H_k \Delta \hat{x}_k^-]
\]

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

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

- Minor change from the standard covariance filter.
Extended KF: Predictor

- Solve the nonlinear diff. eqs. for $\tilde{x}_{k+1}$
  (usually use an approximation)
  $\dot{x}_1 = x_2$
  $\dot{x}_2 = x_1 x_4^2 - K/x_1$
  $\dot{x}_3 = x_4$
  $\dot{x}_4 = -2x_2 x_4 / x_1$, IC $\tilde{x}_k^+$

- Error Covariance Matrix (discrete model)
  $P_{k+1}^- = \phi_k P_k^+ \phi_k^T + Q_k$

Iterated KF (Jazwinski, 1970)

- Iterate the correction equation $L$ times to reduce approximation errors.

- One additional corrector ($L = 2$) substantially improves results.

- After $L$ iterations: $\| \tilde{x}_{k,L}^+ - \tilde{x}_{k,L-1}^+ \| \leq \epsilon$

Block diagram

- Corrector 1: Compute $K(k), P^-(k), P^+(k)$ using $x^* = \tilde{x}_{k,0}^+$ $\Rightarrow \tilde{x}_{k,1}^+$

- Corrector 2: Compute $K(k), P^-(k), P^+(k)$ using $x^* = \tilde{x}_{k,1}^+$ $\Rightarrow \tilde{x}_{k,2}^+$

- Corrector $L$: Compute $K(k), P^-(k), P^+(k)$ using $x^* = \tilde{x}_{k,L-1}^+$ $\Rightarrow \tilde{x}_{k,L}^+$
Conclusion

- Discretization of nonlinear equations is difficult.
- KF is the optimal linear filter.
- The optimal filter may be nonlinear.
- Retaining more Taylor series terms is not always better than linearization.

References