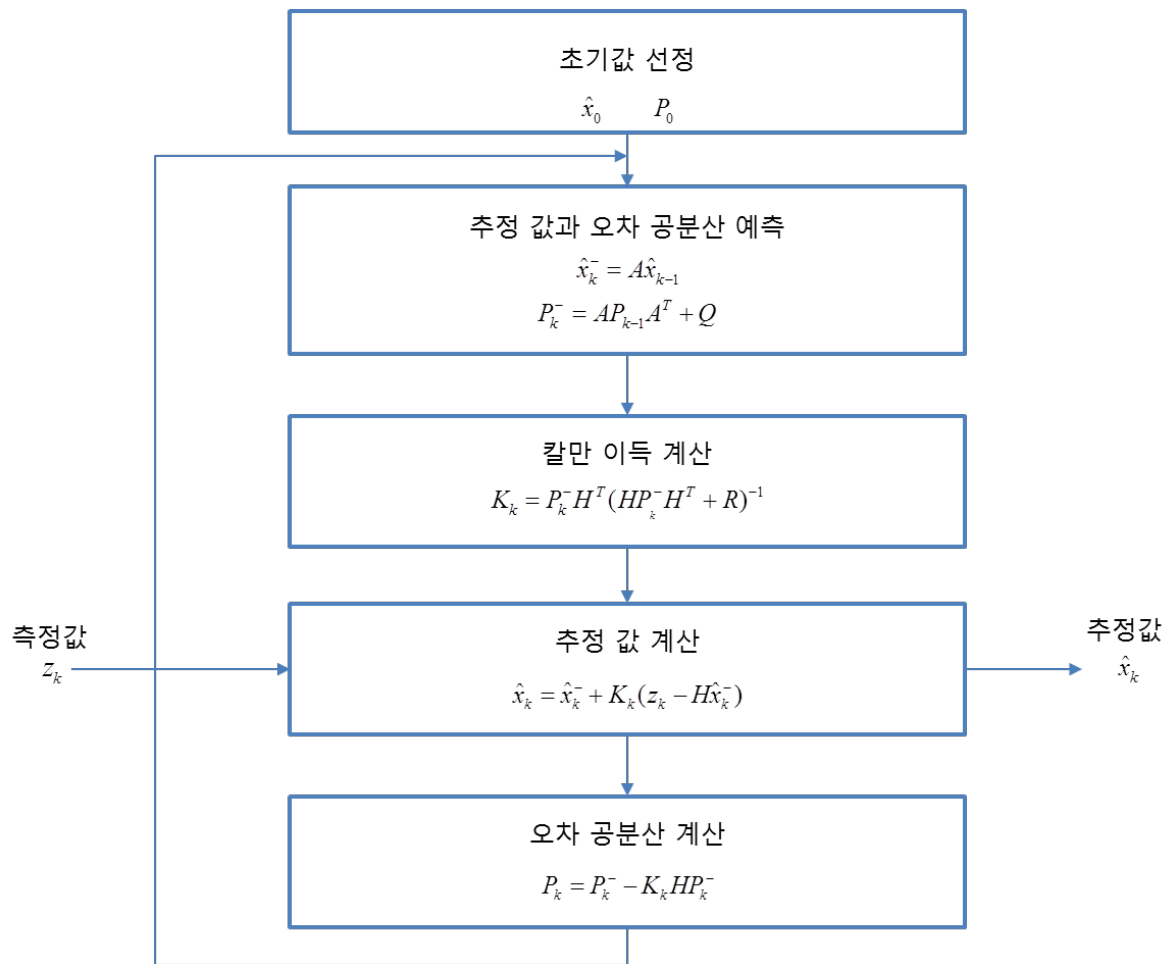


## 칼만필터(Kalman filter)



1) Estimation Process(추정 과정) → a posteriori estimate

Estimation Output(a posteriori estimate)

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) = (1 - K_k H)\hat{x}_k^- + K_k z_k$$

Kalman Gain

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

Kalman gain은 매 recursive 마다 새롭게 계산된다.

오차 공분산

$$P_k = P_k^- - K_k H P_k^-$$

오차 공분산은 Kalman filter의 추정값이 참 값에서 얼마나 차이가 나는지를 나타낸다. 오차 공분산은 추정값의 정확도에 대한 척도가 된다.  $P_k$  크면 추정 오차가 크고,  $P_k$  가 작으면 추정 오차가 작다.

오차 공분산을 다르게 표현하면 다음과 같다.

$$P_k = E\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\}$$

$x_k - \hat{x}_k \rightarrow$  참 값에서 추정 값을 뺀 차이 즉 오차를 의미

따라서 오차 공분산  $P_k$  는 추정 오차의 제곱을 평균한 것을 의미한다.

$$x_k \sim N(\hat{x}_k, P_k)$$

칼만 필터는 변수  $x_k$  의 추정 값에 대한 확률 분포를 따져서 가장 확률이 높은 값을 추정값으로 선택한다.

## 2) Prediction Process(예측 과정) → a priori estimate

예측 과정에서는 시각이  $t_k$  에서  $t_{k+1}$  로 변할 때 추정 값  $\hat{x}_k$  가 어떻게 변할지 추측한다. 즉, 현재 시각의 추정값이 다음 시각  $t_{k+1}$  에서는 어떤 값이 될지를 예측해 본다.

예측값 = a priori estimate

추정값 = a posteriori estimate

$$\hat{x}_{k+1}^- = A\hat{x}_k$$

$$P_{k+1}^- = AP_kA^T + Q$$

추정식을 다시 적어 보면 아래와 같다.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) = A\hat{x}_{k-1} + K_k(z_k - \underbrace{HA\hat{x}_{k-1}}_{\text{측정 값인 } z_k \text{ 의 예측 값을 의미}})$$



측정 값인  $z_k$  에 예측 값을 뺀 오차

예측 단계의 식은 비교적 간단하지만, 칼만 필터의 성능에 상당한 영향을 준다.

위의 식에서 보면 추정값의 성능에 큰 영향을 주는 요인은 예측 값의 정확성에 따라 다르다. 예측 값이 부정확하면 아무리 칼만 이득을 잘 선정한다고 해도 추정값이 부정확할 수 밖에 없다.

시스템 모델에 사용되는 A 와 Q 가 시스템과 많이 다르면 예측 값이 부정확하게 되고, 추정값도 엉뚱한 값이 된다.

### 3) 시스템 모델

$$x_{k+1} = Ax_k + w_k$$

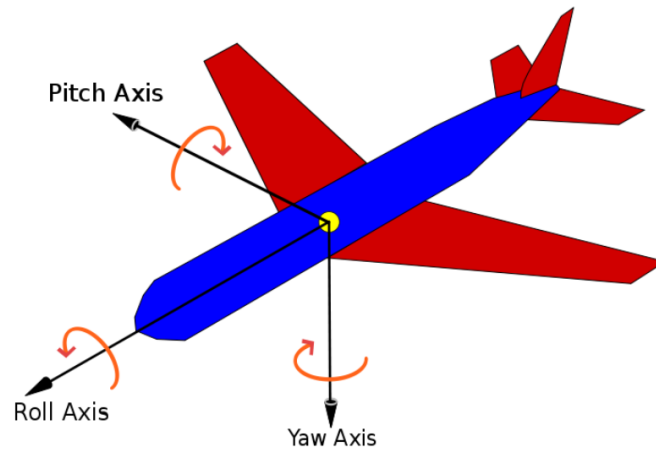
$$z_k = Hx_k + v_k$$

변 수	내 용	행 렬
$x_k$	상태 변수	N X 1 column vector
$z_k$	측정 값	M X 1 column vector
$A$	상태이전 행렬 시간에 따라 시스템이 어떻게 변화하는가를 나타내며 이것은 시스템의 운동 방정식	N X N 행렬
$H$	측정 값과 상태변수의 관계를 나타냄	M X N 행렬
$w_k$	잡음	N X 1 column vector
$v_k$	측정 잡음	M X 1 column vector

시스템 운동 방정식을 1차 미분 또는 차분 방정식으로 표현한 모델을 state space 모델이라고 한다.

## 수평자세 측정

자이로를 사용하여 측정된 각속도로부터 수평자세를 측정한다. 자이로로 측정한 각속도를 적분해서 오일러각을 구할 수 없다. 자이로는 오일러각의 변화율이 아니라 각속도를 측정한 값이기 때문이다.



Roll angle  $\phi$

Pitch angle  $\theta$

Yaw angle  $\varphi$

측정되는 각속도 정보로부터 자세정보를 구하기 위해서는 오일러각 변화율( $\dot{\phi}, \dot{\theta}, \dot{\varphi}$ )을 알아야 한다. 오일러각 변화율을 누적하게 되면 자세정보를 알 수 있게 된다. 측정된 각속도( $p, q, r$ )로부터 오일러각 변화율은 다음의 변환식에 의해 구할 수 있다.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & \sin \phi \\ 0 & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

오일러각의 변화율을 단위 시간으로 누적하여 오일러각을 다음과 같이 구할 수 있다.

$$\begin{bmatrix} \phi \\ \theta \\ \varphi \end{bmatrix}_{k+1} = \begin{bmatrix} \phi \\ \theta \\ \varphi \end{bmatrix}_k + \begin{bmatrix} \phi \\ \theta \\ \varphi \end{bmatrix}^* \Delta t$$

가속도를 이용한 자세 결정

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} 0 & w & -v \\ -w & 0 & u \\ v & -u & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + g \begin{bmatrix} \sin \theta \\ -\cos \theta \sin \phi \\ -\cos \theta \cos \phi \end{bmatrix}$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \text{ 가속도계로 측정한 가속도}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \text{ 이동 가속도}$$

$u, v, w$  이동 속도

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \text{ 각속도}$$

$g$  중력가속도

구해야 하는 수평자세에 대한 정보는 이동속도와 이동 가속도를 알면 충분히 구할 수 있다. 하지만 저급의 장치에서는 측정 불가능하다. 일정한 속도로 직진하는 경우 이동가속도는 0이며 일정한 자세를 유지하기 때문에 각속도  $p, q, r$ 도 0이 된다.

$$u = v = w = 0$$

$$p = q = r = 0$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = g \begin{bmatrix} \sin \theta \\ -\cos \theta \sin \phi \\ -\cos \theta \cos \phi \end{bmatrix}$$

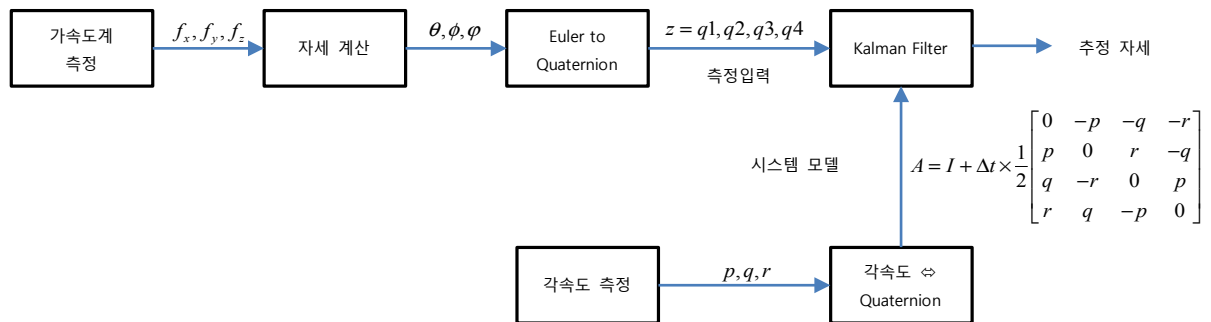
$$\theta = \sin^{-1} \left( \frac{f_x}{g} \right)$$

$$\phi = \sin^{-1} \left( \frac{f_y}{g \cos \theta} \right)$$

따라서 측정된 가속도로부터 수평자세 정보인 roll angle( $\phi$ )과 pitch angle( $\theta$ ) 공식을 유도해 낼 수 있다.



## 센서 융합(sensor fusion)을 통한 자세 결정



가속도계로 구한 자세는 시간이 지나도 오차가 커지지 않고 일정범위로 제한되는 장점이 있다. 반면 자이로는 자세변화는 잘 감지하지만 시간이 지나면 오차가 누적되어 발산하는 문제가 있다. 즉, 단기적으로는 자이로 자세가 더 낫지만, 중장기적으로는 가속도계의 자세가 더 낫다고 말할 수 있다.

칼만 필터의 설계는 시스템 모델을 만드는 것인데, 가속도계와 자이로를 사용하여 구하고자 하는 것은 자세 정보( $\theta, \phi, \varphi$ )이다.

$$x_{k+1} = Ax_{k+1} + w_k$$

$$\begin{bmatrix} q1 \\ q2 \\ q3 \\ q4 \end{bmatrix}_{k+1} = I + \Delta t \times \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q1 \\ q2 \\ q3 \\ q4 \end{bmatrix}_k$$

시스템 모델을 구하기 위해 측정된 각속도 정보 p, q, r을 사용하는 quaternion은 위와 같다. 즉, system matrix A는 다음과 같으며, system matrix A는 각속도 측정이 update될 때마다 매번 다시 update된다.

$$A = I + \Delta t \times \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}$$

가속도계 정보를 기반으로 구한 자세정보( $\theta, \phi, \varphi$ )를 Kalman filter의 측정 입력으로 사용한다. 이렇게 하기 위해서는 자세정보를 quaternion으로 변경해야 하는데 아래와 같이 잘 알려진 방법을 사용한다.

$$\begin{bmatrix} q1 \\ q2 \\ q3 \\ q4 \end{bmatrix} = \begin{bmatrix} \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\varphi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\varphi}{2} \\ \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\varphi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\varphi}{2} \\ \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\varphi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\varphi}{2} \\ \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\varphi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\varphi}{2} \end{bmatrix}$$

위의 표현에서 보면 q1, q2, q3, q4에 대한 모든 quaternion이 필요하다. 따라서 측정 모델에 대한 H matrix는 다음과 같이 단위 행렬이 된다.

$$z_k = Hx_k + v_k$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

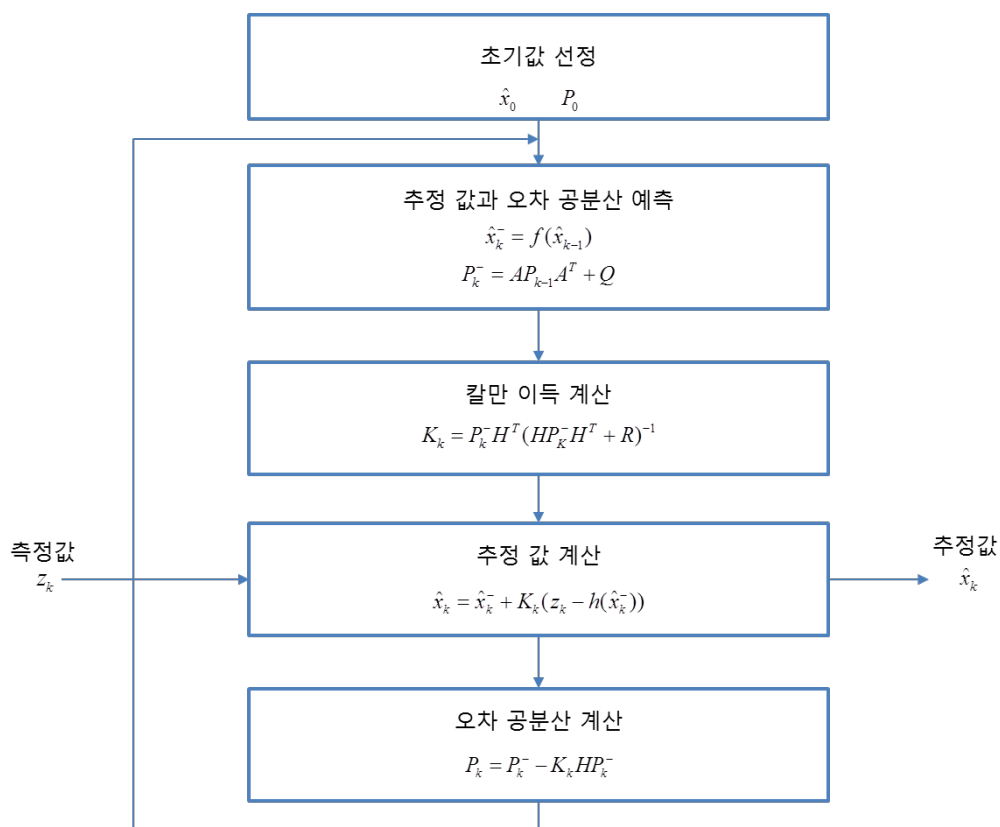
Q matrix와 R matrix는 경험적으로 다음과 같이 설정 할 수 있다.

$$Q=\begin{bmatrix}0.0001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001\end{bmatrix}$$

$$R=\begin{bmatrix}10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10\end{bmatrix}$$

## 확장 칼만 필터(extended Kalman filter)

칼만 필터의 경우 선형 시스템에 대해 지연 없이 뛰어난 노이즈 제거 성능을 보인다. 특히, 시스템을 알고 있기 때문에 측정값의 미분 또는 적분값을 구하는 경우에도 시스템 모델에 파라미터를 포함시키게 될 경우 거의 정확하게 구할 수 있다. 그러나 시스템이 선형일 경우에만 사용할 수 있기 때문에 한계를 가진다. 이를 위해 비선형 시스템을 특정 구간에서 선형화 하여 칼만 필터를 사용하기도 하지만 모델링 시 지정한 구간에서만 의미 있는 결과를 도출 할 수 있다. 이를 해결하고자 비선형 시스템에서 칼만 필터를 사용할 수 있도록 만든 것이 확장 칼만 필터이다. 칼만 필터는 선형시스템에 적용가능하나 대부분의 실제 시스템은 비선형 시스템이다. 따라서 최초로 칼만 필터가 적용된 분야도 확장 칼만 필터를 이용한 비선형 시스템이었다. 확장 칼만 필터의 단점은 알고리즘이 발산할 수 있다는 위험을 가지고 있는 것이다. 확장 칼만 필터 알고리즘의 전체는 칼만 필터와 동일 하지만 단지 차이가 나는 점은 예측과 추정단계의 식이 다르다는 점과 시스템 행렬을 구하기 위해 편미분방식의 자코비안을 사용한다는 점이다.



확장 칼만 필터는 칼만 필터와 동일한 과정을 따른다. 추정값과 오차 공분산 예측단계와 추정값 계산단계에서 A와 H 행렬이  $f(\bullet)$ 와  $h(\bullet)$ 로 바뀐 것 이외에는 완벽하게 칼만 필터와 동일하다. 비선형 시스템의 모델을 통해서 보면 다음과 같은 차이점이 있다.

비선형 시스템 모델

$$x_{k+1} = f(x_k) + w_k$$

$$z_k = h(x_k) + v_k$$

확장 칼만 필터에서는 선형시스템과 달리 상태변수  $x_k$ 와 계수 A 또는 H를 따로 분리하여 쓰지 못 한다. 아래의 표에 선형 시스템과 비선형 시스템의 차이를 나타내었다.

선형시스템	비선형 시스템
$Ax_k$	$f(x_k)$
$Hx_k$	$h(x_k)$
Kalman Filter	Extended Kalman Filter
$\hat{x}_k^- = Ax_{k-1}$	$\hat{x}_k^- = f(\hat{x}_{k-1})$
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$	$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-))$

확장 칼만 필터의 예측과 추정단계에서는 비선형 시스템 모델인  $f(\hat{x}_{k-1})$ 와  $h(\hat{x}_k^-)$ 을 그대로 사용한다. 확장 칼만필터 알고리즘에서 반복적으로 사용되는 시스템 행렬 A와 H는 시스템 모델  $f(\bullet)$ 와  $h(\bullet)$ 를 사용하여 자코비안(Jacobian)을 구해서 계산 가능하다. 또한 Q와 R 행렬은 칼만 필터의 그것과 동일한 것을 적용 가능하다.

$$A = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_k}$$

$$H = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_k^-}$$

만약 시스템 모델이 다음과 같이 표현 할 수 있을 경우 자코비안 행렬 A는 다음과 같이 구할 있다.

시스템 모델

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + w$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} ax_1 + bx_2 + cx_3 \\ dx_1 + ex_2 + fx_3 \\ gx_1 + hx_2 + ix_3 \end{bmatrix} + w$$

자코비안을 구하기 위한  $f(\bullet)$ 를 다음과 같이 정의할 수 있다.

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} ax_1 + bx_2 + cx_3 \\ dx_1 + ex_2 + fx_3 \\ gx_1 + hx_2 + ix_3 \end{bmatrix}$$

$f(\bullet)$ 에 대한 자코비안 행렬은 다음과 같이 편미분으로 구할 수 있다.

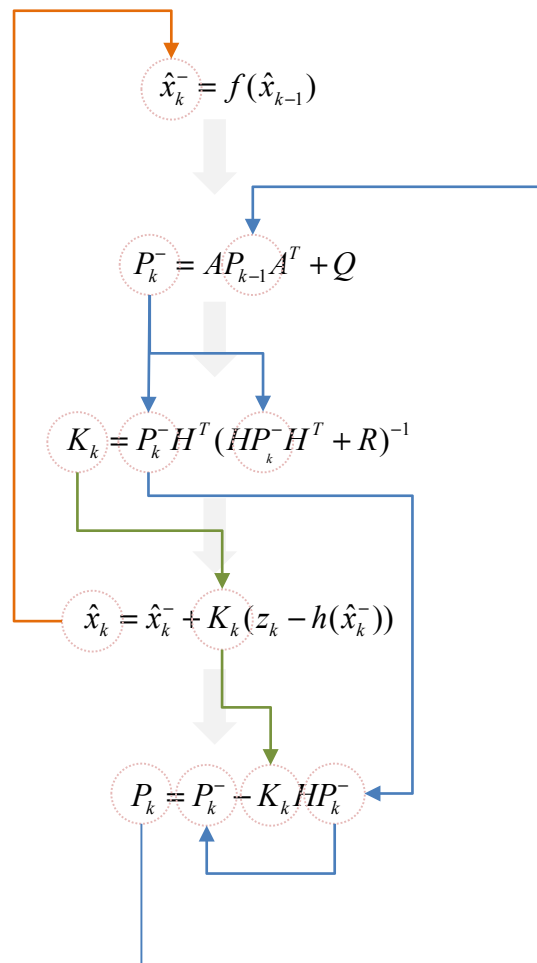
$$Jacobian A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}$$

자코비안 행렬의 일반적인 형태는 다음과 같이 정의 할 수 있다.

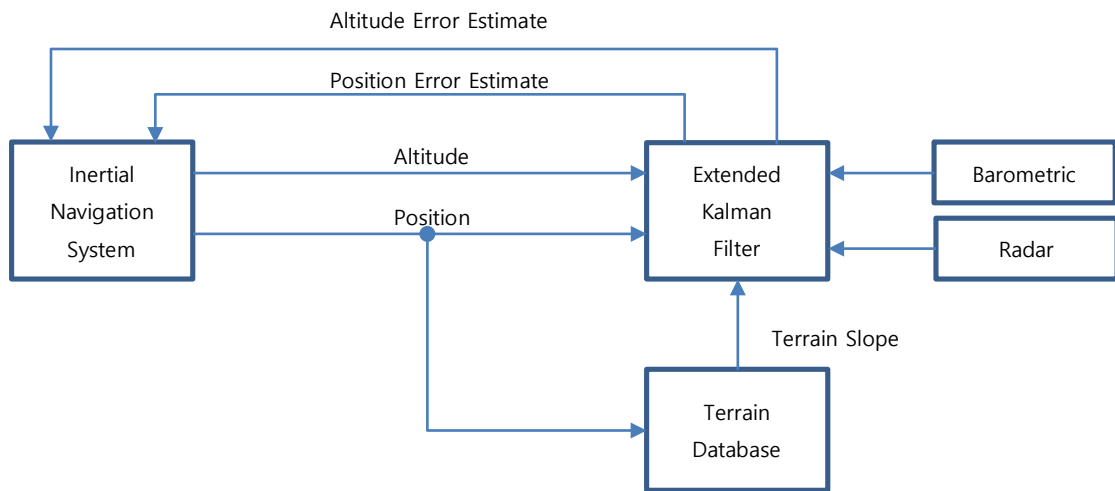
$$f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)$$

$$Jacobian = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

확장 칼만 필터는 미리 지정한 선형화 기준점을 사용하지 않고 추정값( $\hat{x}_k$ )을 기준으로 삼는다. 이것이 시스템의 실제 상태와 가장 가까운 값이라 보고 이 값을 기준으로 선형 모델을 계산하는 것이다. 따라서 선형화 기준점을 결정하기 어려운 시스템에 적합하다. 하지만 확장 칼만 필터에서는 자코비안으로 구한 행렬의 초기 값을 얻을 수 없기 때문에 초기값을 임의의 추정값으로 최초에 한번 주어야 하는데, 이때 초기 값에 따라 결과가 발산 할 수 있다.



## Extended Kalman Filter for Terrain Reference Navigation



매 measurement update step에서 measurement들은 저장된 DTED와 비교되고 그리고 vehicle의 position을 추측한다. TRN을 위한 Kalman filter를 구현하기 위해서는 measurement matrix는 local slope linearization 기술을 이용하여 선형화 되어야 한다. Error 추정치는 primary navigation system에 되먹임 되어야 하고, navigation system의 error를 compensate해야 한다.

### 1) System Model

Nonlinear terrain referenced navigation system의 state space representation은 다음과 같다.

$$x_{t+1} = f(t, x_t) + w_t$$

$$y_t = h(t, x_t) + v_t$$

$x_t$ : state vector

$$x = [x, y, h, v_x, v_y]^T$$

$x$  : latitude

$y$  : longitude



$h$  : height

$v_x$  : velocity along x

$v_y$  : velocity along y

$y_t$ : measurement variable

$w_t, v_t$  : mutually independent noise

EKF 알고리즘을 적용시키기 위해서  $f(t, x_t)$ 와  $h(t, x_t)$ 는 estimate  $\hat{x}$  인근에서 선형화 되어야 한다.

$$F_t = \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_t} \quad H_t = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_t}$$

Error state equation은 다음과 같이 유도된다.

$$\delta x_{t+1} = F_t \cdot \delta x_t + w_t$$

$$\delta x = \begin{bmatrix} \delta x \\ \delta y \\ \delta h \\ \delta v_x \\ \delta v_y \end{bmatrix}, \quad F_t = \begin{bmatrix} 1 & 0 & 0 & T & 0 \\ 0 & 1 & 0 & 0 & T \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad w_t = \begin{bmatrix} w_x \\ w_y \\ w_h \\ w_{v_x} \\ w_{v_y} \end{bmatrix}$$

T는 Sampling Time을 의미

Measurement models은 다음과 같이 주어진다.

$$y_t = h_t^{ralt} - (h_t^{NAV} - h_t^{DB}) = H_t \cdot \delta x_t + v_t$$

$h_t^{ralt}$  : height of radio altimeter

$h_t^{NAV}$  : height of navigation system

$h_t^{DB}$  : height of DTED

Measurement matrix  $H_t$ 를 구하기 위하여 local slope linearization technique를 적용하면 다음과 같다.

$$H_t = \begin{bmatrix} \frac{\partial h_t^{DB}}{\partial \delta x} & \frac{\partial h_t^{DB}}{\partial \delta y} & 1 & 0 & 0 \end{bmatrix}$$

## 2) Extended Kalman Filter

### 1. State propagation step

#### a. Prior state

$$\delta x_{t+1|t} = F \cdot \delta x_{t|t}$$

#### b. Prior covariance matrix(using law of error propagation)

$$P_{t+1|t} = F_t P_{t|t} F_t^T + Q_t$$

### 2. Measurement update step

#### a. Kalman gain matrix

$$K_{t+1|t} = P_{t+1|t} H_{t+1}^T [H_{t+1} P_{t+1|t} H_{t+1}^T + R_{t+1}]^{-1}$$

#### b. Posterior estimate

$$\delta x_{t+1|t+1} = \delta x_{t+1|t} + K_{t+1|t} [y_{t+1} - H_{t+1} \delta x_{t+1|t}]$$

#### c. Covariance matrix

$$P_{t+1|t+1} = [I - K_{t+1|t} H_{t+1}] P_{t+1|t}$$

추가 사항

다음과 같은 비선형 시스템과 측정 모델을 가정한다.

$$X = f(X, u) + w \quad (1)$$

$$Z = h(X, u) + v \quad (2)$$

$w, v$ 는 시스템의 잡음과 측정 시스템의 잡음을 각각 나타낸다. 제어 입력  $u$ 는 주어지는 값이라 가정한다.

$$X = f(X, u) + w$$

식(1)을 만족하는 궤적  $X_0$ 로부터 다음과 같이 나타낼 수 있다.

$$X_0 = f(X_0, u)$$

따라서  $X$ 는  $X_0$ 와 섭동량(perturbation)의 변화  $x$ 의 합의로 나타낼 수 있다.

$$X = X_0 + x \quad (3)$$

$$X = f(X_{0+x}, u) + w \approx f(X_0, u) + \left. \frac{\partial f}{\partial X} \right|_{X_0, u} x + w \quad (4)$$

(3), (4)로부터 다음과 같이 선형화된  $x$ 를 구할 수 있다.

$$X_0 + x = f(X_0, u) + \left. \frac{\partial f}{\partial X} \right|_{X_0, u} x + w$$

$$x = Fx + w, \text{ where } F = \left. \frac{\partial f}{\partial X} \right|_{X_0, u}$$

측정 모델 식(2)에 대해서도 선형화를 진행한다.

$Z_0 = h(X_0, u)$ 로부터 식(2)를 다음과 같이 표현 할 수 있다.

$$Z = Z_0 + z$$

$$Z = Z_0 + z = h(X_0, u) + \left. \frac{\partial h}{\partial X} \right|_{X_0, u} x + v \tag{6}$$

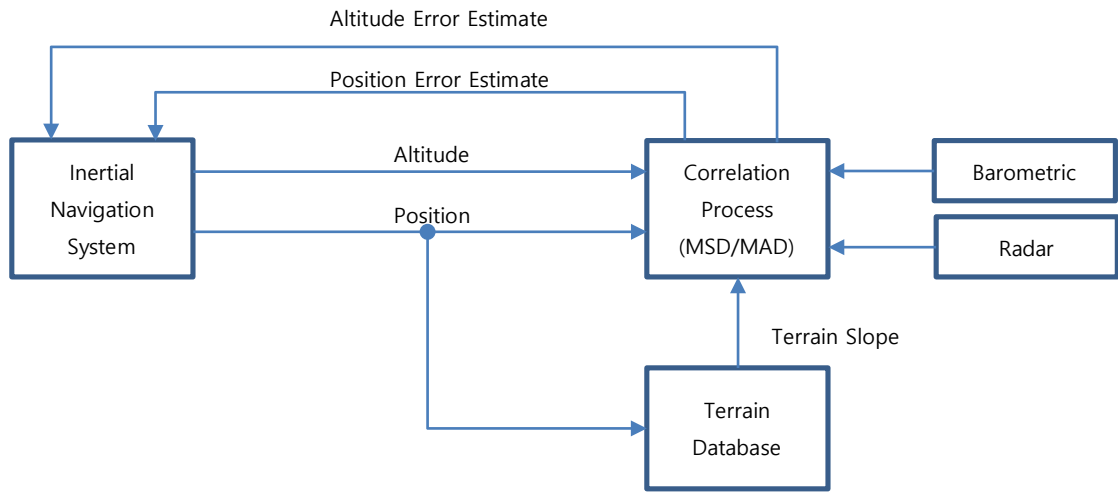
식(6)을 간소화 하면 다음과 같은 선형화된 측정 모델을 구할 수 있다.

$$z = Hx + v, \text{ where } H = \left. \frac{\partial h}{\partial X} \right|_{X_0, u}$$

비선형 모델	선형 모델
$X = f(X, u) + w$	$x = Fx + w \text{ } (F = \left. \frac{\partial f}{\partial X} \right _{X_0, u})$
$Z = h(X, u) + v$	$z = Hx + v \text{ } (H = \left. \frac{\partial h}{\partial X} \right _{X_0, u})$

EKF를 DBRN 추정알고리즘에 적용하기 위하여 추정모델을 수립해 보면 시스템 모델식은 아래와 같다.

## Profile Matching based TRN system



연속적인 Terrain height measurement를 DTEM(digital Terrain Elevation Model)과 비교함으로써 위치를 찾아내는 방식을 Profile Matching based TRN system이라 한다.

통상적으로 연속적인 Terrain height measurement와 주어진 DTEM과의 correlation을 구함으로써 가능

Profile Matching based TRN system의 대표적인 시스템은 TERrain Contour Matching(TERCOM)이 있다.

TERCOM은 Batch processing 즉, 일정한 measurement를 측정한 후 일괄적으로 DTEM과 Correlation을 구해 위치를 찾아 내는 방식을 따른다.

대표적인 Correlation 알고리즘은 MAD(Mean Absolute Difference)와 MSD(Mean Square Difference)

$$MAD_{n,m} = \frac{1}{N} \sum_{i=1}^N |h_{meas}(i) - h_{DB}(n+i, m)|$$

$h_{meas}$  : series of terrain height measurement

$h_{DB}$  : DTEM height

$MAD_{n,m}$  : mean absolute difference in the  $n^{th}$  and  $m^{th}$  terrain height profile

$N$  : number of sample

Position update는 MAD가 가장 작은  $n$ 과  $m$ 을 찾아 업데이트 한다.

$$P_{MAD}(n, m) = P\left(\arg \min_{n, m} MAD_{n, m}\right)$$

$\arg \min_{n, m}$  : index( $n, m$ ) having minimum value of  $MAD_{n, m}$

MSD는 MAD유사하지만 조금더 수학적으로 복잡하다.

$$MSD_{n, m} = \frac{1}{N} \sum_{i=1}^N \{h_{meas}(i) - h_{DB}(n+i, m)\}^2$$

$$P_{MSD}(n, m) = P\left(\arg \min_{n, m} MSD_{n, m}\right)$$

Profile Matching based TRN system은 평탄지형에서 보다는 지형의 특성 변화가 많은 곳에서 그 검출 성능이 우수하다. 즉, 지형의 험준도가 높을수록 TRN의 정확도는 상승한다.

Terrain의 특성을 나타내는 index로  $\sigma_T$ 와  $\sigma_Z$ 가 있다.

$\sigma_T$  : Terrain height의 standard deviation로 평균 height( $\bar{h}$ )와 측정된 height( $h_i$ )의 차이로부터 지형의 험준도를 나타낸다.

$$\sigma_T = \sqrt{\frac{1}{N} \sum_{i=1}^N (h_i - \bar{h})^2}$$

$$\bar{h} = \frac{1}{N} \sum_{i=1}^N h_i$$

$\sigma_Z$ : 인접한 sample들의 간의 변화에 대한 standard deviation으로 지형의 유일성을 나타낸다.

$$\sigma_Z = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta h_i - \Delta \bar{h})^2}$$

$$\Delta h_i = h_i - h_{i+1}$$

$$\Delta \bar{h} = \frac{1}{N-1} \sum_{i=1}^{N-1} \Delta h_i$$