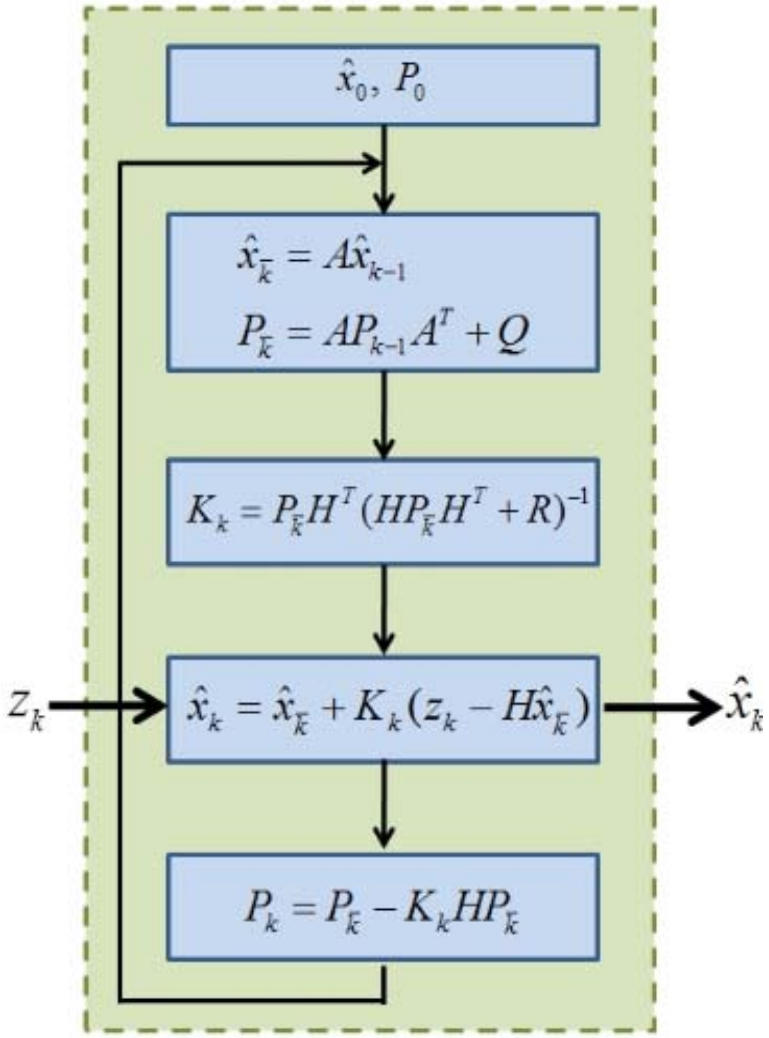
Extended Kalman filter

2019.04.04

차례

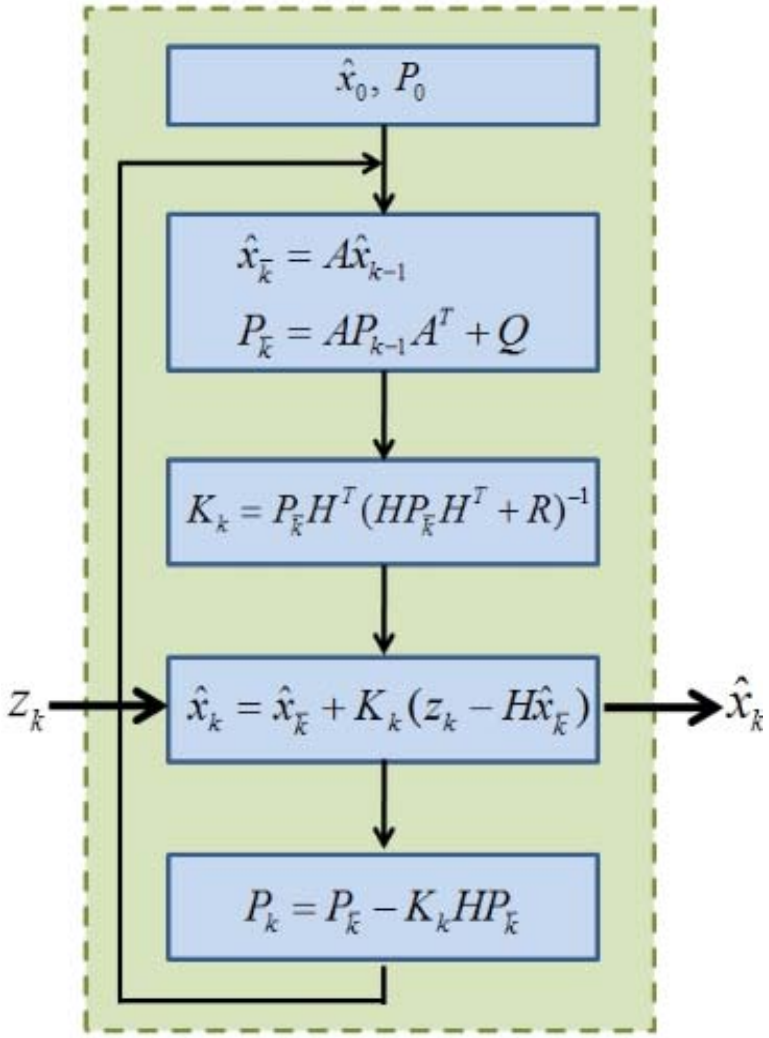
1. Linearization Kalman Filter

1. Linearization Kalman Filter



Kalman Filter algorithm

1. Linearization Kalman Filter



Kalman Filter algorithm

$x_{k+1} = Ax_k + w_k$ $Q = w_k$의 공분산 행렬

$z_k = Hx_k + v_k$ $R = v_k$의 공분산 행렬

x_k는 상태 변수

z_k는 측정값

A는 상태전이행렬

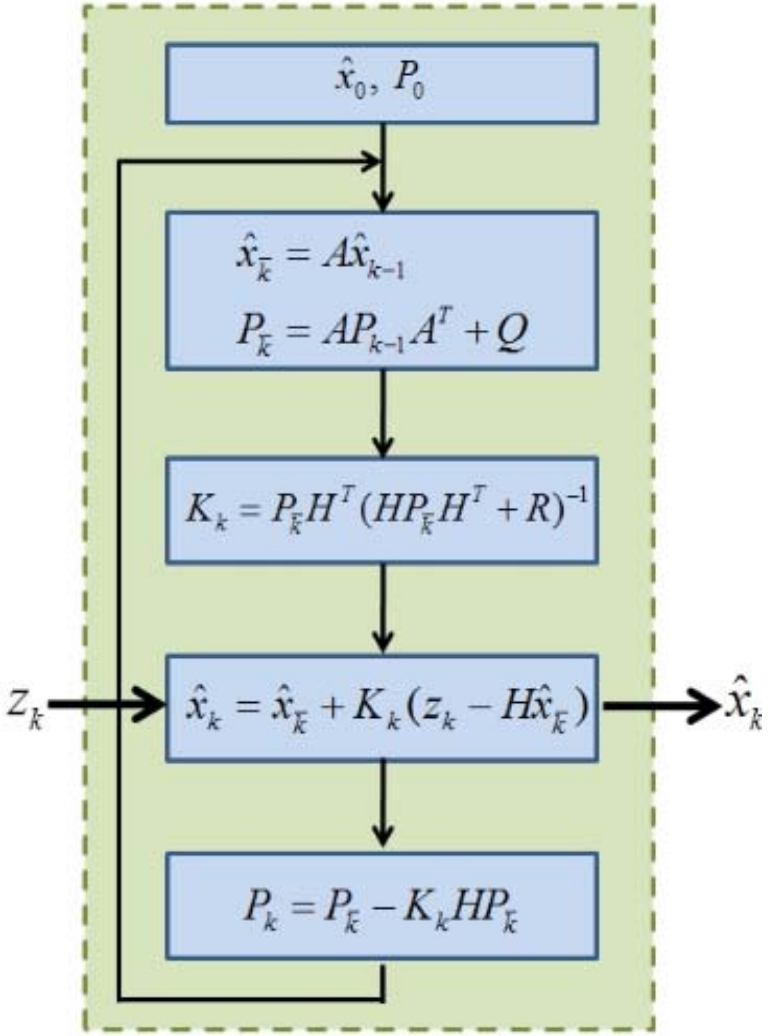
H는 mXn 행렬

w_k는 잡음

w_k는 측정 잡음

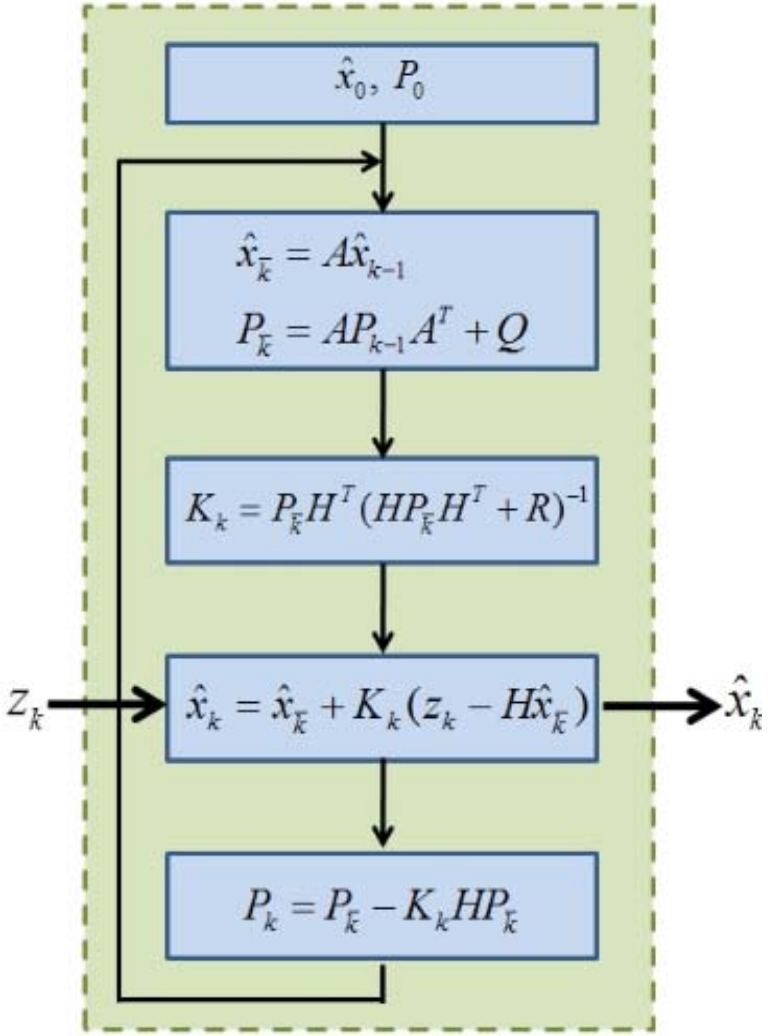
2. Extended Kalman Filter

2. Extended Kalman Filter

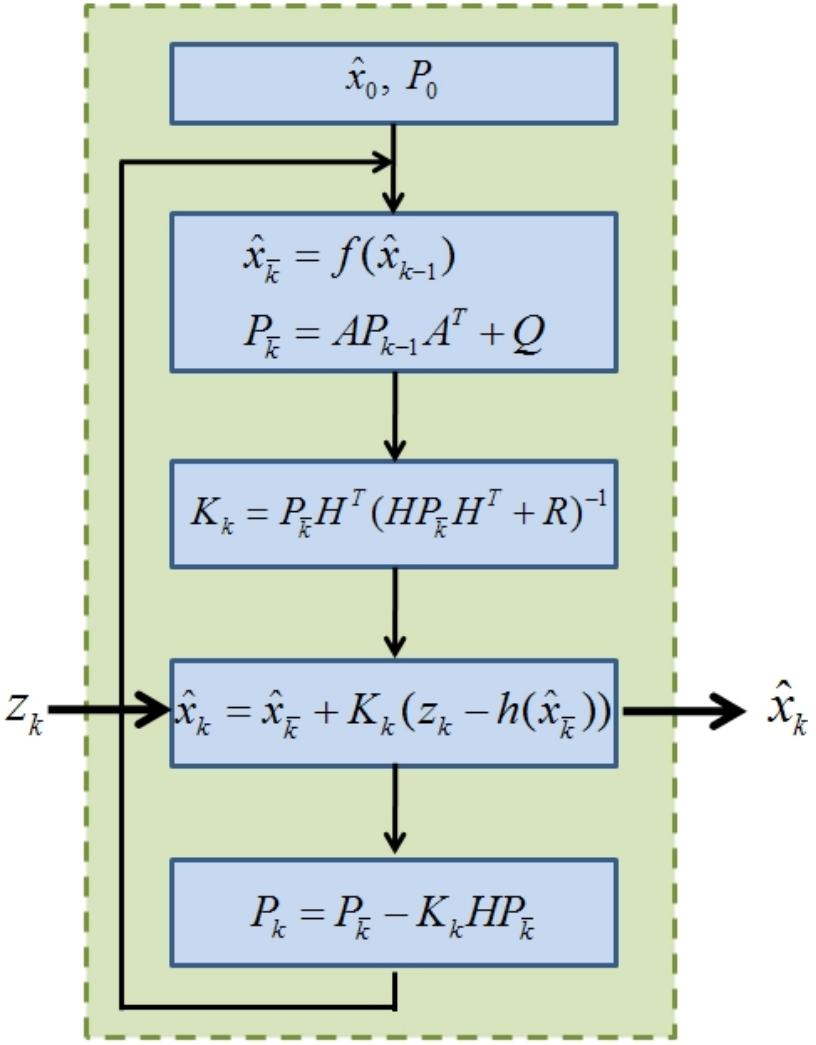


Kalman Filter algorithm

2. Extended Kalman Filter

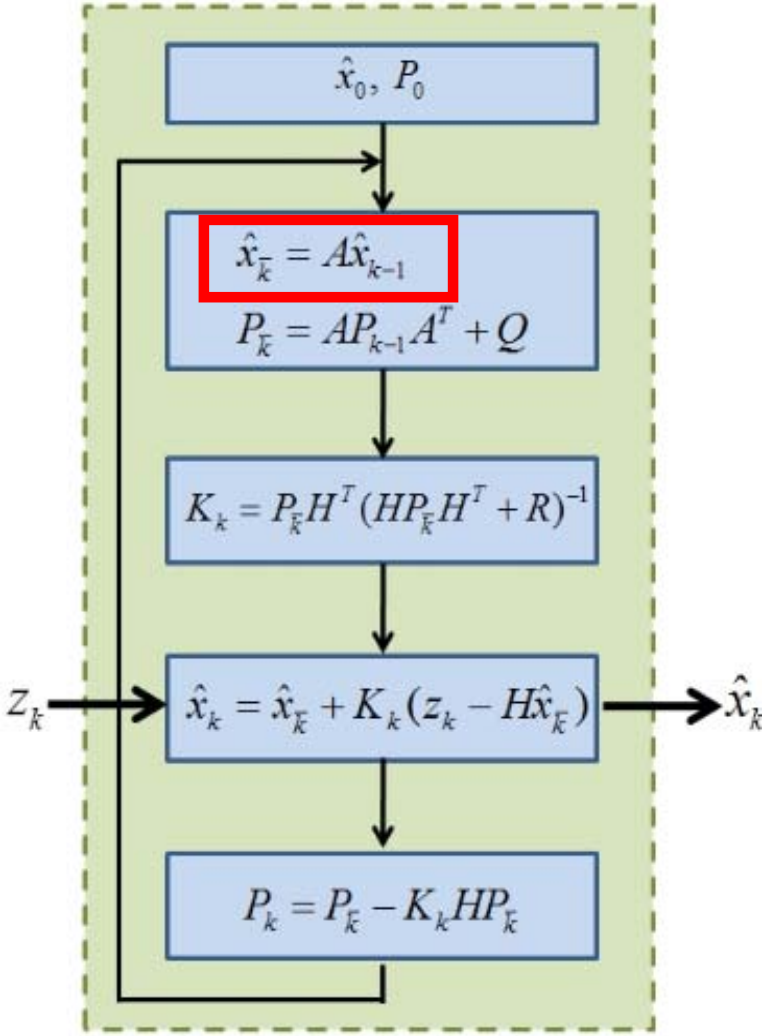


Kalman Filter algorithm

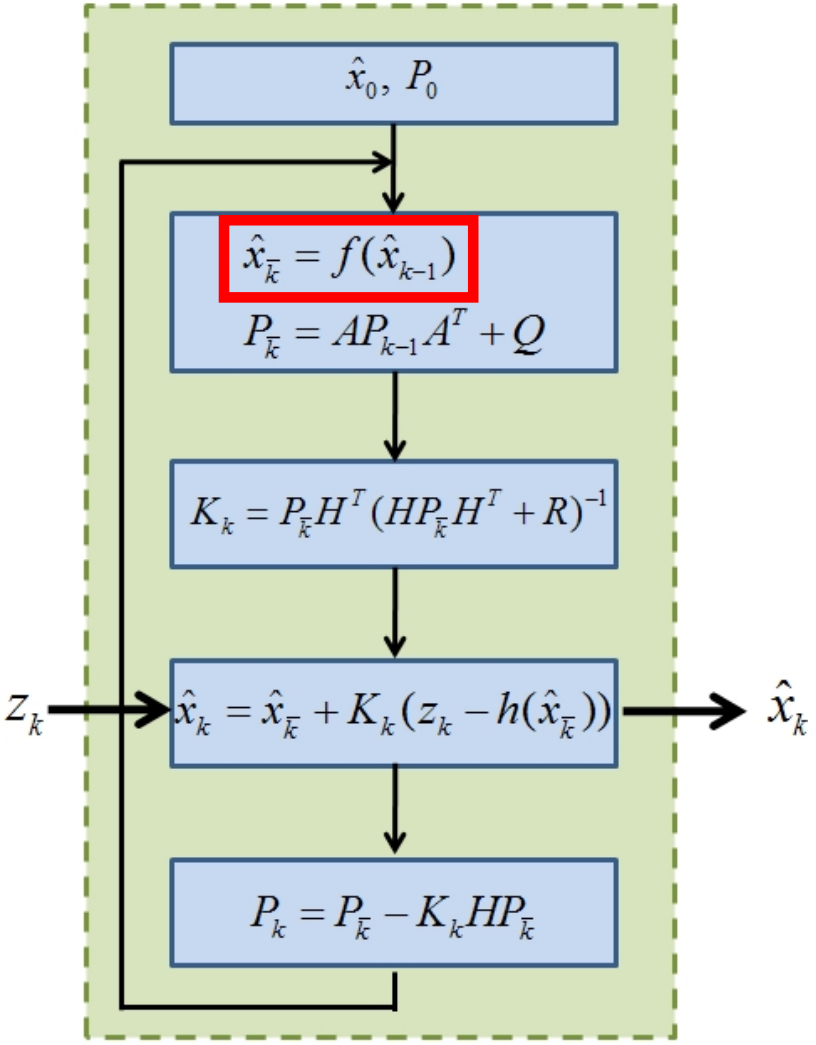


EKF algorithm

2. Extended Kalman Filter

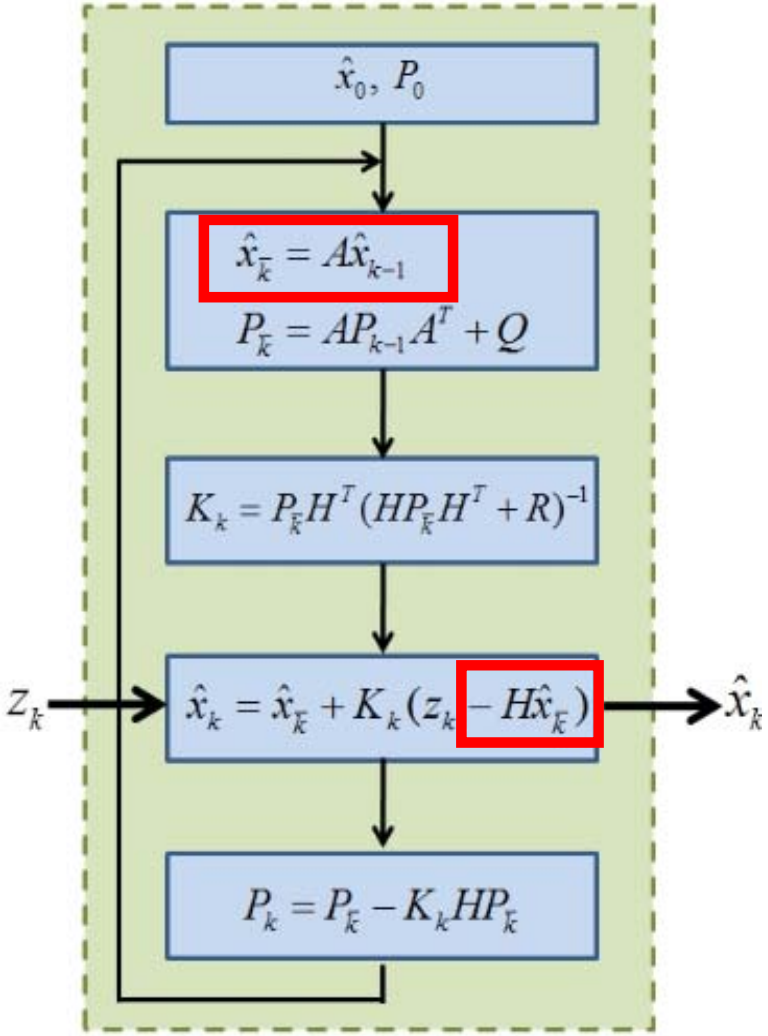


Kalman Filter algorithm

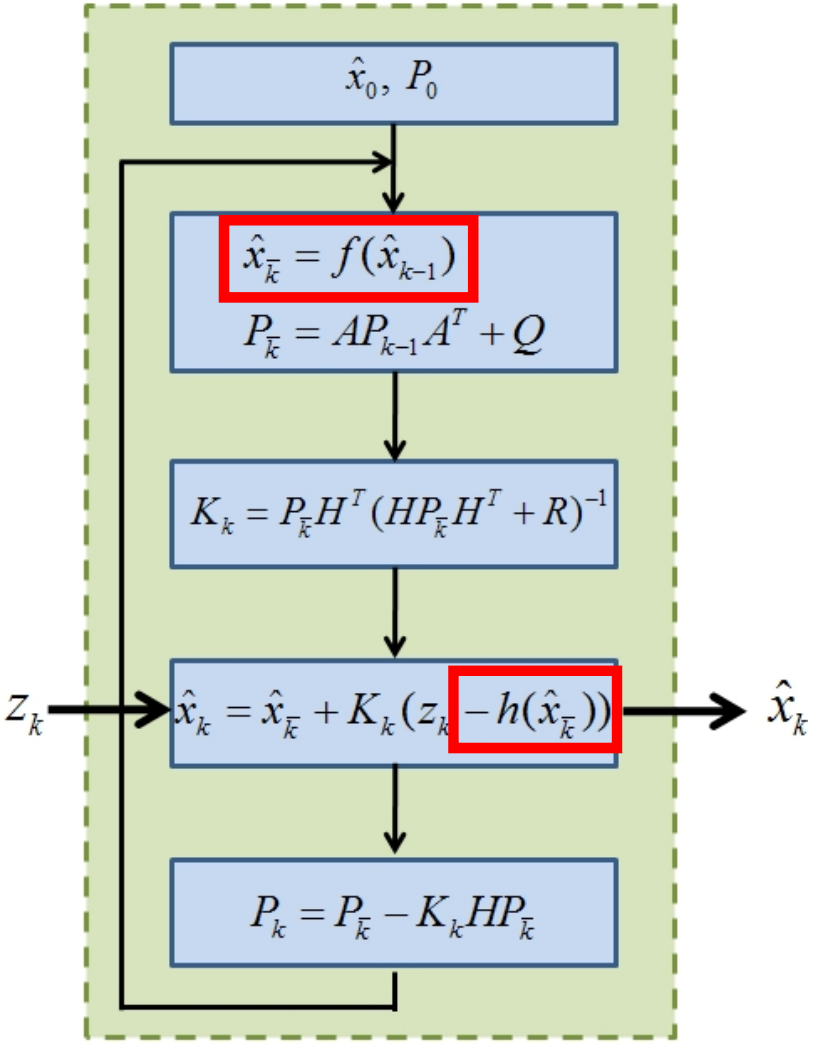


EKF algorithm

2. Extended Kalman Filter



Kalman Filter algorithm

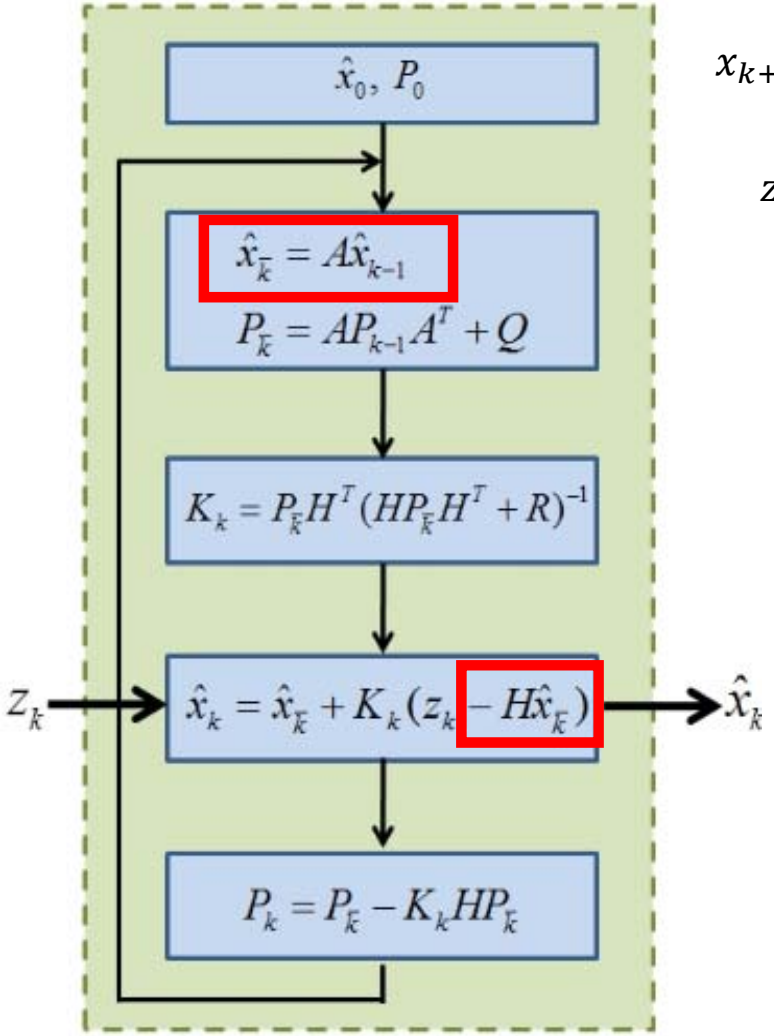


EKF algorithm

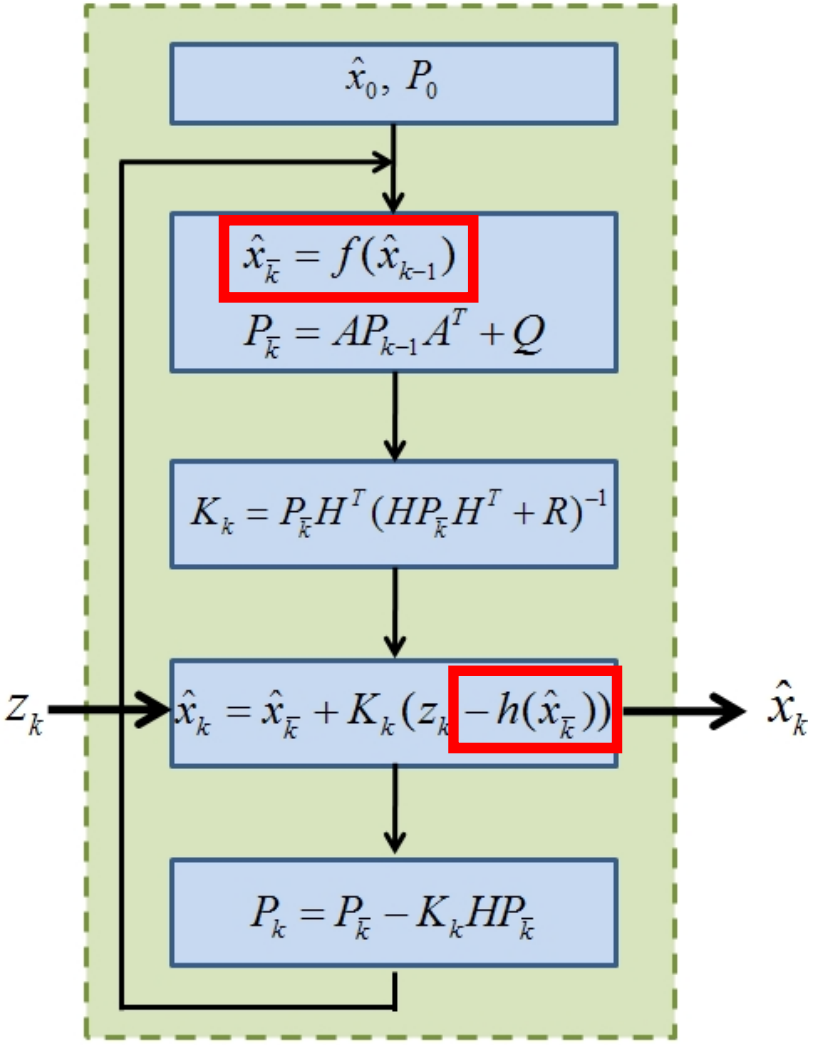
2. Extended Kalman Filter

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

$$z_k = Hx_k + v_k$$



Kalman Filter algorithm

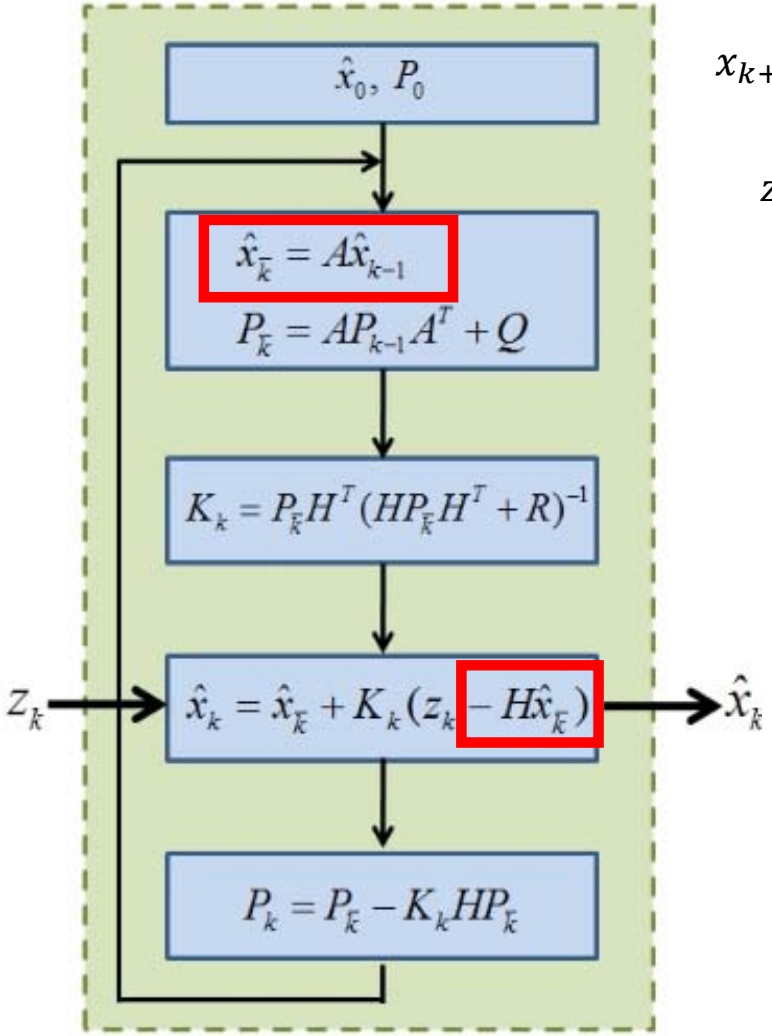


EKF algorithm

2. Extended Kalman Filter

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

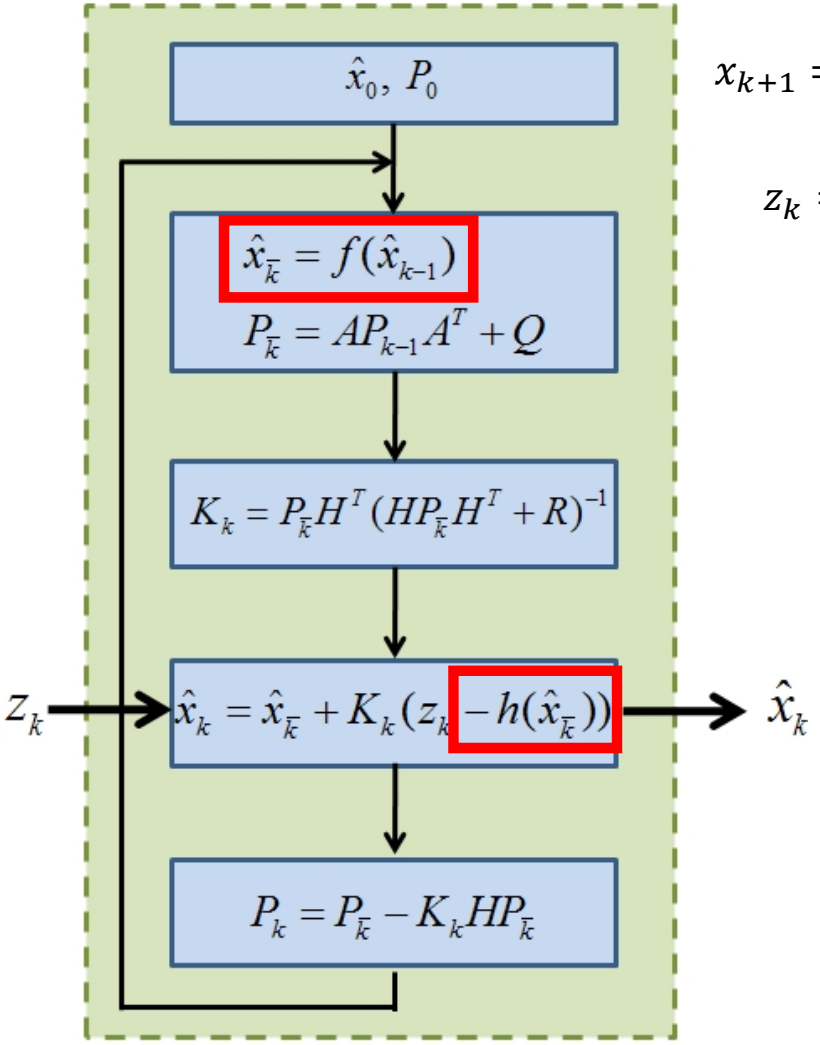
$$z_k = Hx_k + v_k$$



Kalman Filter algorithm

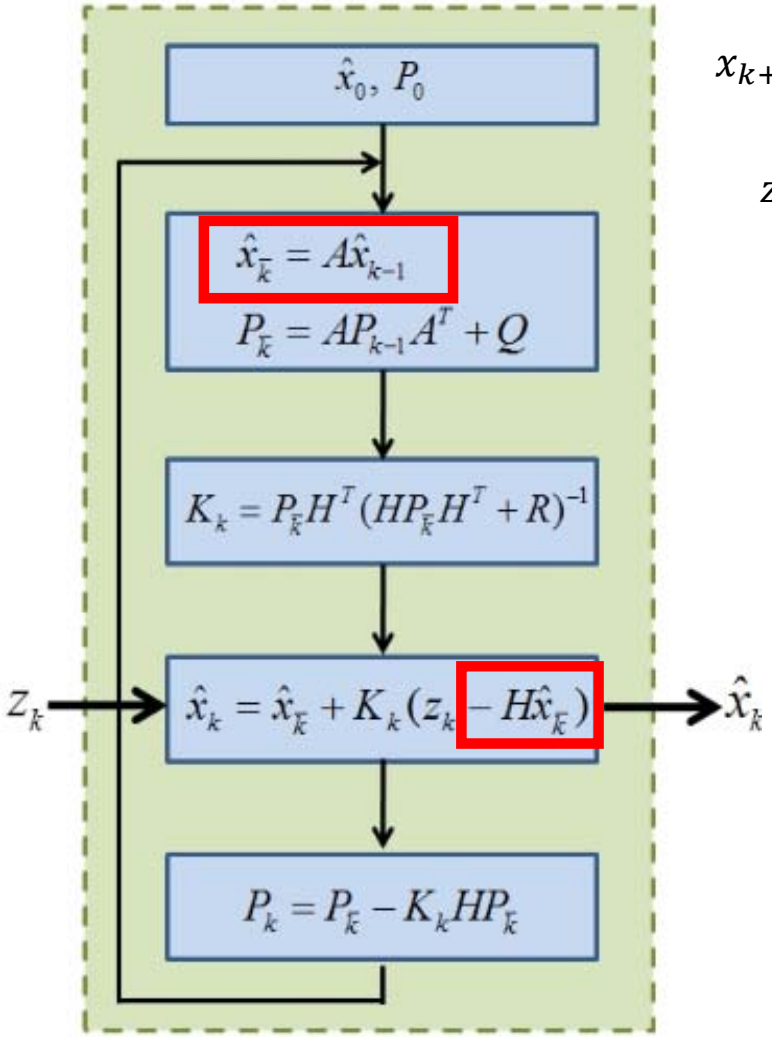
$$x_{k+1} = f(x_k) + w_k$$

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



EKF algorithm

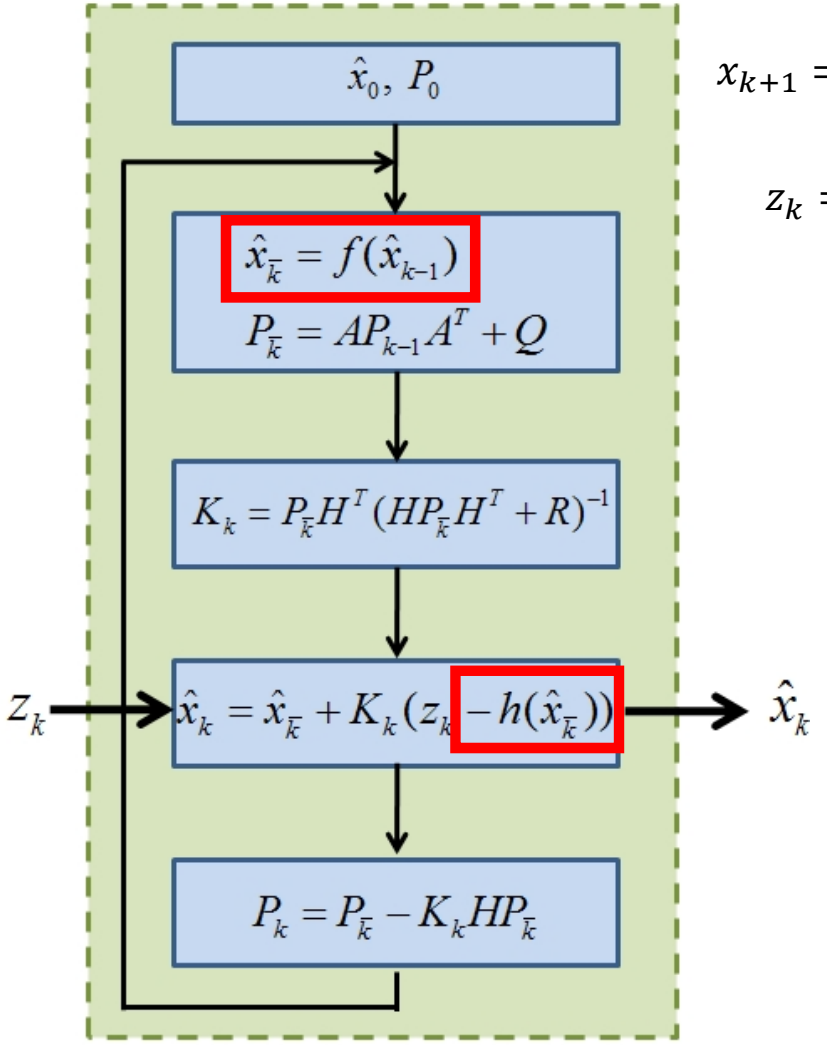
2. Extended Kalman Filter



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

$$z_k = Hx_k + v_k$$

Kalman Filter algorithm

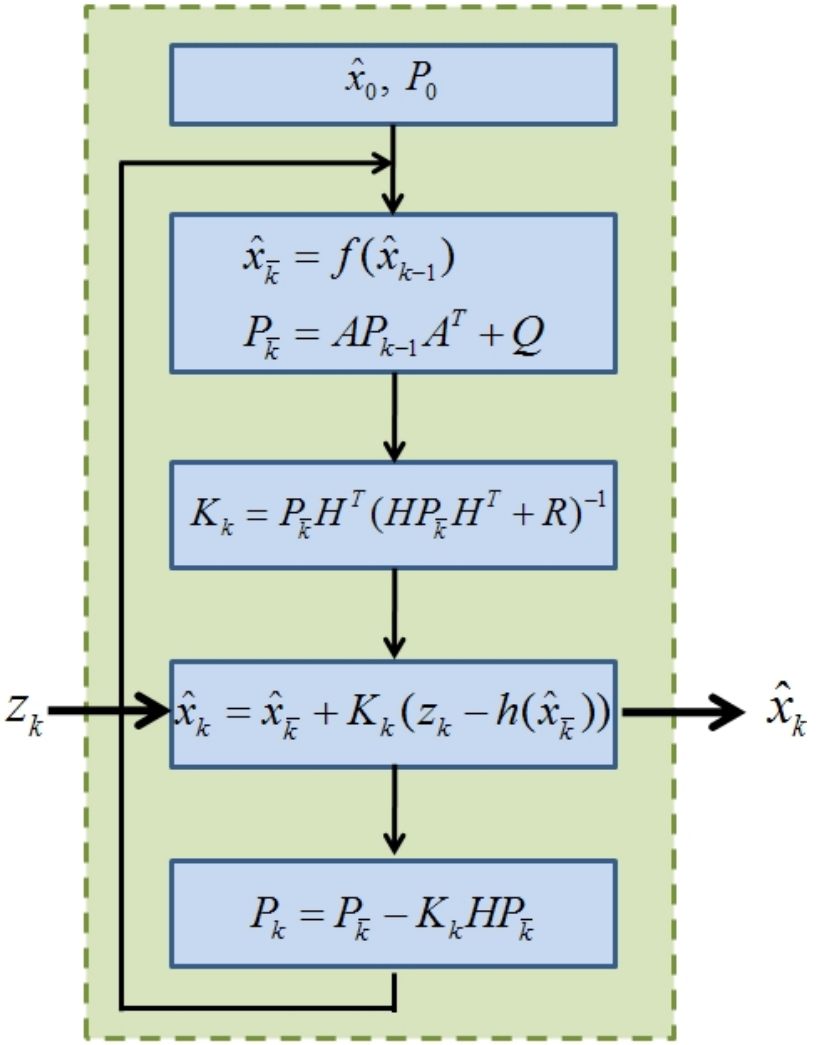


$$x_{k+1} = f(x_k) + w_k$$

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

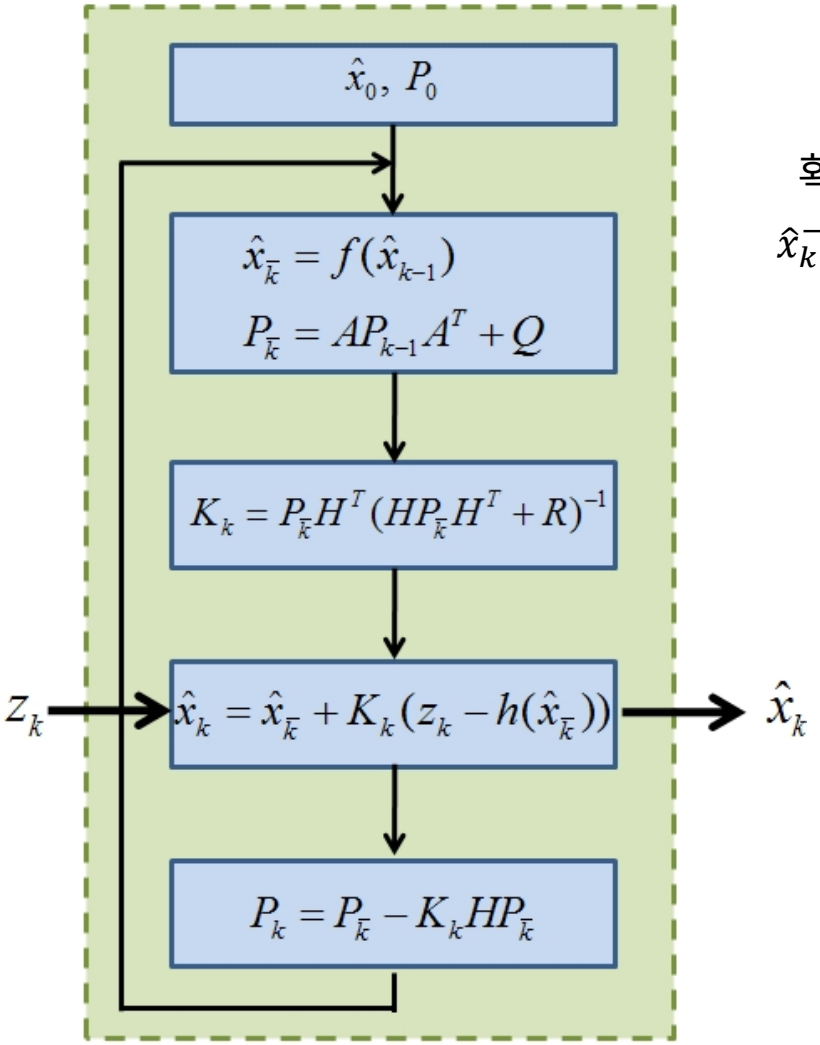
EKF algorithm

2. Extended Kalman Filter



EKF algorithm

2. Extended Kalman Filter

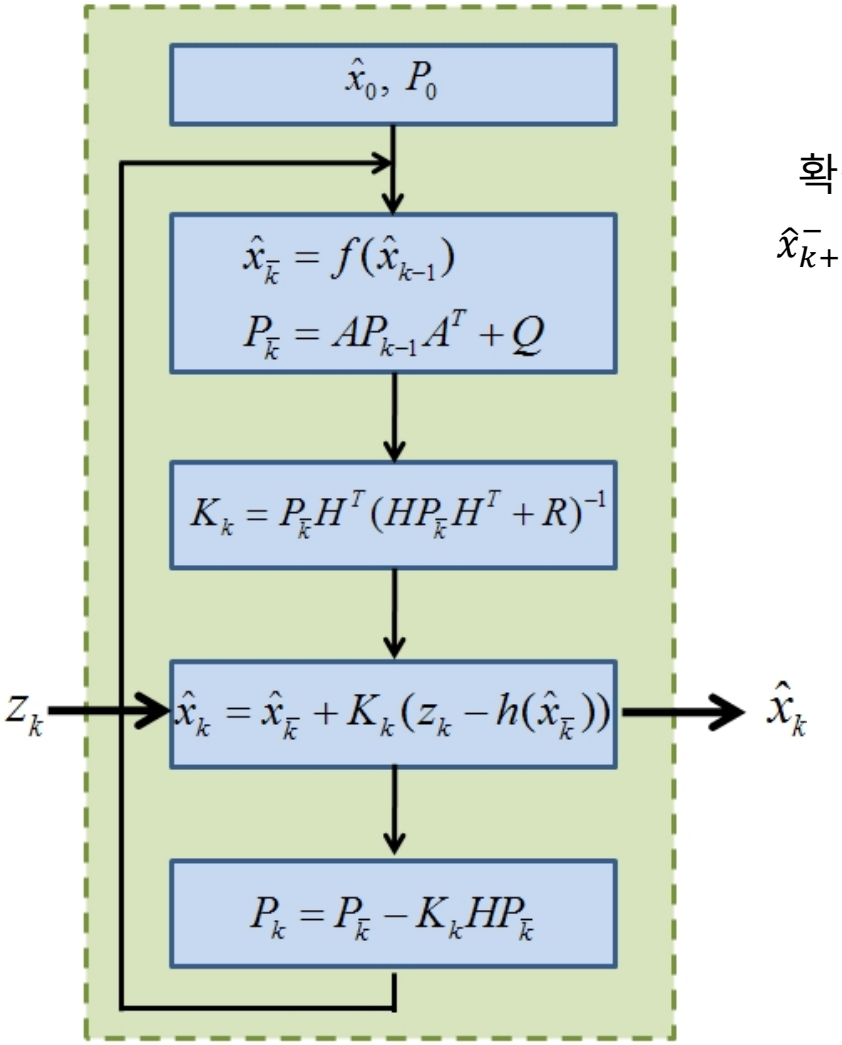


EKF algorithm

확장 칼만

$\hat{x}_{k+1}^{-} = f(\hat{x}_{k-1})$

2. Extended Kalman Filter



EKF algorithm

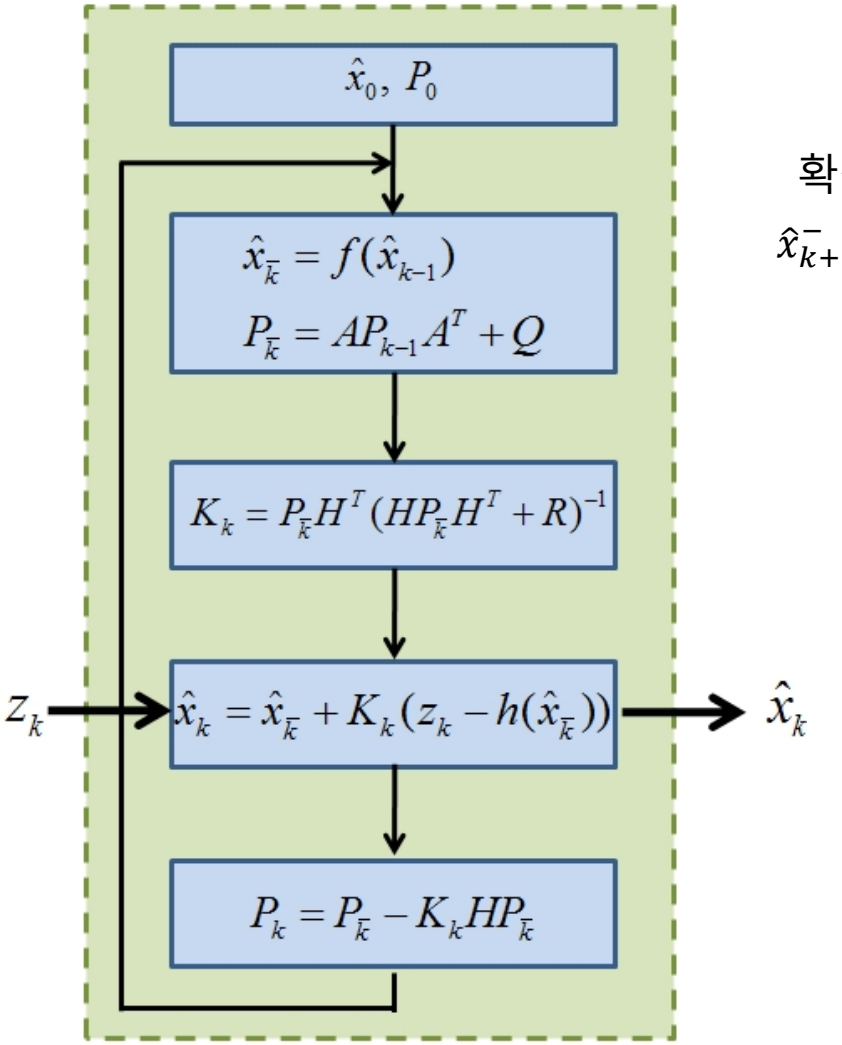
확장 칼만

$\hat{x}_{k+1}^- = f(\hat{x}_{k-1})$

확장 칼만

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

2. Extended Kalman Filter



EKF algorithm

확장 칼만

$\hat{x}_{k+1}^- = f(\hat{x}_{k-1})$

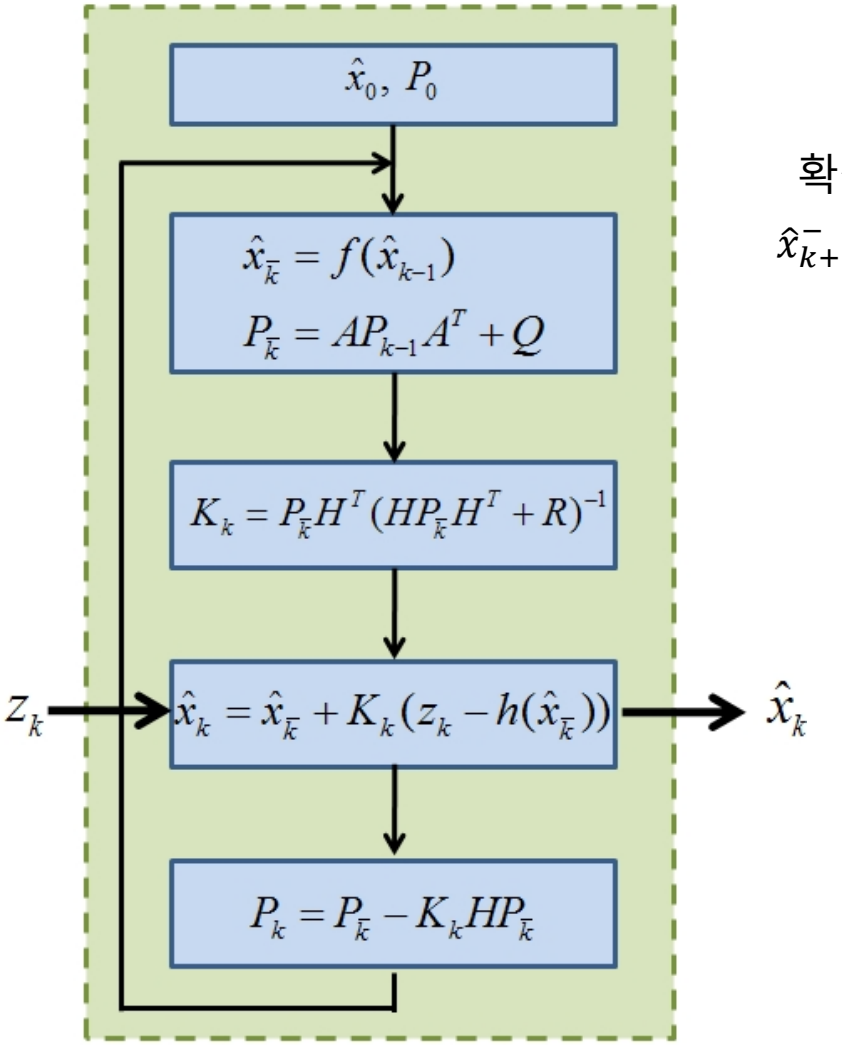
선형 칼만

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

확장 칼만

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

2. Extended Kalman Filter



EKF algorithm

확장 칼만

$\hat{x}_{k+1}^- = f(\hat{x}_{k-1})$

선형 칼만

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

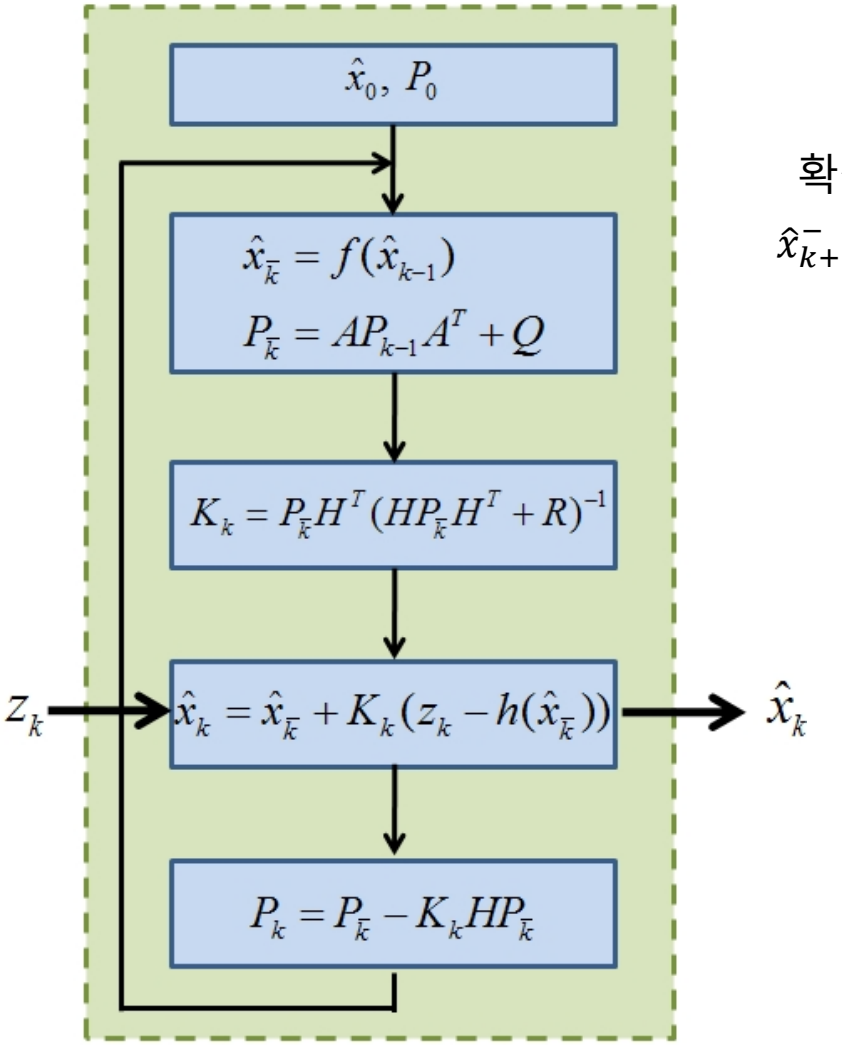
확장 칼만

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

선형 칼만

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

2. Extended Kalman Filter



EKF algorithm

확장 칼만

$$\hat{x}_{k+1}^- = f(\hat{x}_{k-1})$$

선형 칼만

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

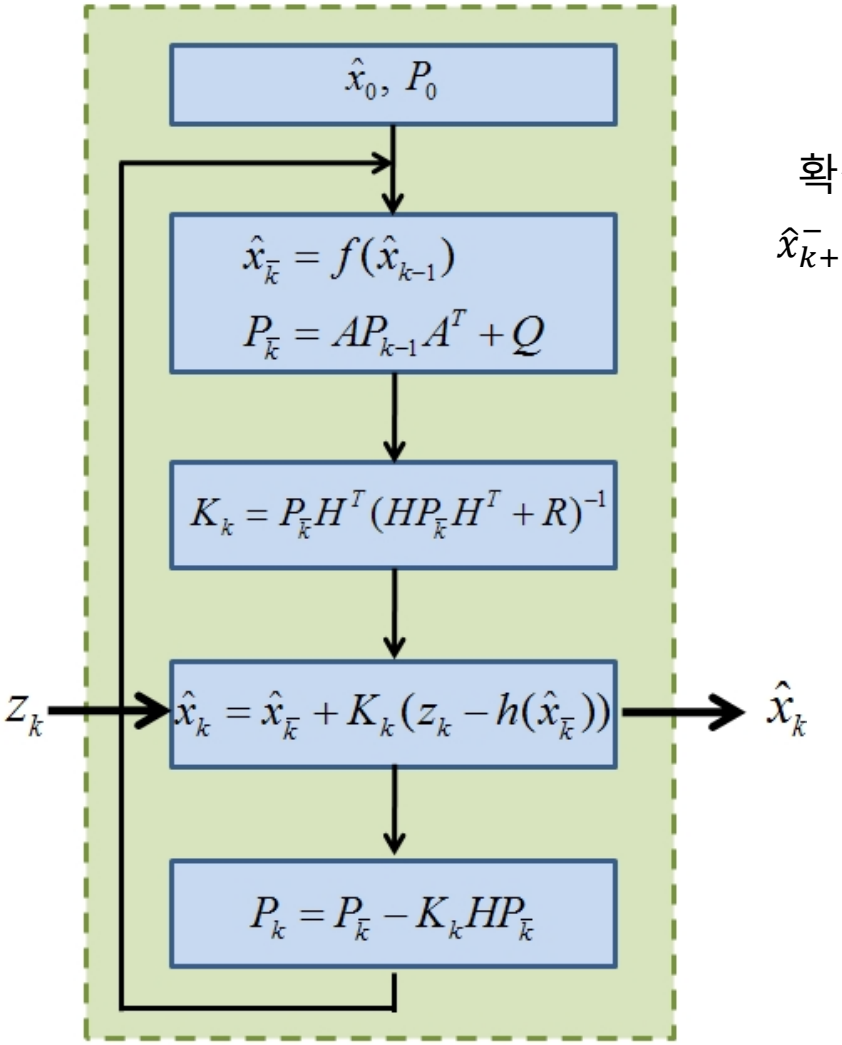
확장 칼만

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

선형 칼만

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

2. Extended Kalman Filter



EKF algorithm

확장 칼만

$\hat{x}_{k+1}^- = f(\hat{x}_{k-1})$

선형 칼만

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

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

확장 칼만

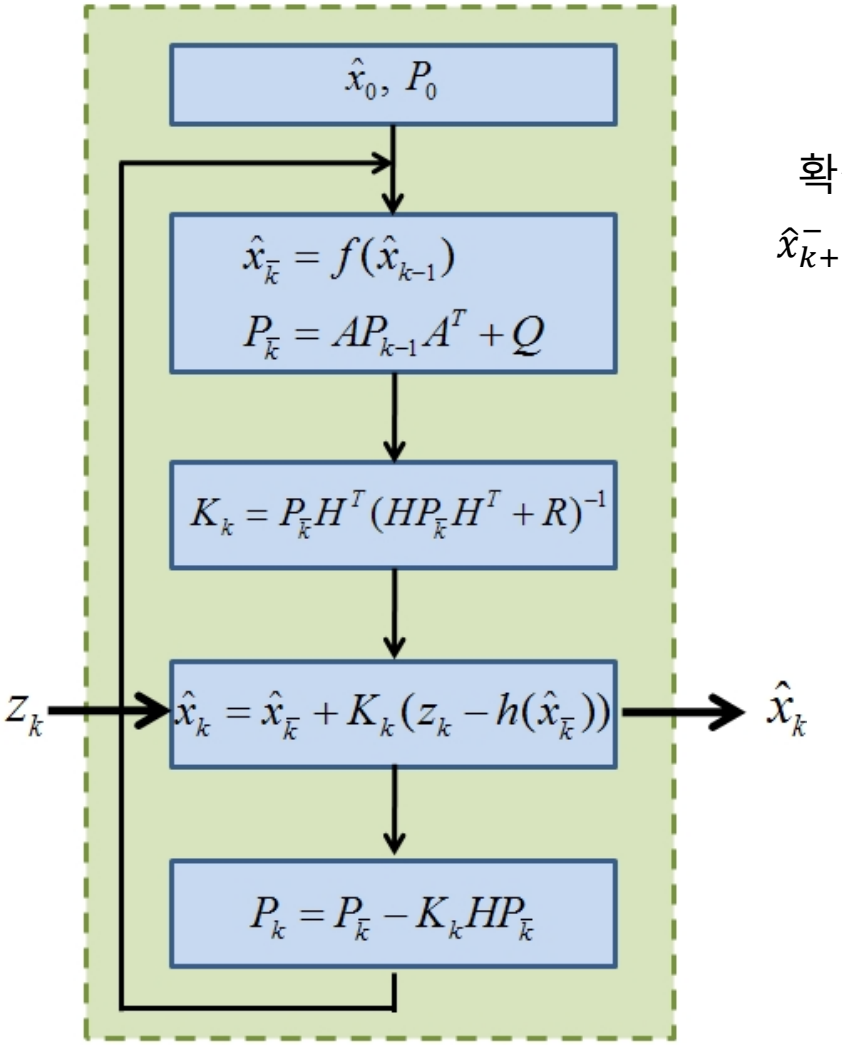
$\hat{x}_k = \hat{x}_k^- + K_k\{z_k - h(\hat{x}_k^-)\}$

선형 칼만

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

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

2. Extended Kalman Filter



EKF algorithm

확장 칼만

$\hat{x}_{k+1}^- = f(\hat{x}_{k-1})$

선형 칼만

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

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

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

확장 칼만

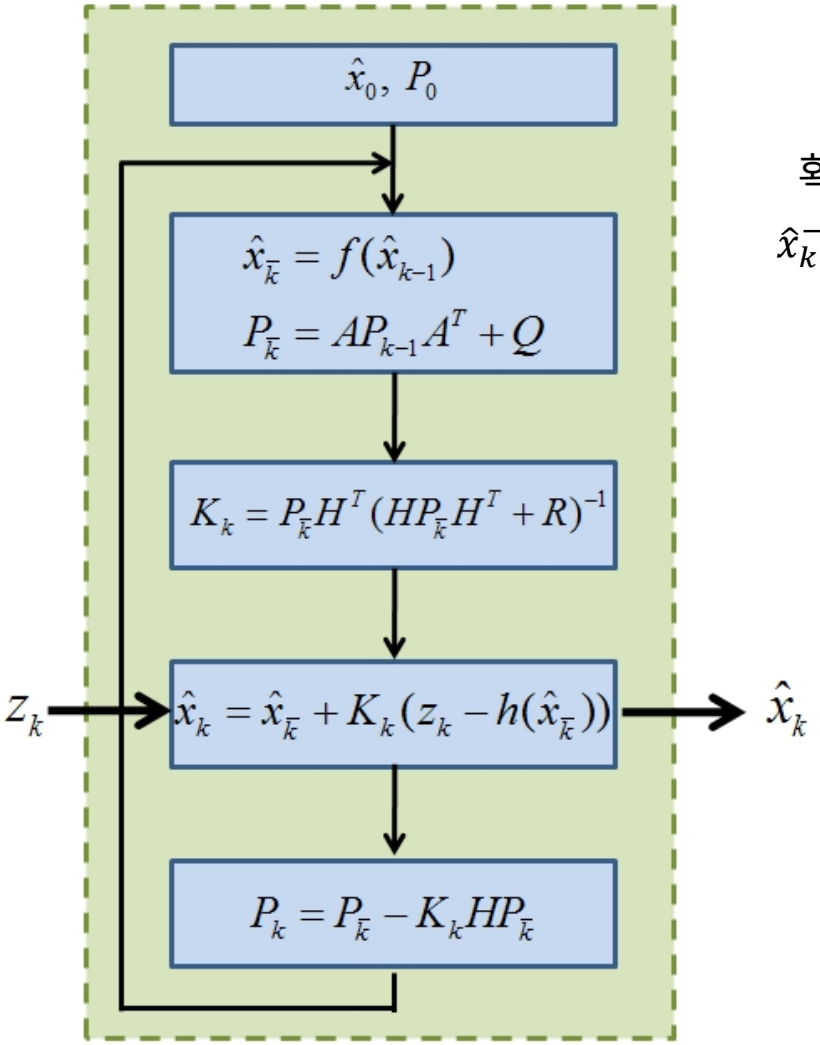
$\hat{x}_k = \hat{x}_k^- + K_k\{z_k - h(\hat{x}_k^-)\}$

선형 칼만

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

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

2. Extended Kalman Filter



EKF algorithm

확장 칼만

$\hat{x}_{k+1}^- = f(\hat{x}_{k-1})$

선형 칼만

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

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

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

확장 칼만

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

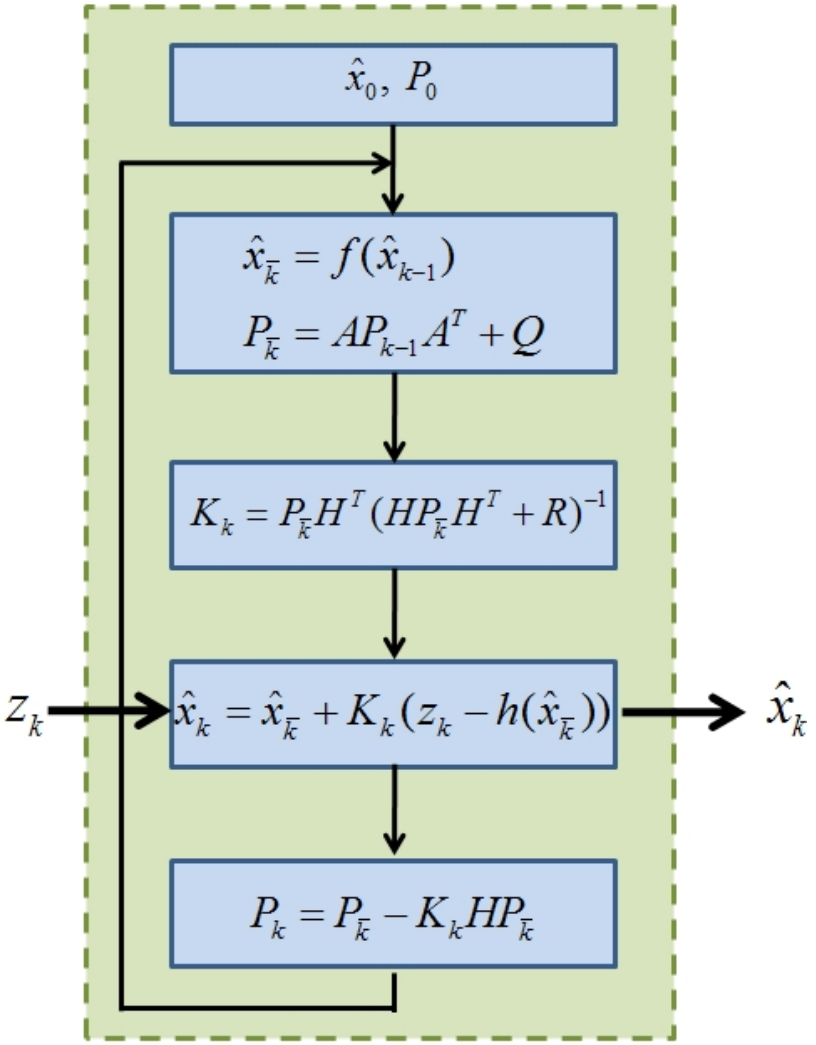
선형 칼만

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

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

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

2. Extended Kalman Filter



EKF algorithm

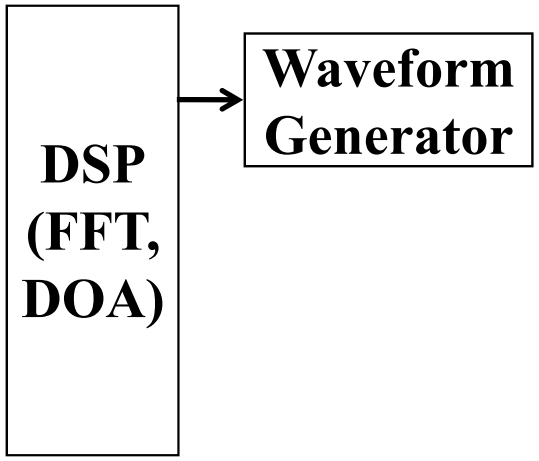
$$A \equiv \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_k} \qquad H \equiv \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k}$$

직전 추정값을 기준으로 선형 모델을 구한다.

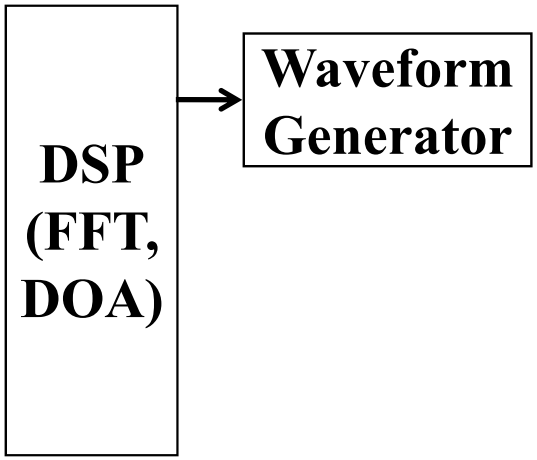
3. what is RADAR?

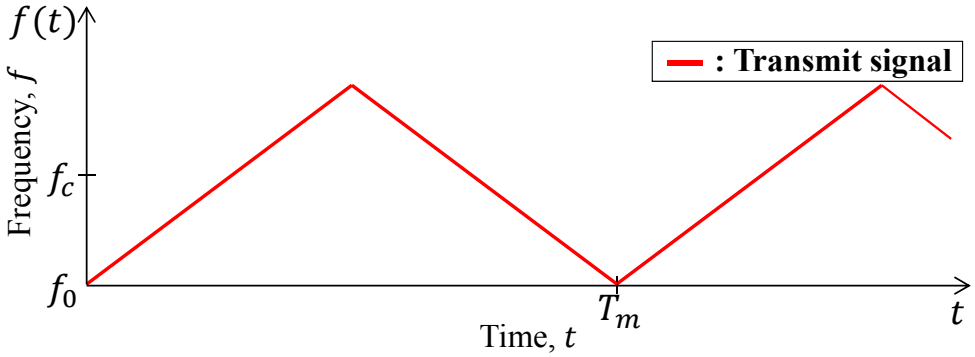
3. what is RADAR?

3. what is RADAR?

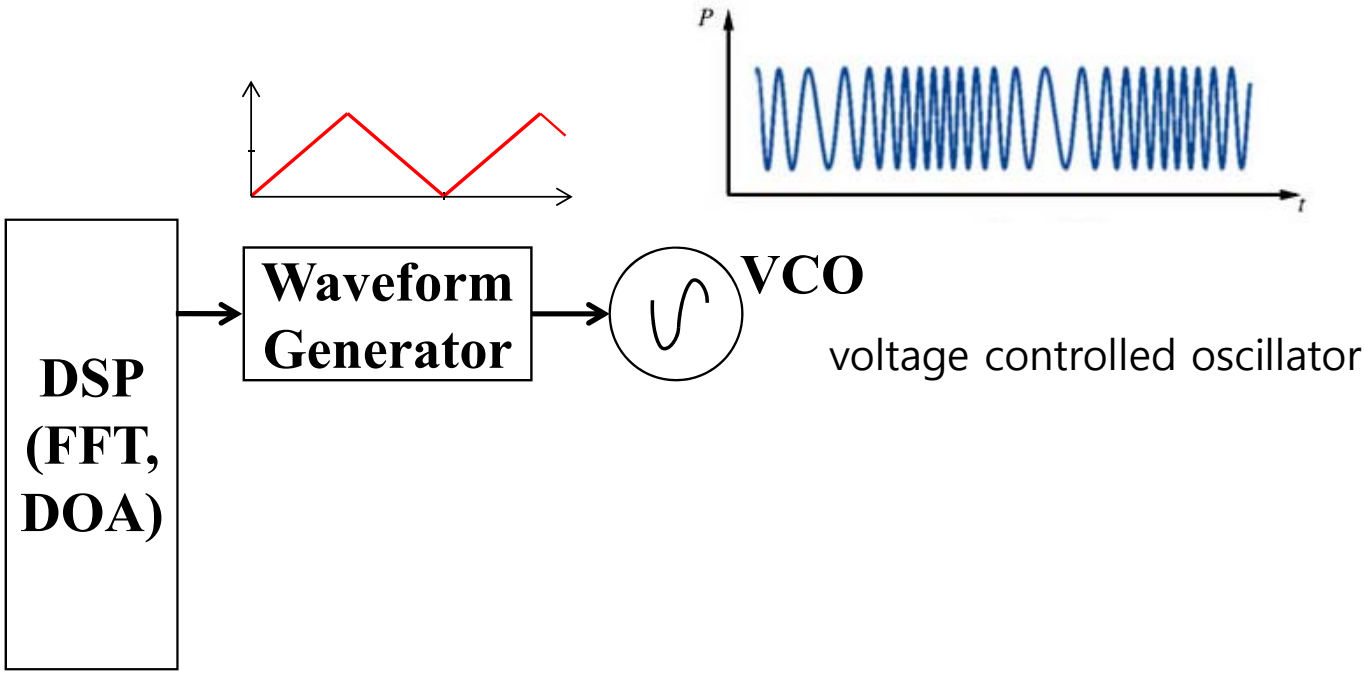


3. what is RADAR?

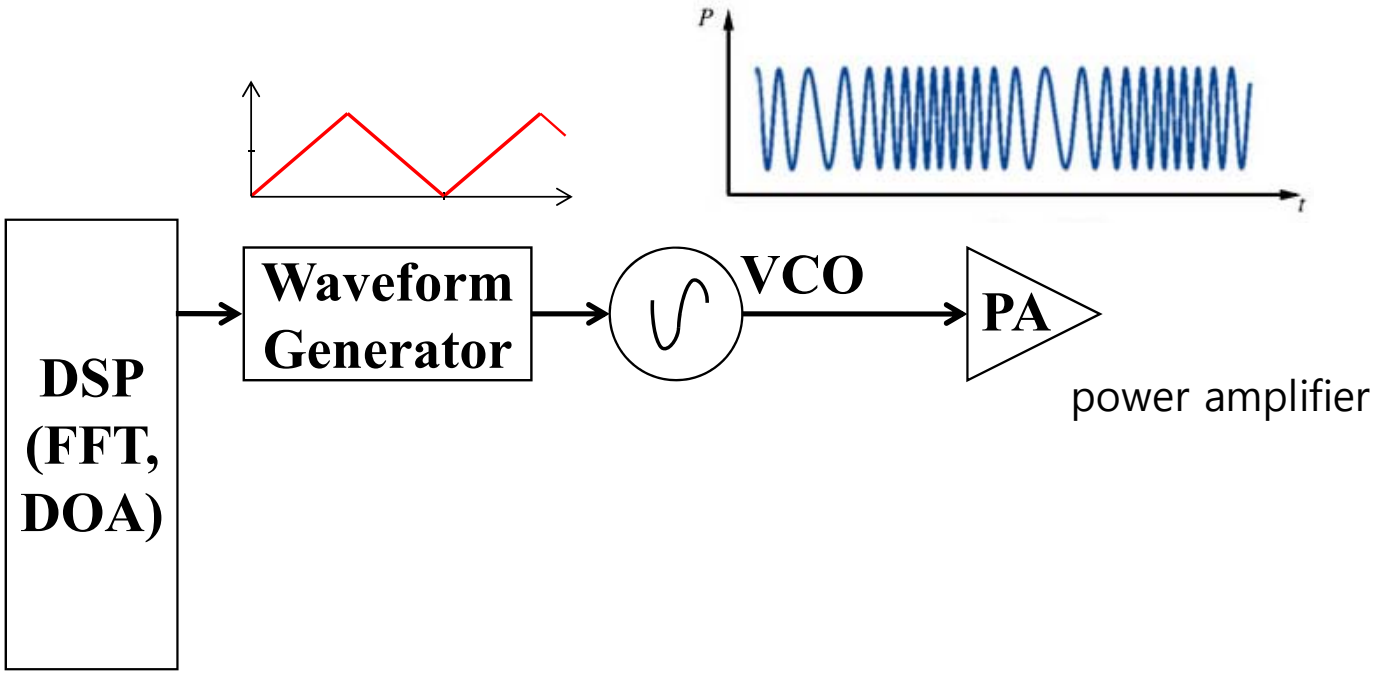




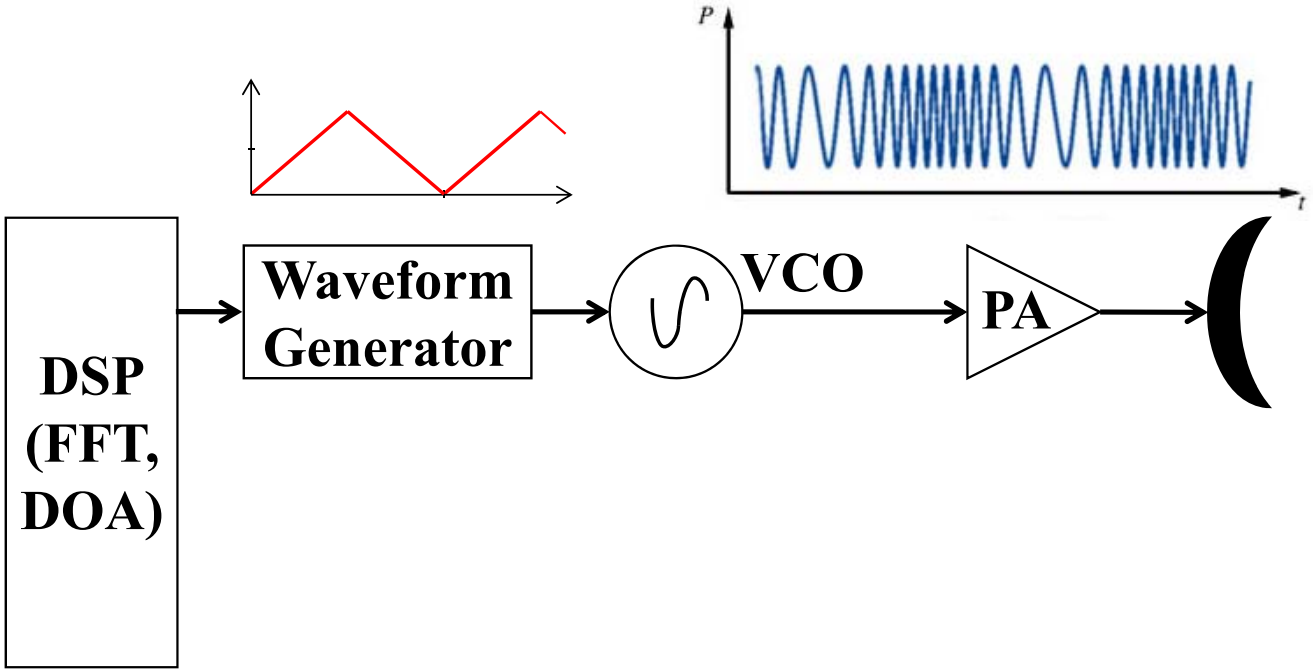
3. what is RADAR?



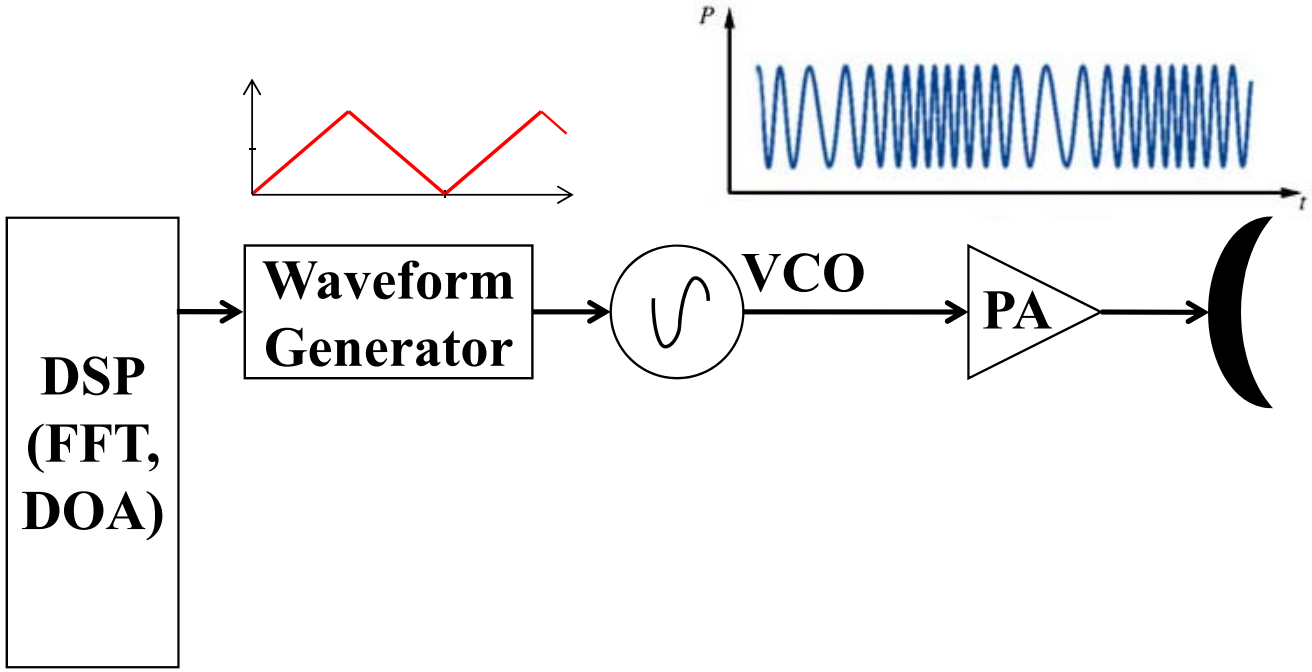
3. what is RADAR?

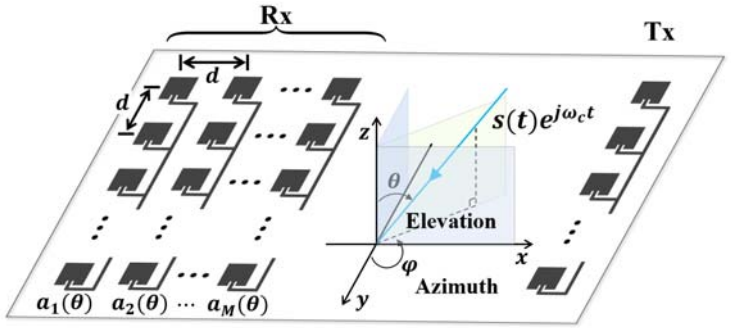


3. what is RADAR?

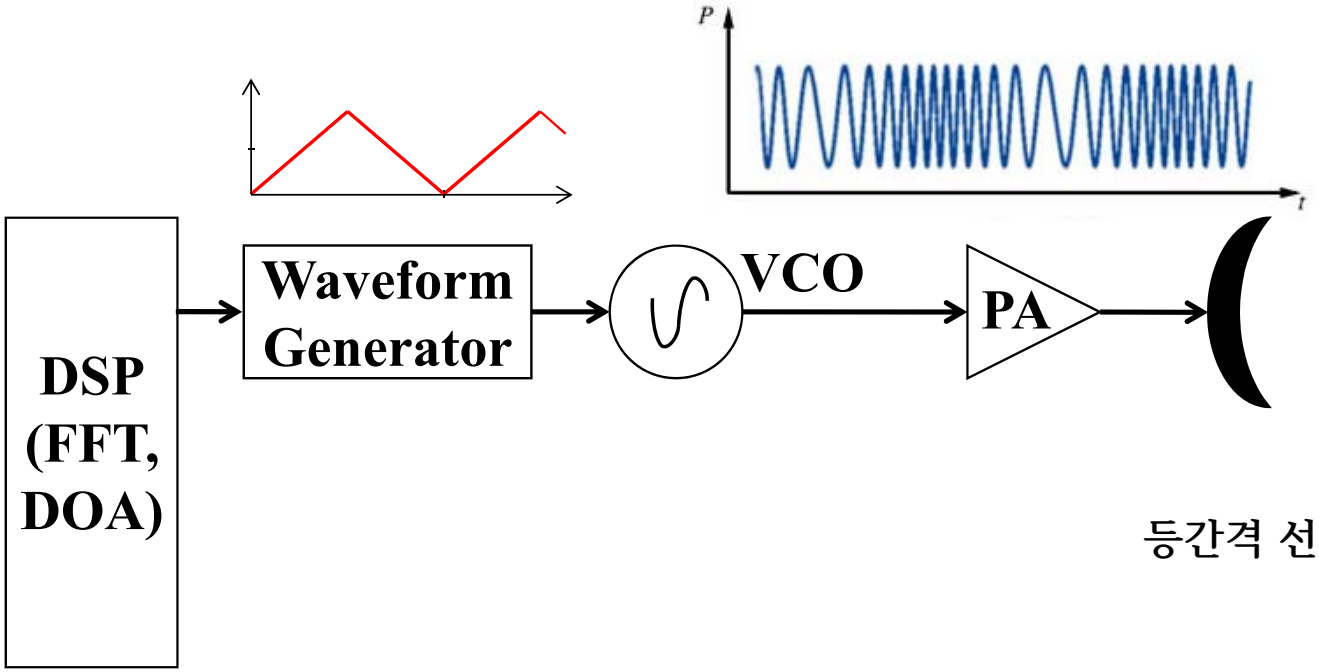


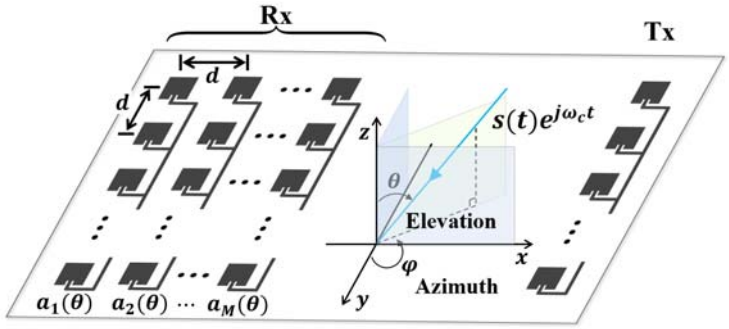
3. what is RADAR?





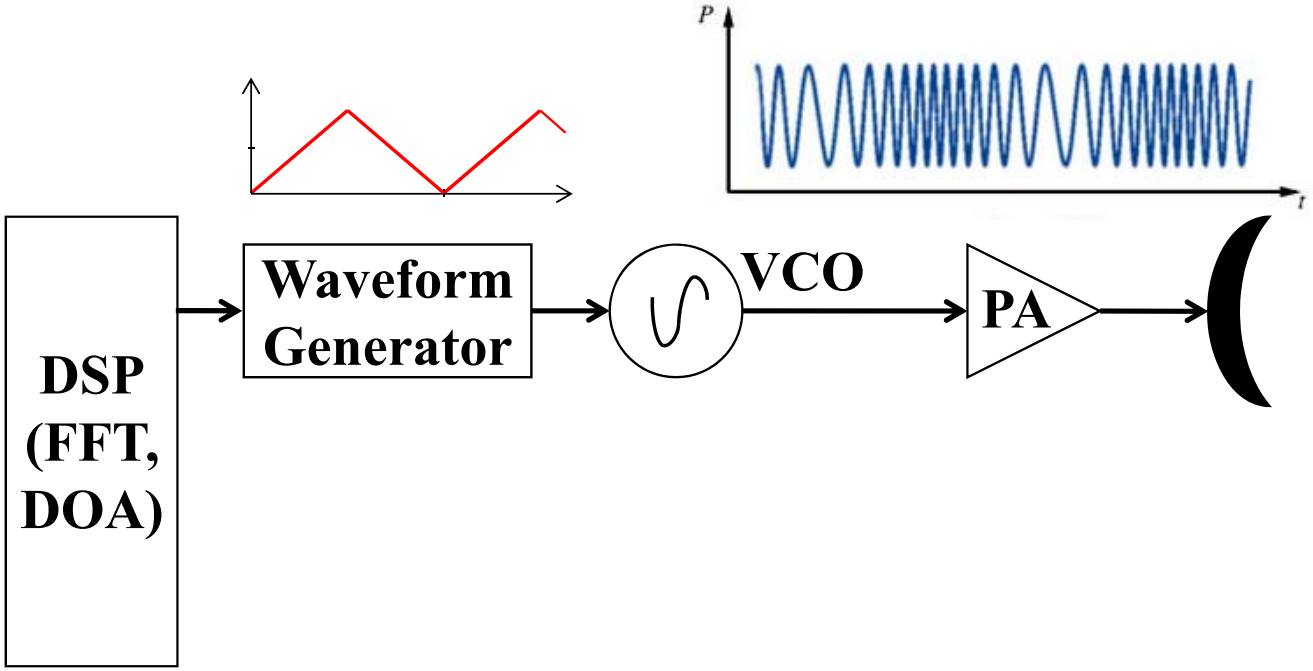
3. what is RADAR?

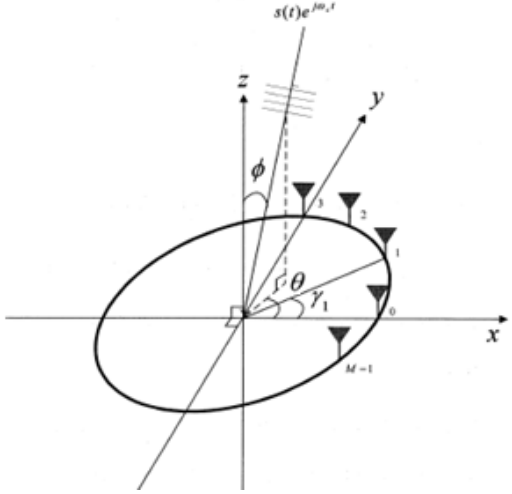




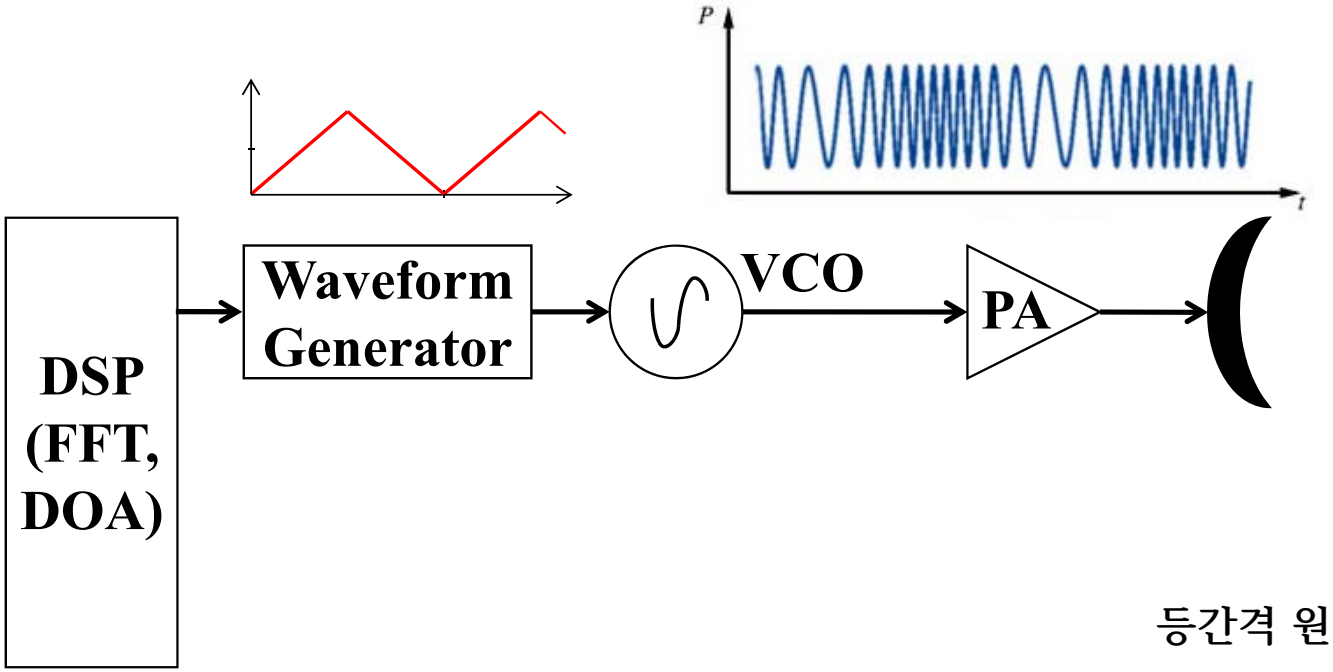
등간격 선형 어레이 (Uniform Linear Array : ULA)

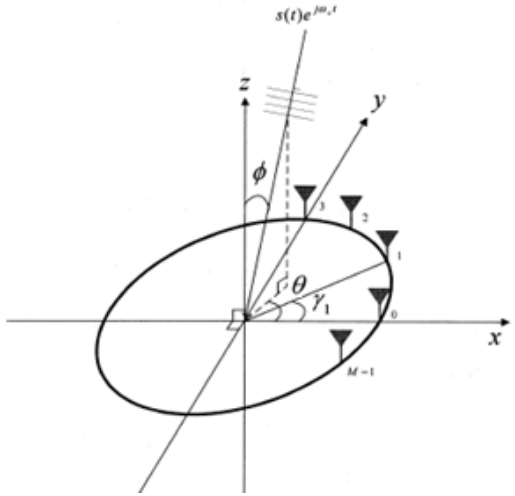
3. what is RADAR?





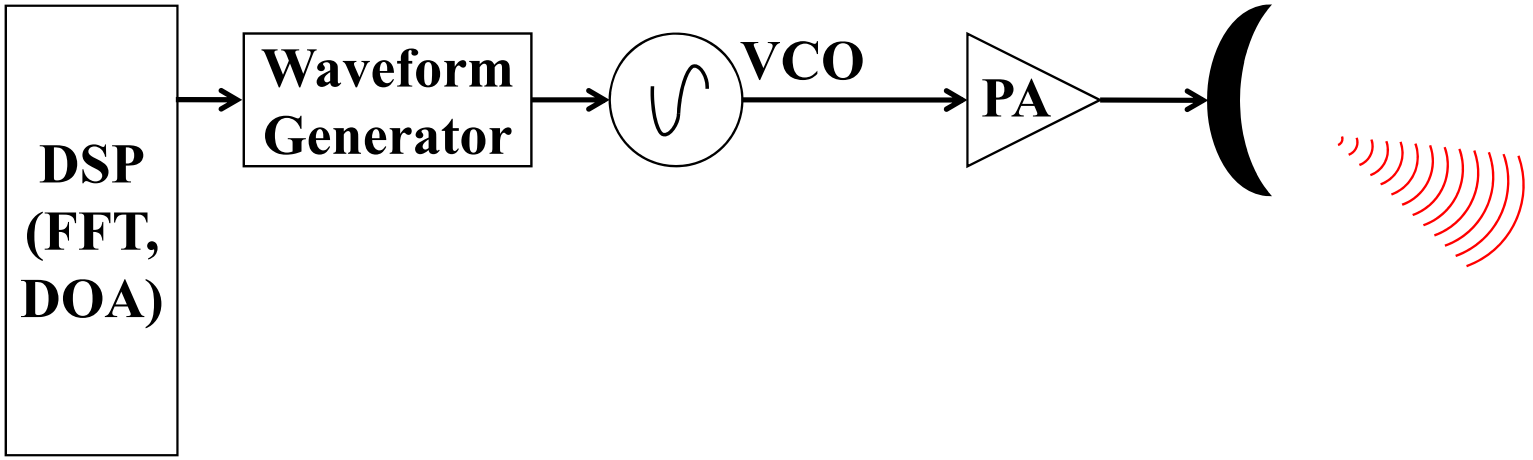
3. what is RADAR?



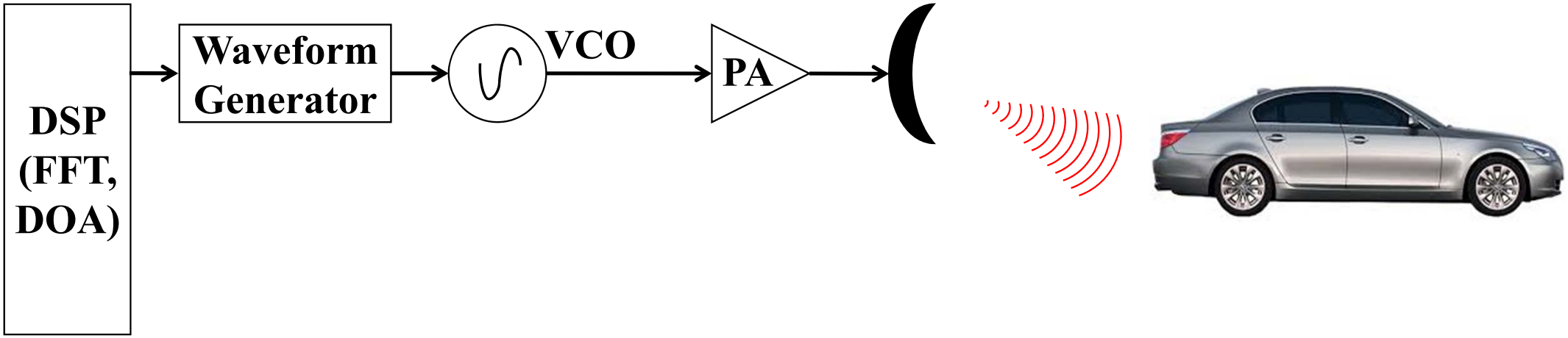


등간격 원형 어레이 (Uniform Circular Array : UCA)

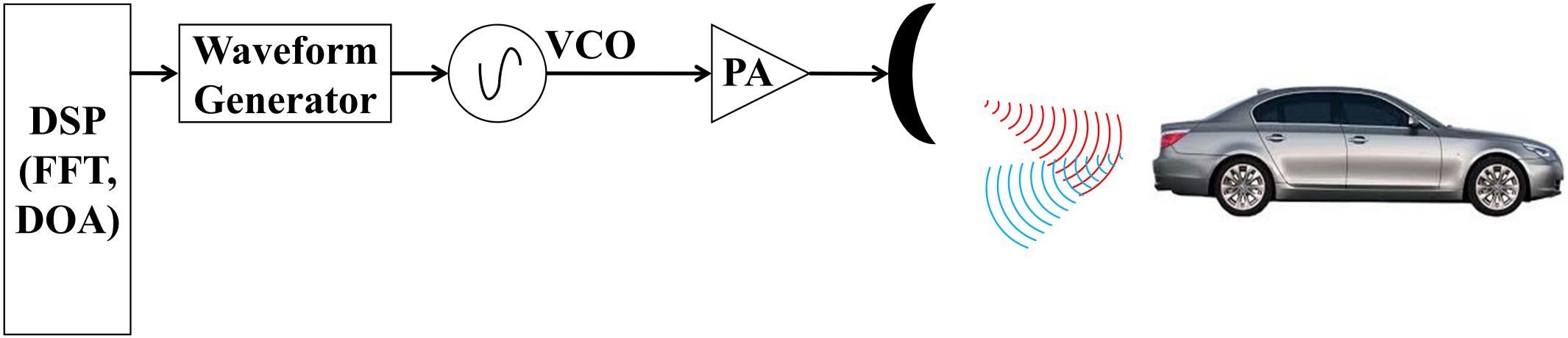
3. what is RADAR?



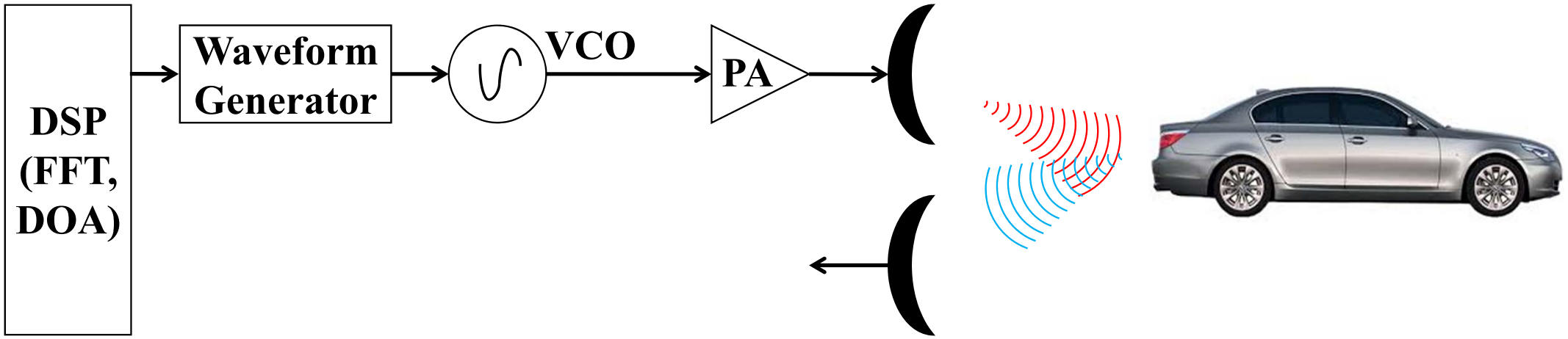
3. what is RADAR?



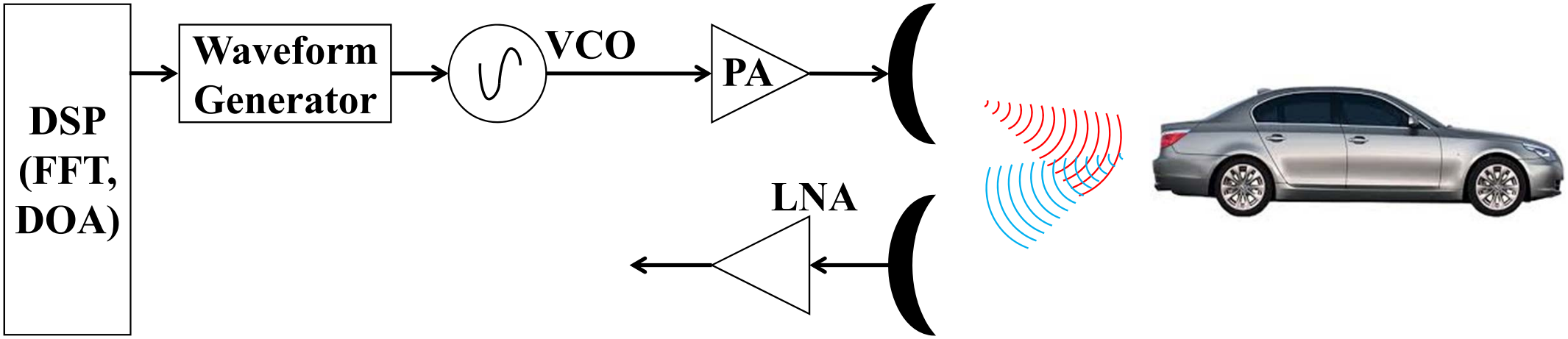
3. what is RADAR?



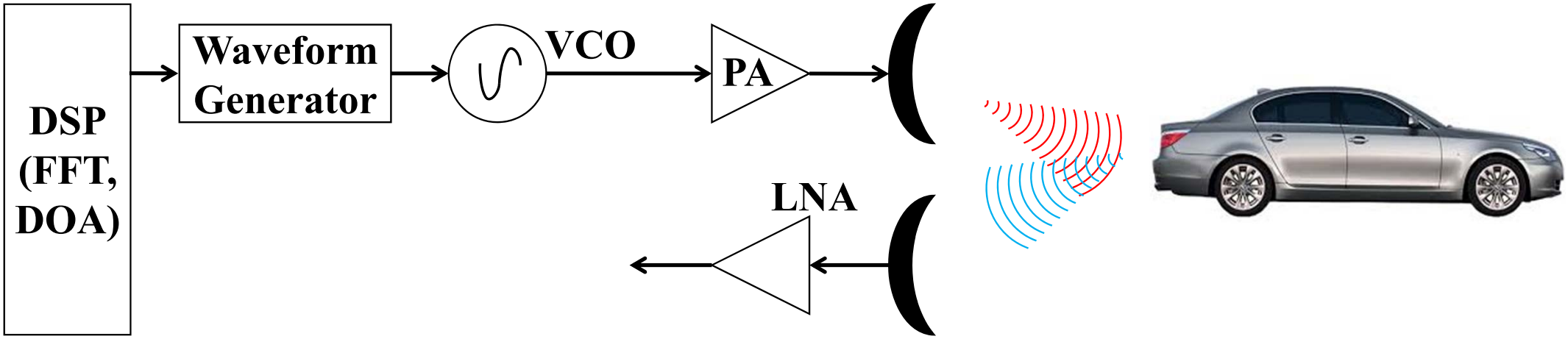
3. what is RADAR?



3. what is RADAR?

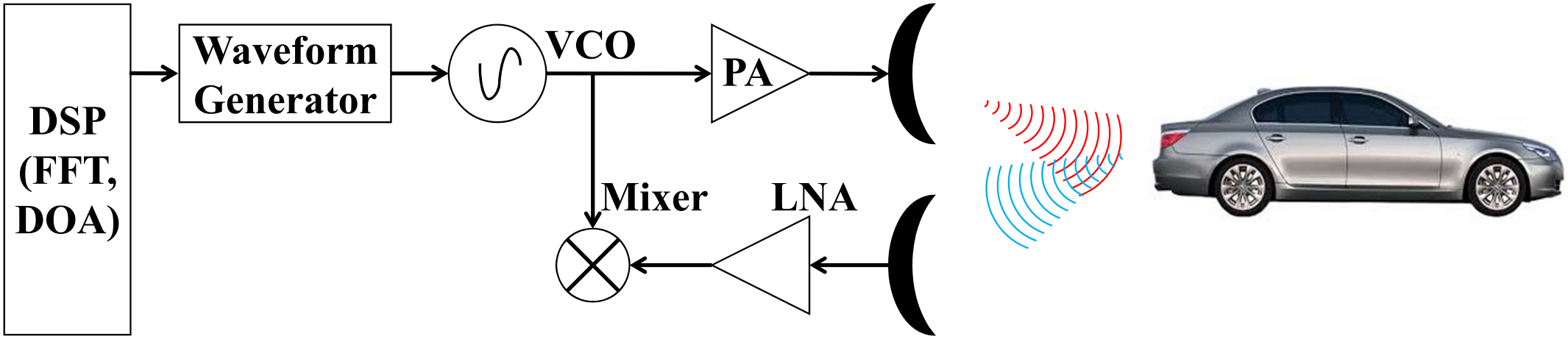


3. what is RADAR?

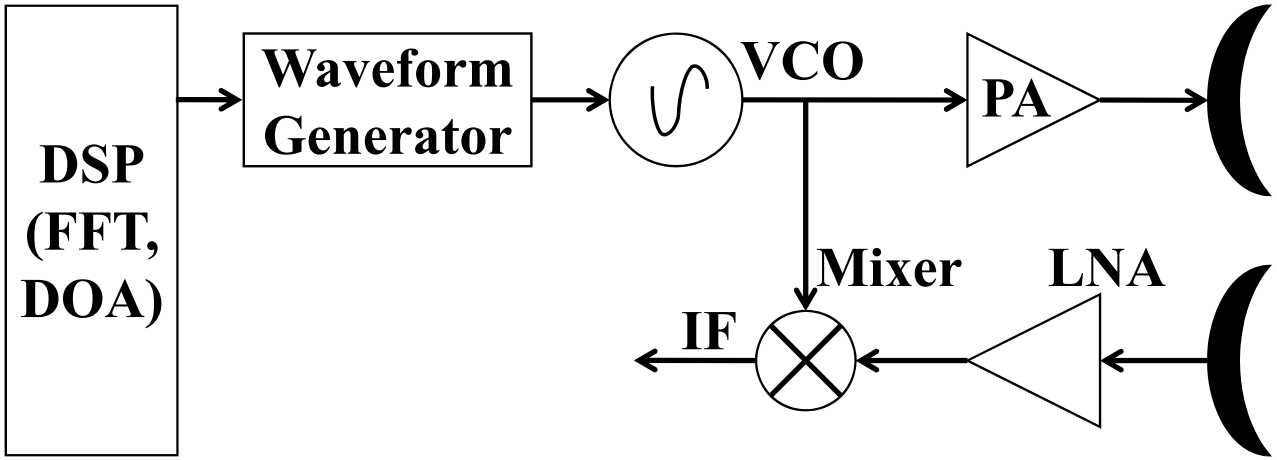


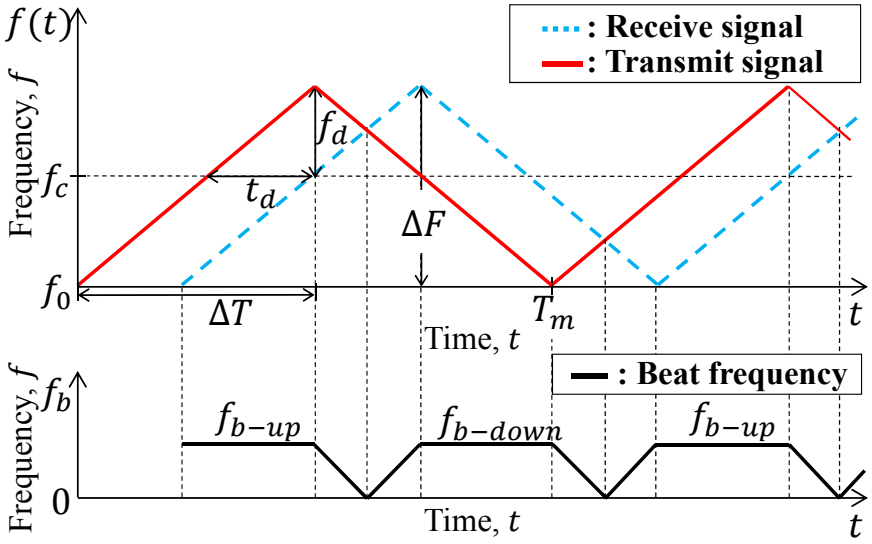
Low Noise Amplifier

3. what is RADAR?

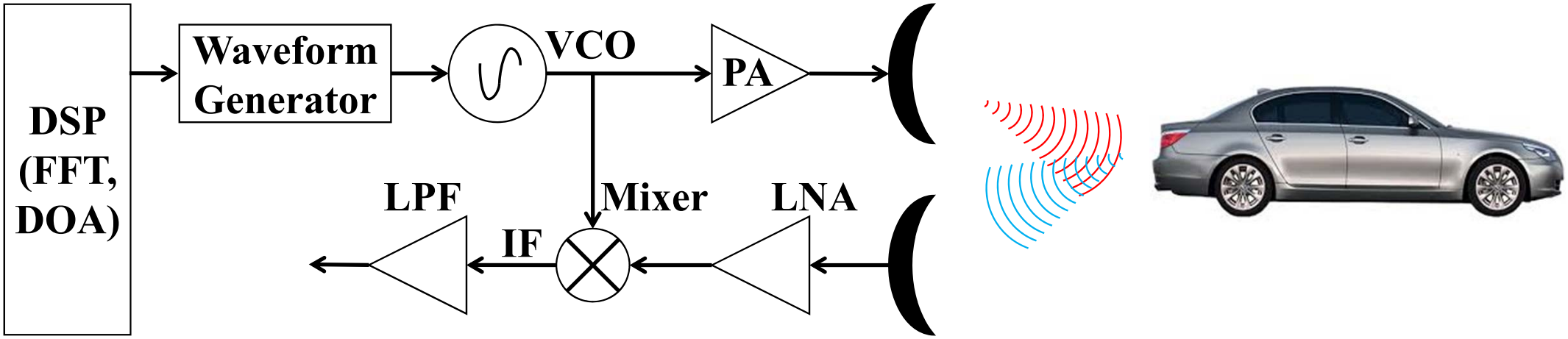


3. what is RADAR?

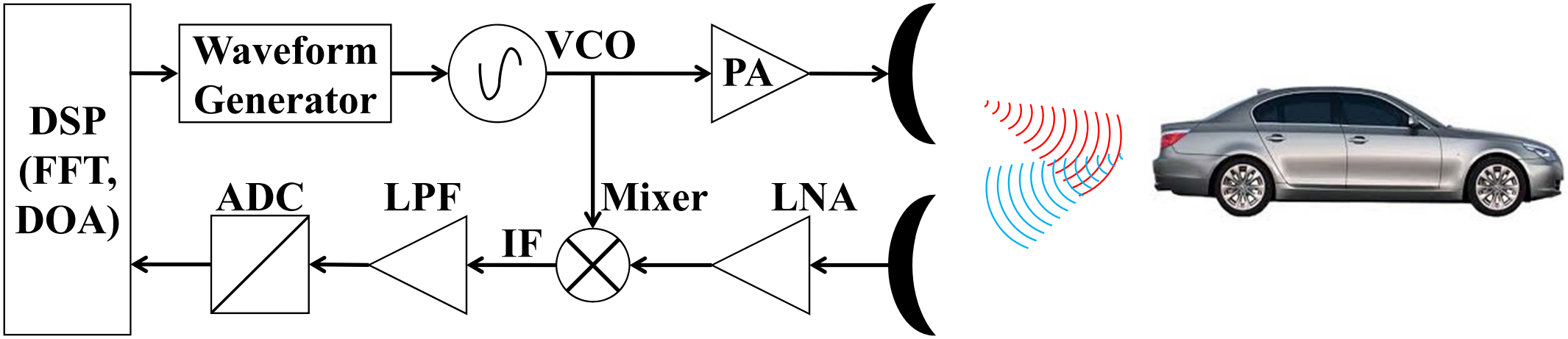




3. what is RADAR?



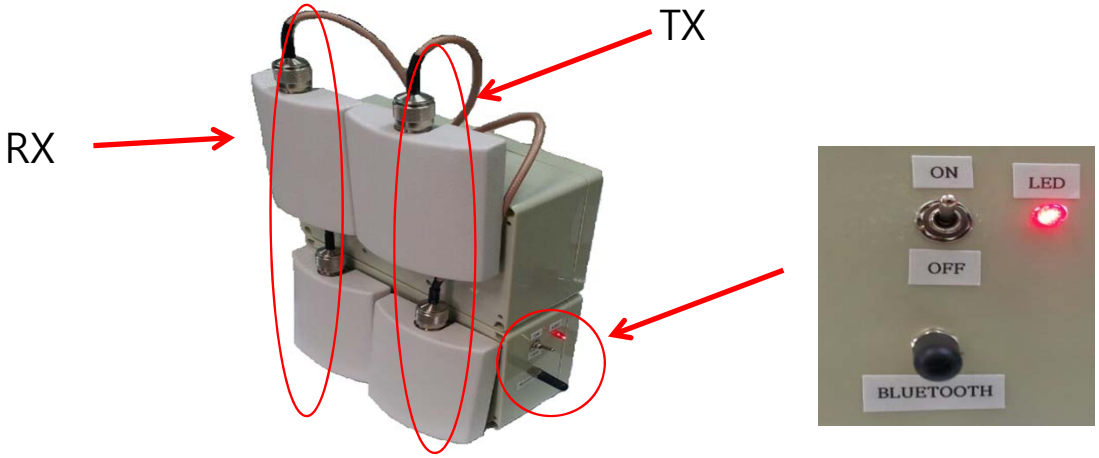
3. what is RADAR?



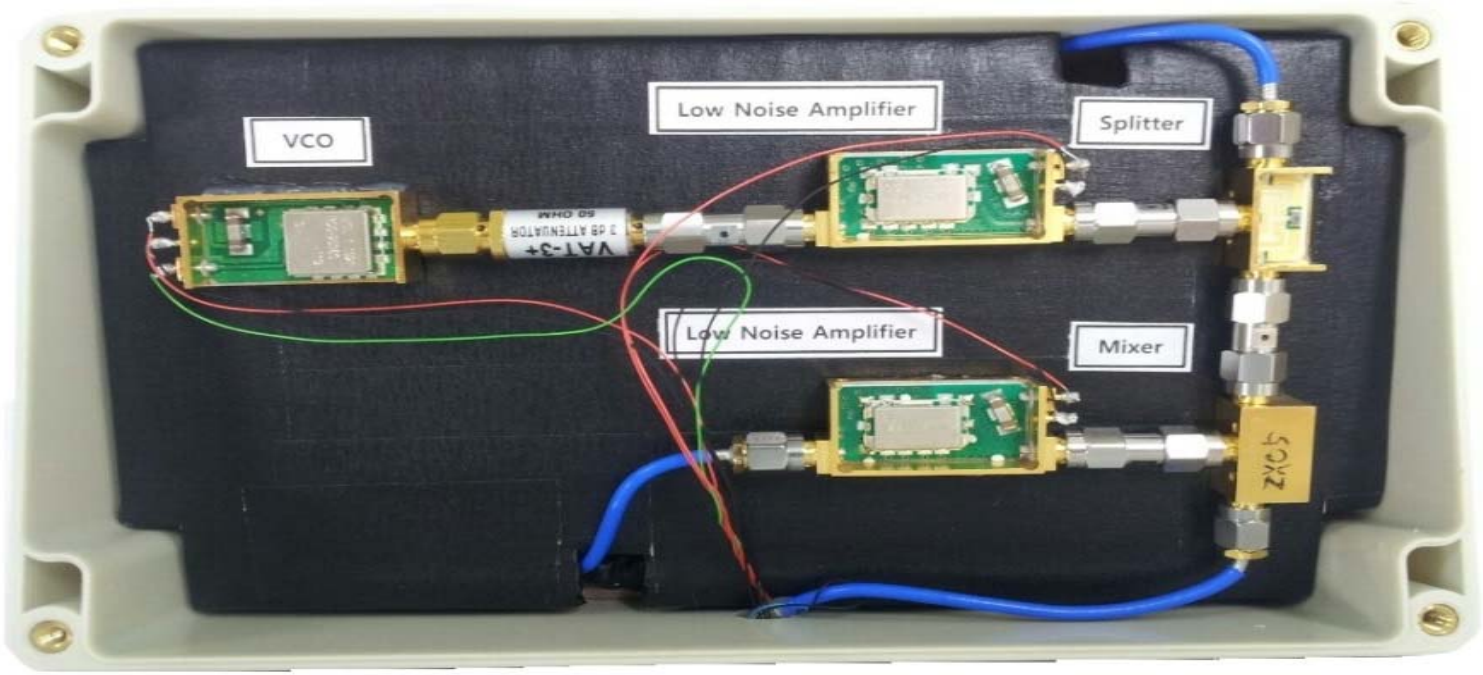
3. what is RADAR?

3. what is RADAR?

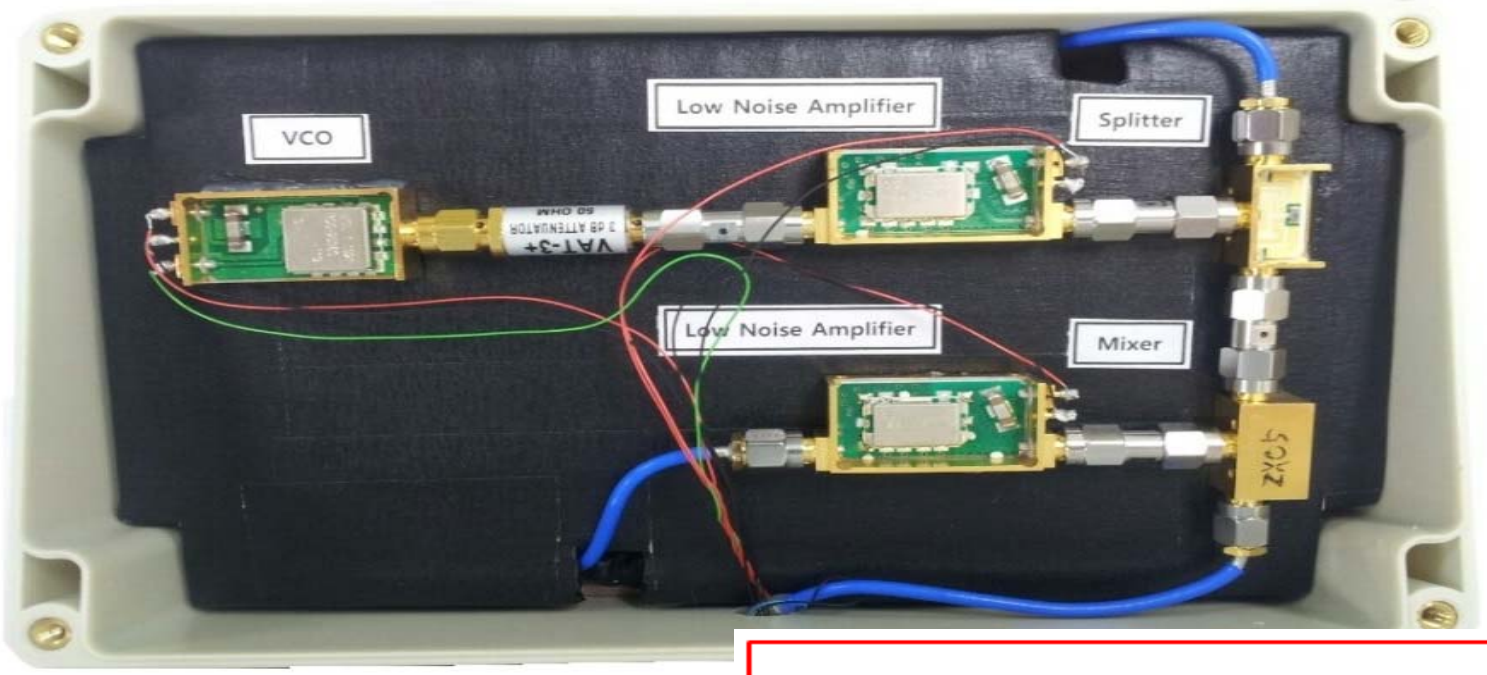
3. what is RADAR?

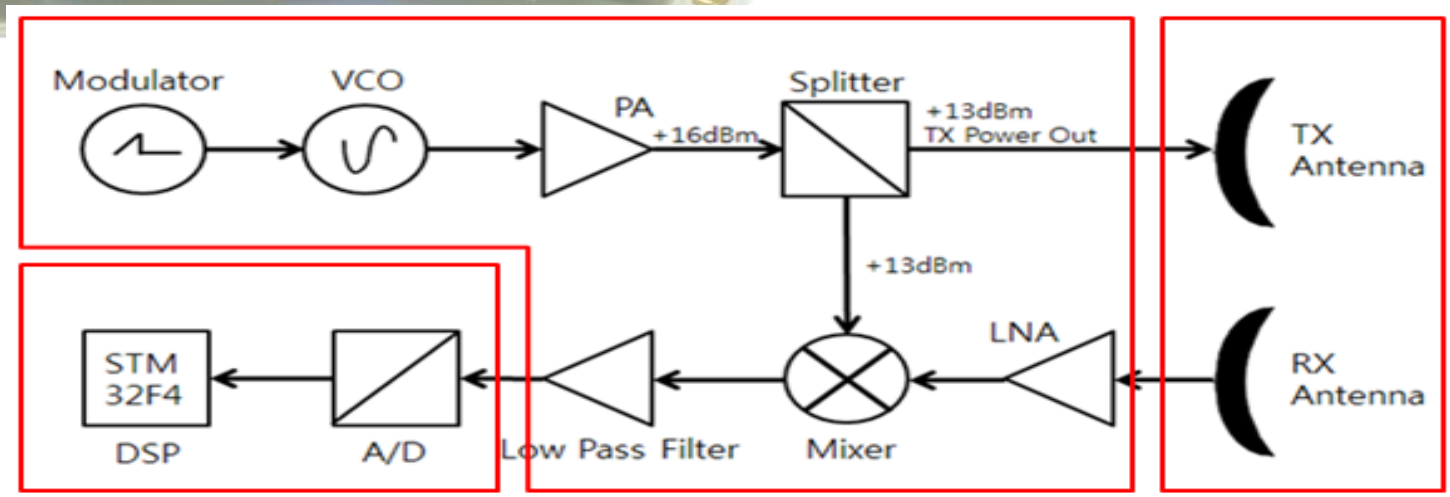


3. what is RADAR?

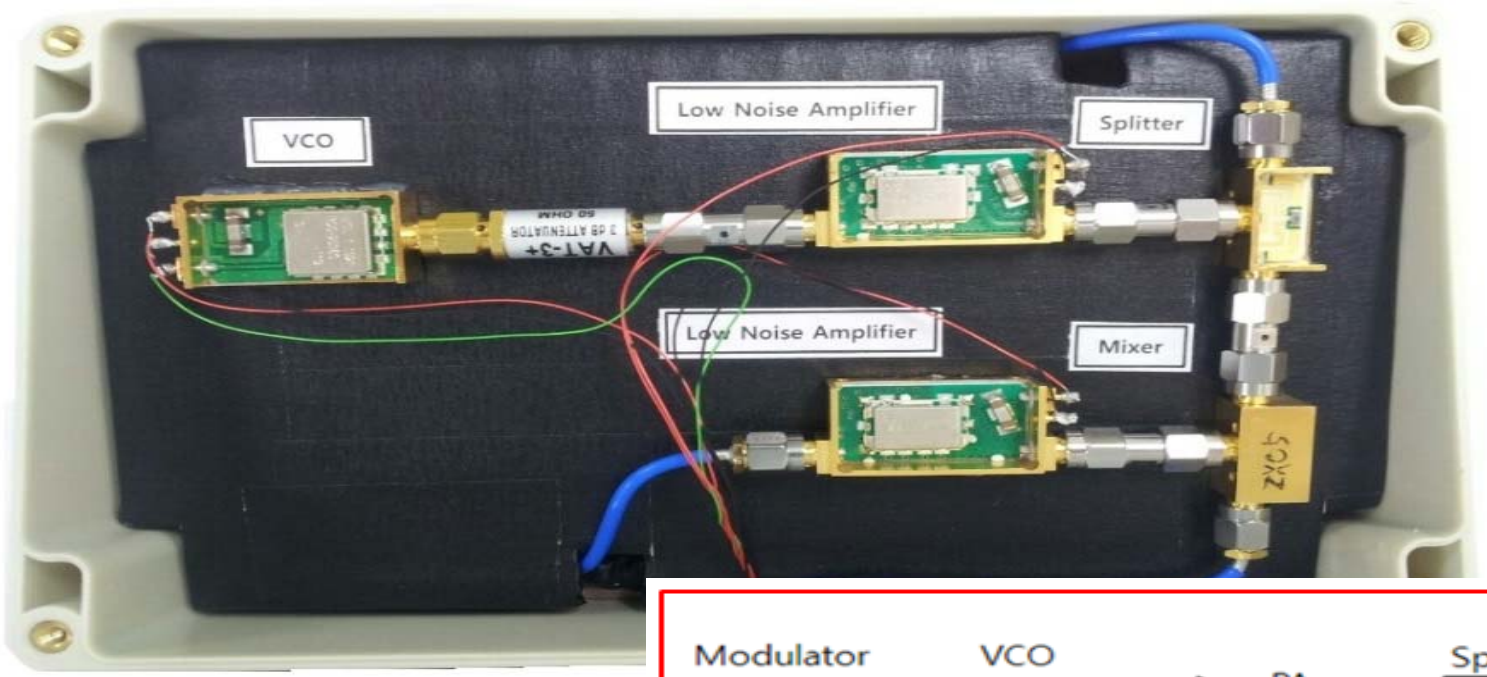


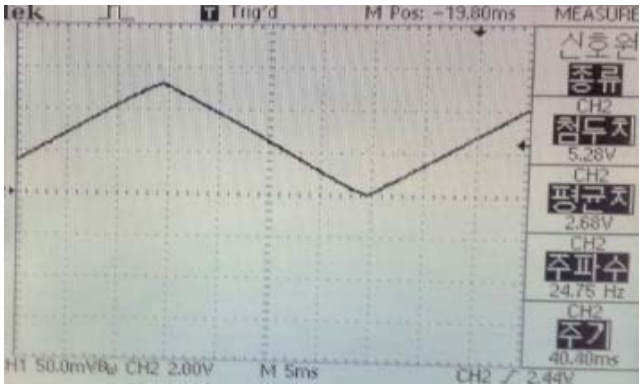
3. what is RADAR?

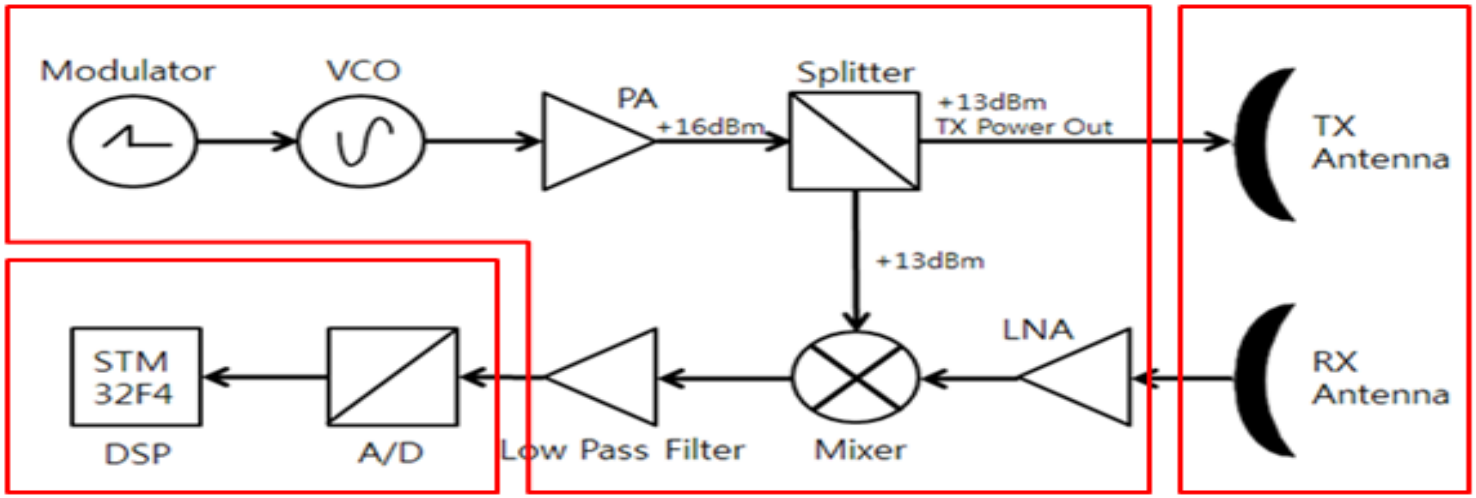




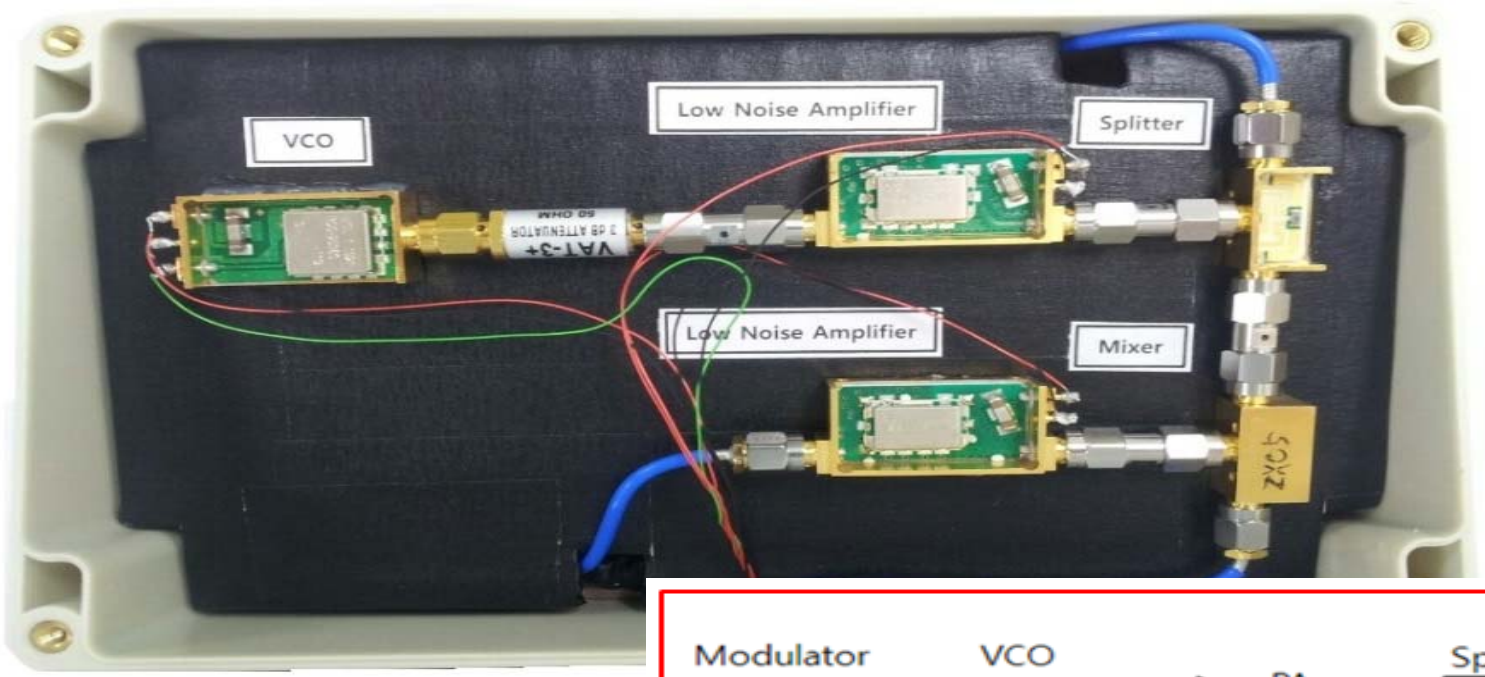
3. what is RADAR?

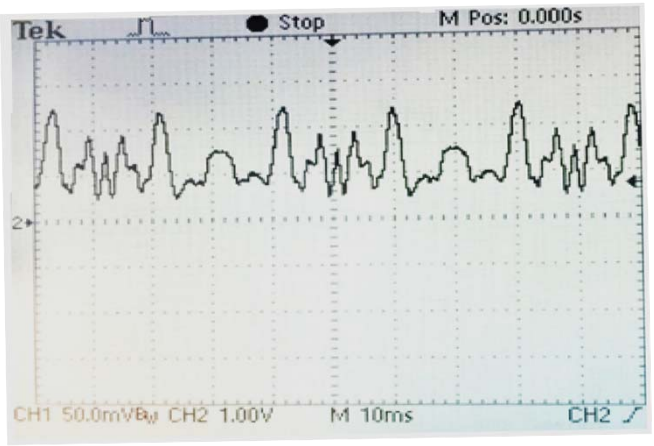


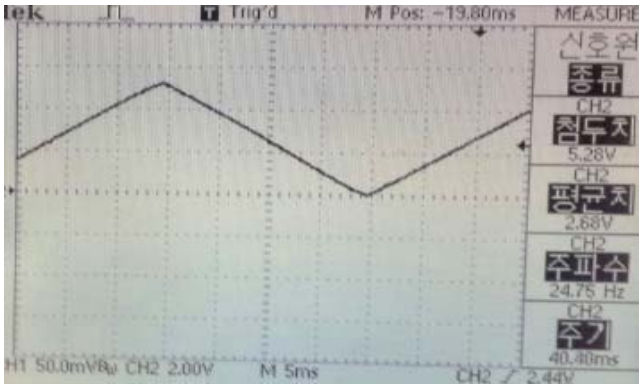


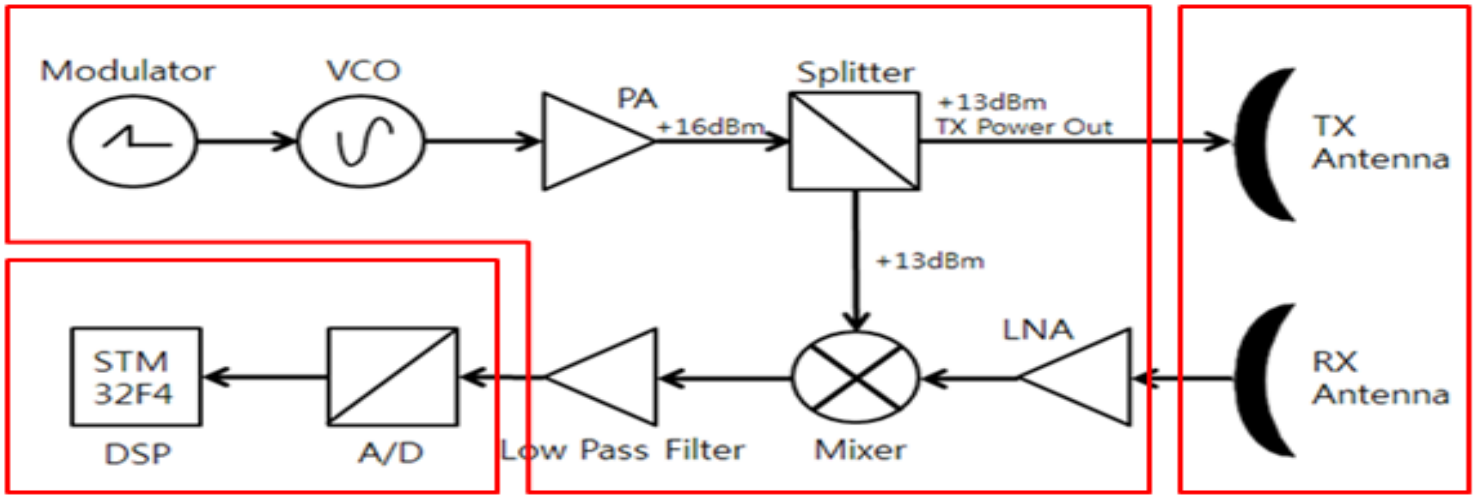


3. what is RADAR?

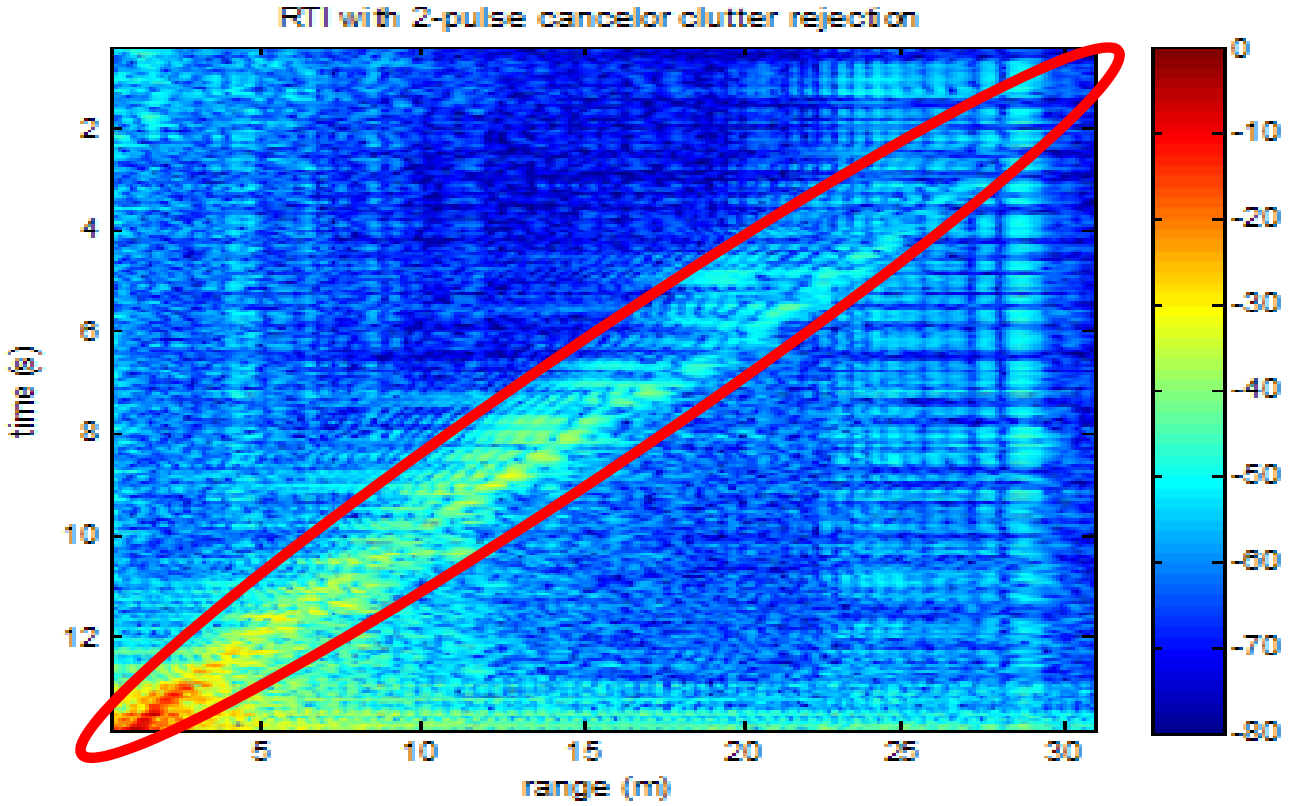






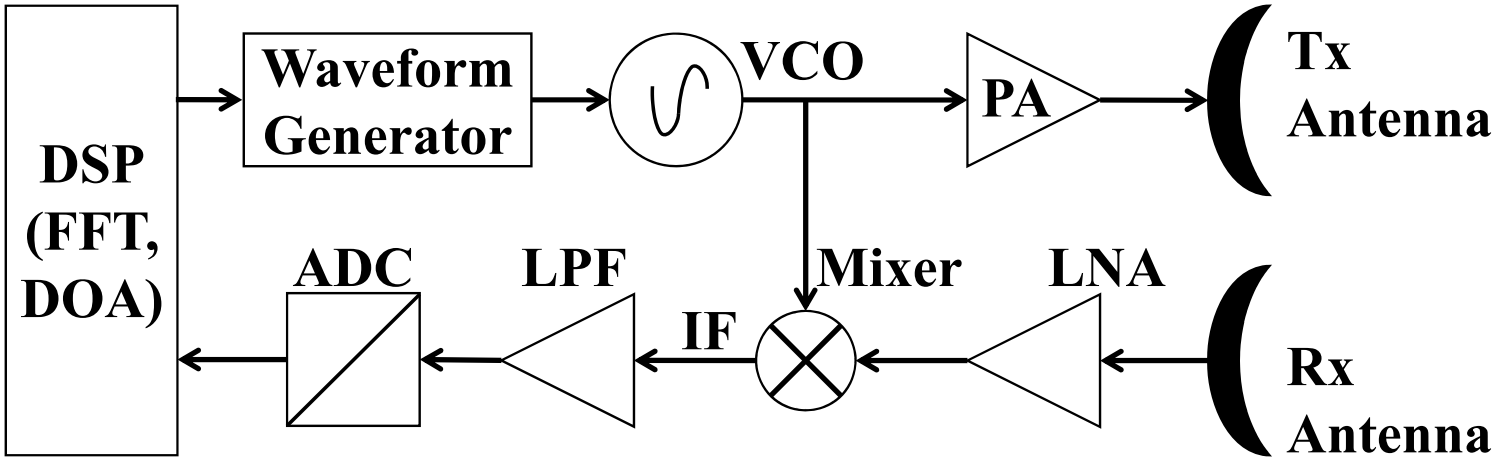


3. what is RADAR?

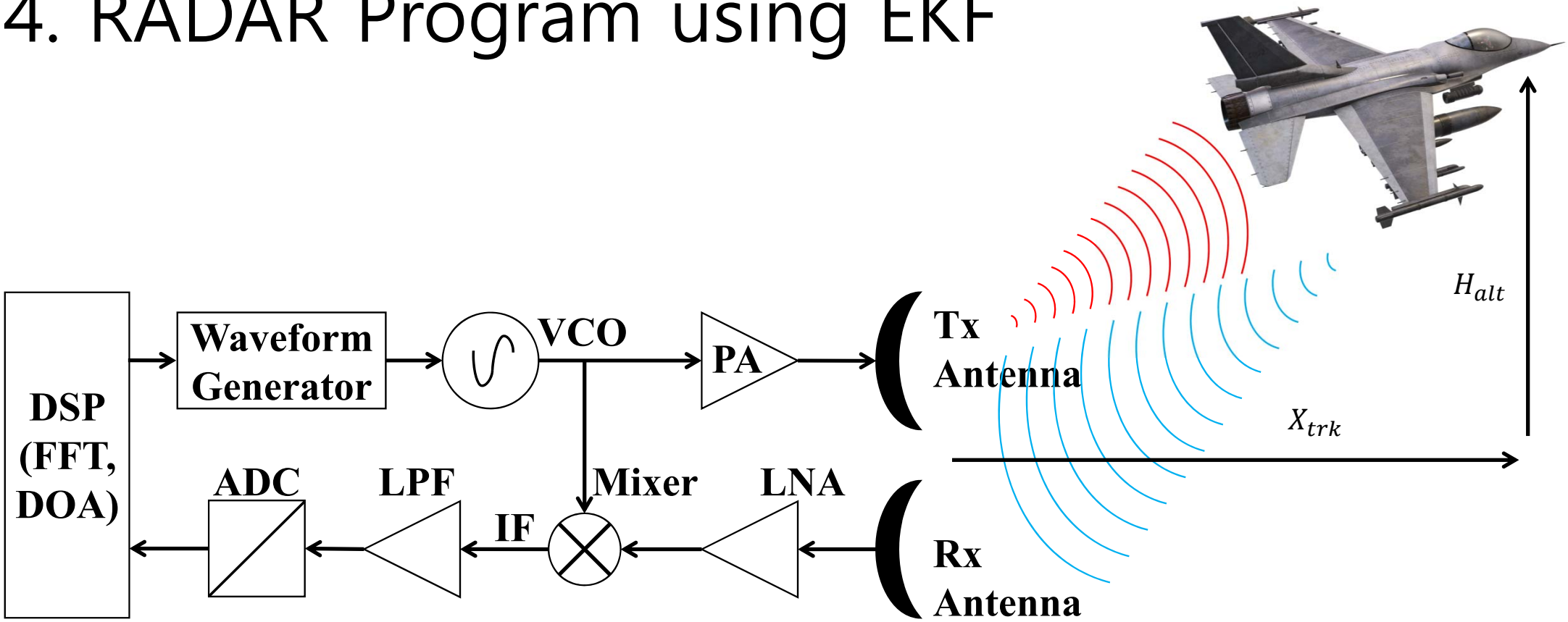


4. RADAR Program using EKF

4. RADAR Program using EKF



4. RADAR Program using EKF



4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분

$$\dot{x}_1 = x_2$$

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분 $\dot{x}_1 = x_2$

이동속도 미분 $\dot{x}_2 = w_1$

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분	이동속도 미분	고도 미분
$\dot{x}_1 = x_2$	$\dot{x}_2 = w_1$	$\dot{x}_3 = w_2$

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분	이동속도 미분	고도 미분
$\dot{x}_1 = x_2$	$\dot{x}_2 = w_1$	$\dot{x}_3 = w_2$

측정 모델 설계

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분	이동속도 미분	고도 미분
$\dot{x}_1 = x_2$	$\dot{x}_2 = w_1$	$\dot{x}_3 = w_2$

측정 모델 설계

- 레이더가 측정하는 값은 이동 물체까지의 직선 거리이다.

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

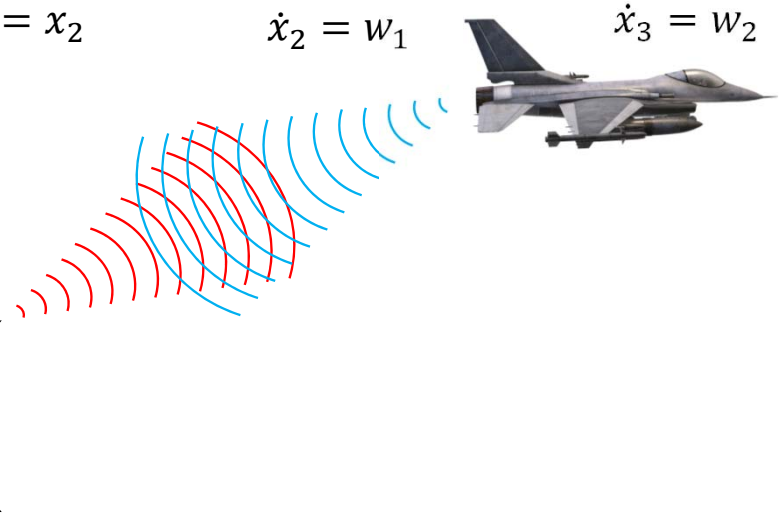
물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분 $\dot{x}_1 = x_2$

이동속도 미분 $\dot{x}_2 = w_1$

고도 미분 $\dot{x}_3 = w_2$



측정 모델 설계

- 레이더가 측정하는 값은 이동 물체까지의 직선 거리이다.

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

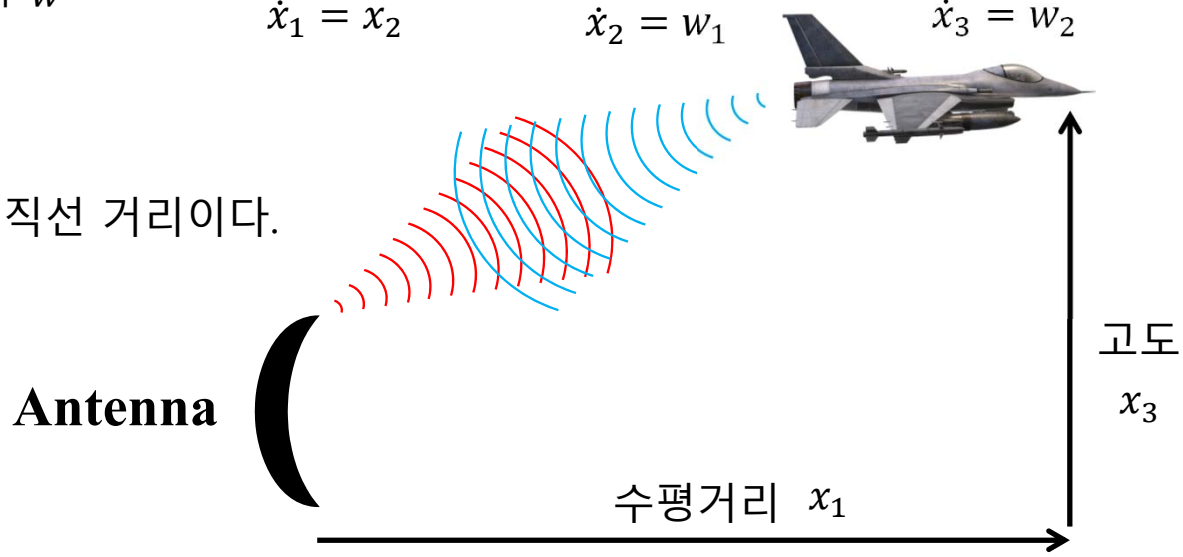
물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분 $\dot{x}_1 = x_2$

이동속도 미분 $\dot{x}_2 = w_1$

고도 미분 $\dot{x}_3 = w_2$



측정 모델 설계

- 레이더가 측정하는 값은 이동 물체까지의 직선 거리이다.

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

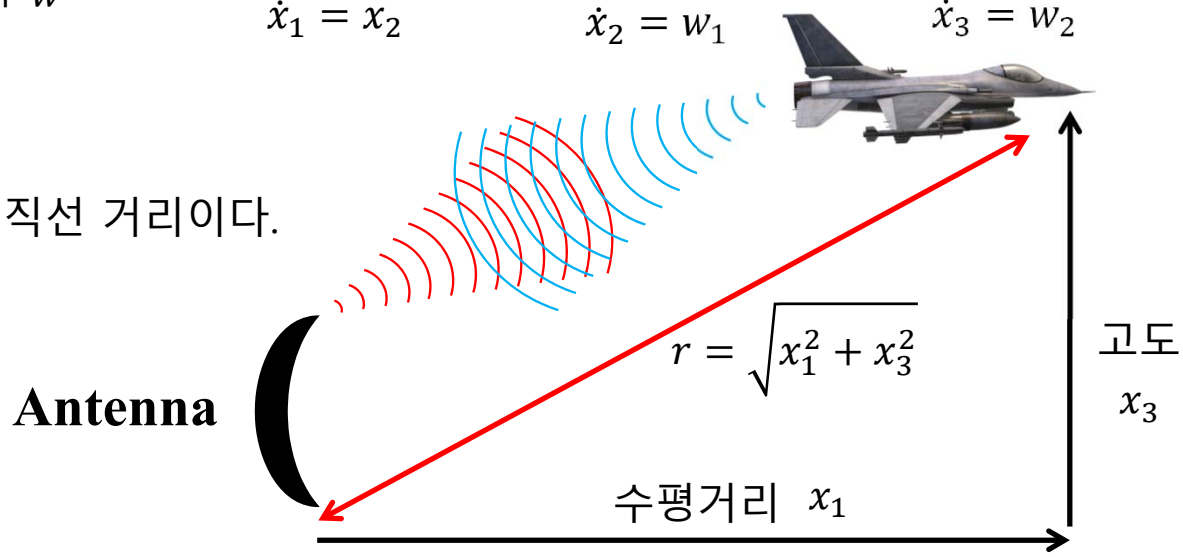
물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분 $\dot{x}_1 = x_2$

이동속도 미분 $\dot{x}_2 = w_1$

고도 미분 $\dot{x}_3 = w_2$



측정 모델 설계

- 레이더가 측정하는 값은 이동 물체까지의 직선 거리이다.

4. RADAR Program using EKF

시스템의 상태변수는 다음과 같이 정의한다.

$$x = \begin{Bmatrix} \text{수평거리} \\ \text{이동속도} \\ \text{고도} \end{Bmatrix} \equiv \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

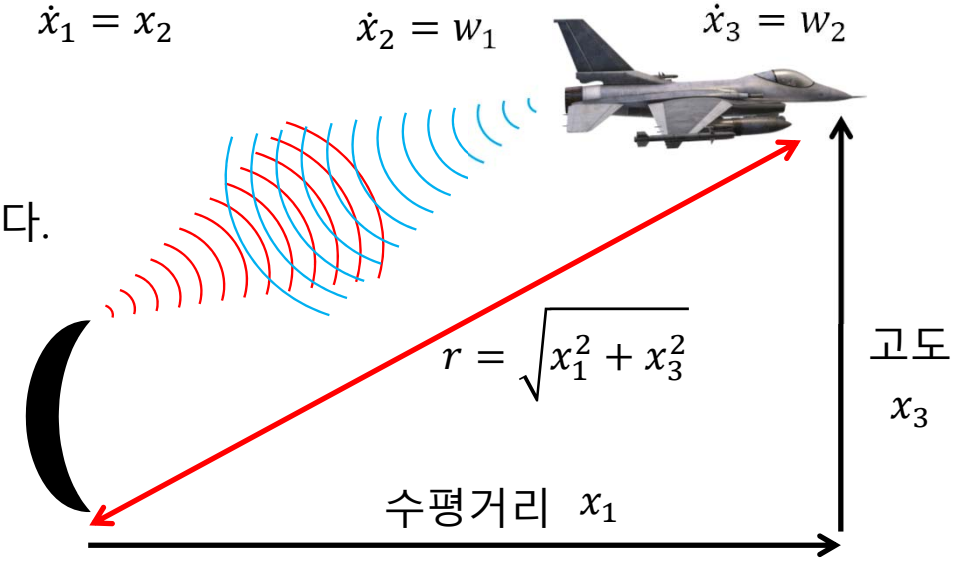
물체의 이동 속도와 고도는 일정하므로 시스템의 운동 방정식은 다음과 같다.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

수평거리 미분 $\dot{x}_1 = x_2$

이동속도 미분 $\dot{x}_2 = w_1$

고도 미분 $\dot{x}_3 = w_2$



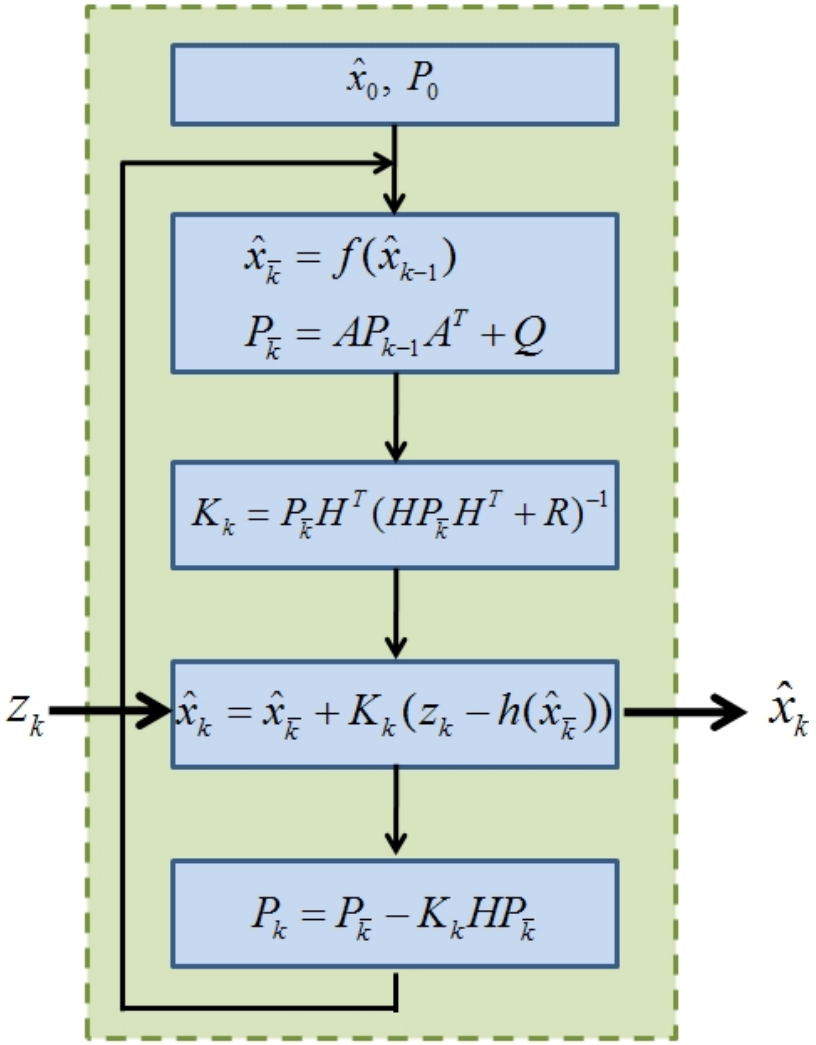
측정 모델 설계

- 레이더가 측정하는 값은 이동 물체까지의 직선 거리이다.

$$r = \sqrt{x_1^2 + x_3^2} + v \equiv h(x) + v$$

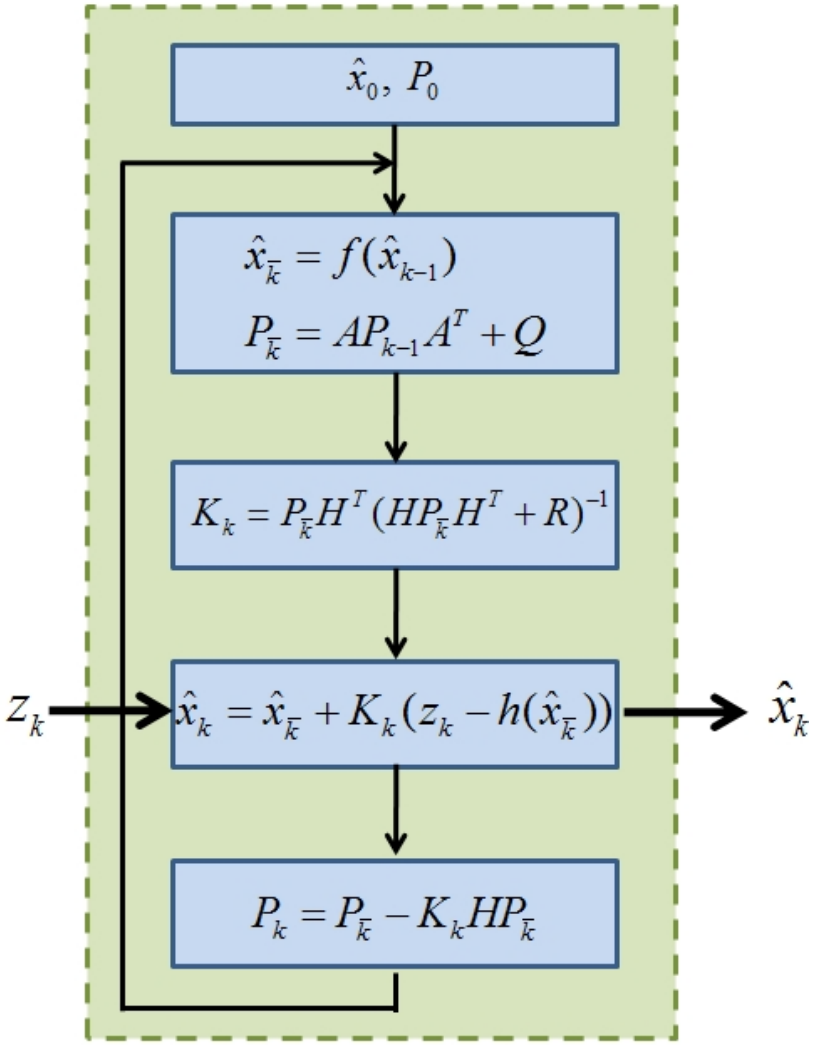
4. RADAR Program using EKF

4. RADAR Program using EKF



EKF algorithm

4. RADAR Program using EKF



EKF algorithm

$x_{k+1} = f(x_k) + w_k$ $Q = w_k$의 공분산 행렬

$z_k = h(x_k) + v_k$ $R = v_k$의 공분산 행렬

x_k는 상태 변수

z_k는 측정값

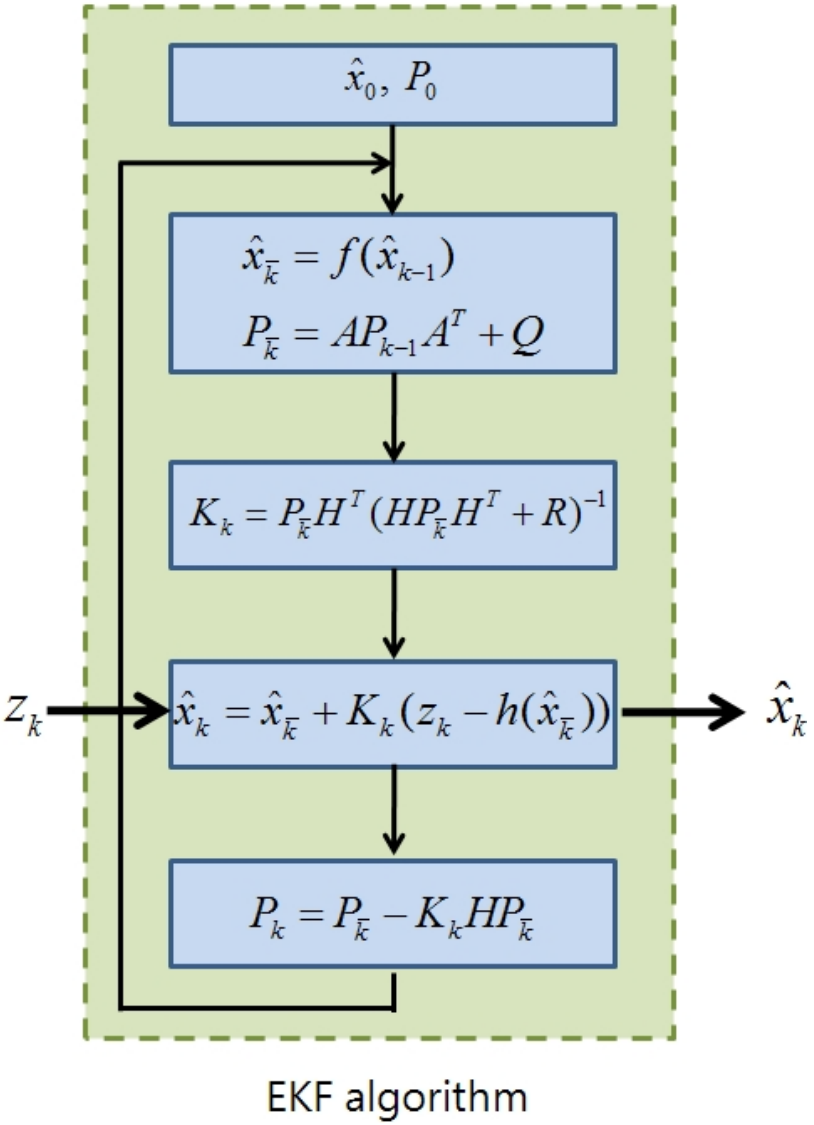
A는 상태전이행렬

H는 mXn 행렬

w_k는 잡음

w_k는 측정 잡음

4. RADAR Program using EKF



EKF algorithm

$x_{k+1} = f(x_k) + w_k$ $Q = w_k$의 공분산 행렬

$z_k = h(x_k) + v_k$ $R = v_k$의 공분산 행렬

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

x_k는 상태 변수

z_k는 측정값

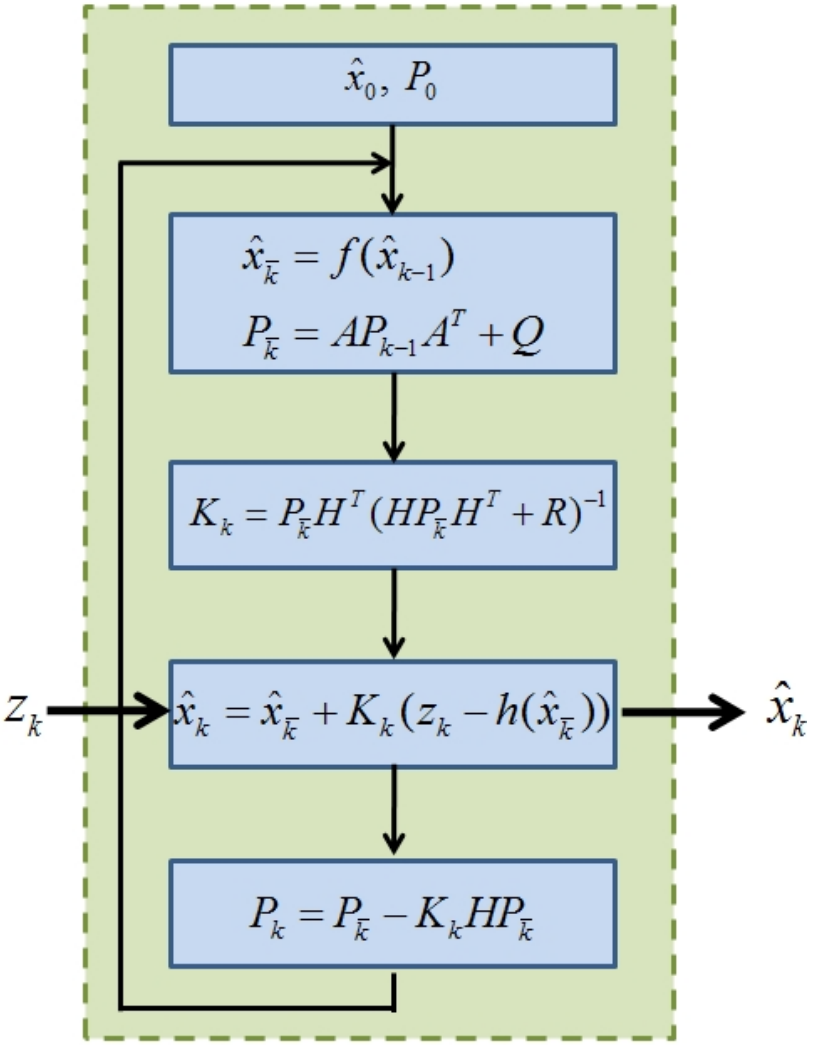
A는 상태전이행렬

H는 mXn 행렬

w_k는 잡음

w_k는 측정 잡음

4. RADAR Program using EKF



EKF algorithm

$x_{k+1} = f(x_k) + w_k$ $Q = w_k$의 공분산 행렬

$z_k = h(x_k) + v_k$ $R = v_k$의 공분산 행렬

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

x_k는 상태 변수

z_k는 측정값

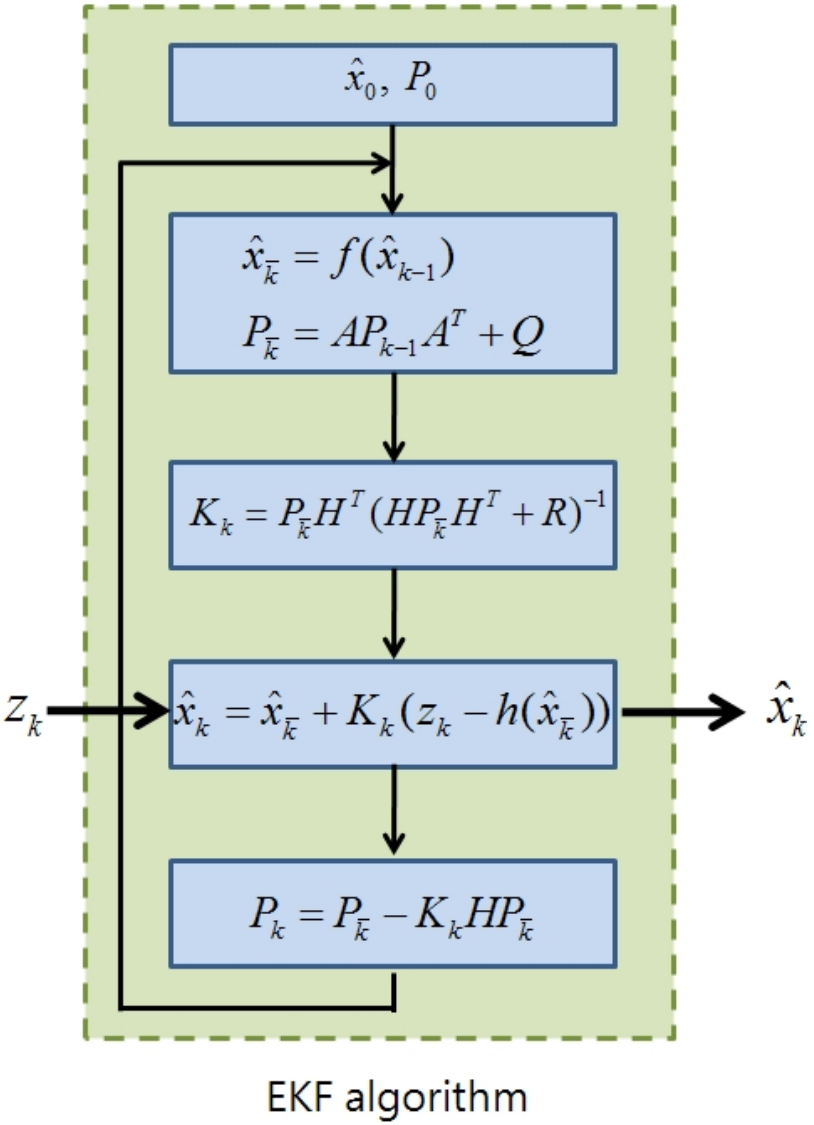
A는 상태전이행렬

H는 mXn 행렬

w_k는 잡음

w_k는 측정 잡음

4. RADAR Program using EKF



EKF algorithm

$x_{k+1} = f(x_k) + w_k$ $\quad Q = w_k$의 공분산 행렬

$z_k = h(x_k) + v_k$ $\quad R = v_k$의 공분산 행렬

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

x_k는 상태 변수

z_k는 측정값

A는 상태전이행렬

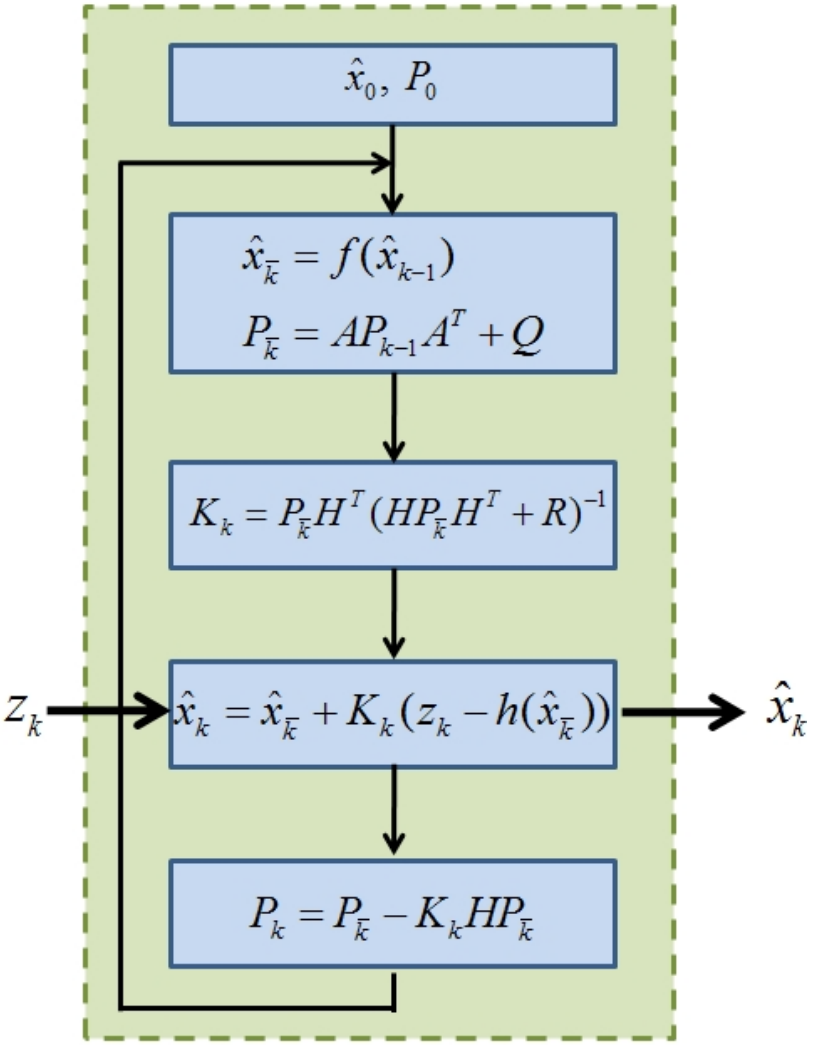
H는 mXn 행렬

w_k는 잡음

w_k는 측정 잡음

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

4. RADAR Program using EKF



EKF algorithm

$x_{k+1} = f(x_k) + w_k$ $\quad Q = w_k$의 공분산 행렬

$z_k = h(x_k) + v_k$ $\quad R = v_k$의 공분산 행렬

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

x_k는 상태 변수

z_k는 측정값

A는 상태전이행렬

H는 mXn 행렬

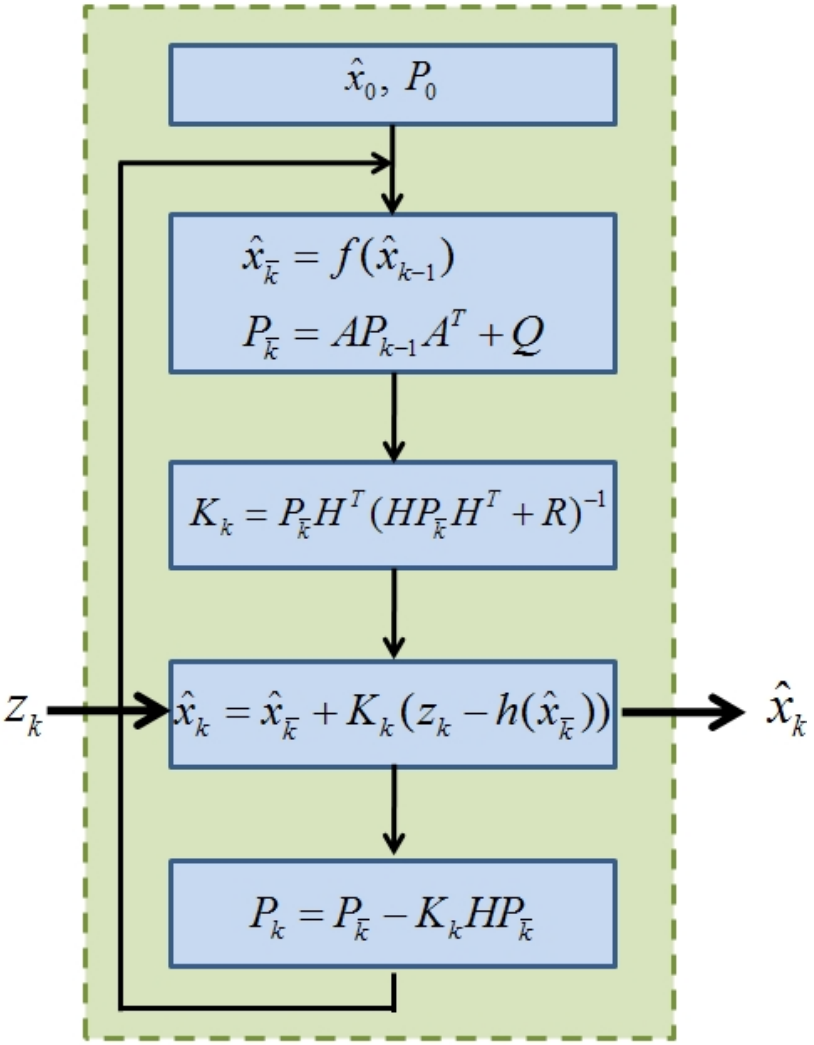
w_k는 잡음

w_k는 측정 잡음

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

$$r = \sqrt{x_1^2 + x_3^2} + v \equiv h(x) + v$$

4. RADAR Program using EKF



EKF algorithm

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

$z_k = h(x_k) + v_k$

$Q = w_k$의 공분산 행렬

$R = v_k$의 공분산 행렬

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

x_k는 상태 변수

z_k는 측정값

A는 상태전이행렬

H는 mXn 행렬

w_k는 잡음

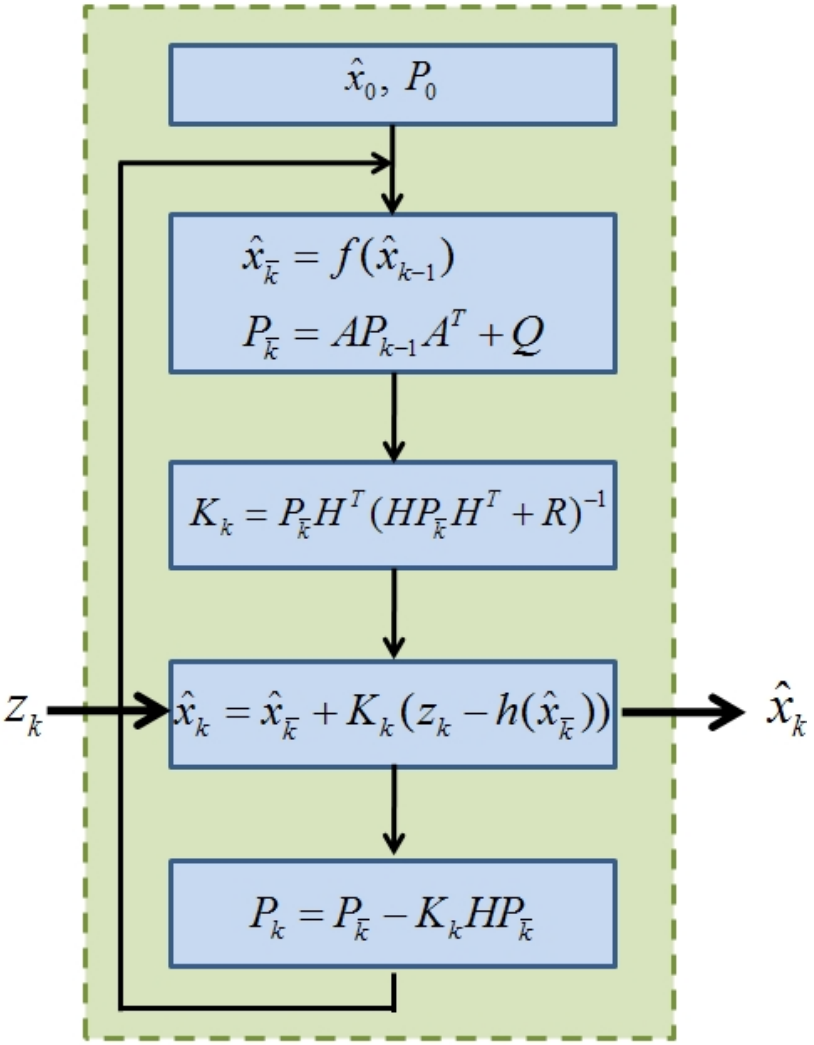
w_k는 측정 잡음

$$\begin{Bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{Bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{Bmatrix}x_1\\ x_2\\ x_3\end{Bmatrix} + \begin{Bmatrix}0\\ w_1\\ w_2\end{Bmatrix} \equiv Ax + w$$

$$A = I + \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \times dt$$

$$r = \sqrt{x_1^2 + x_3^2} + v \equiv h(x) + v$$

4. RADAR Program using EKF



EKF algorithm

$x_{k+1} = f(x_k) + w_k$ $\quad Q = w_k$의 공분산 행렬

$z_k = h(x_k) + v_k$ $\quad R = v_k$의 공분산 행렬

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

x_k는 상태 변수

z_k는 측정값

A는 상태전이행렬

H는 mXn 행렬

w_k는 잡음

w_k는 측정 잡음

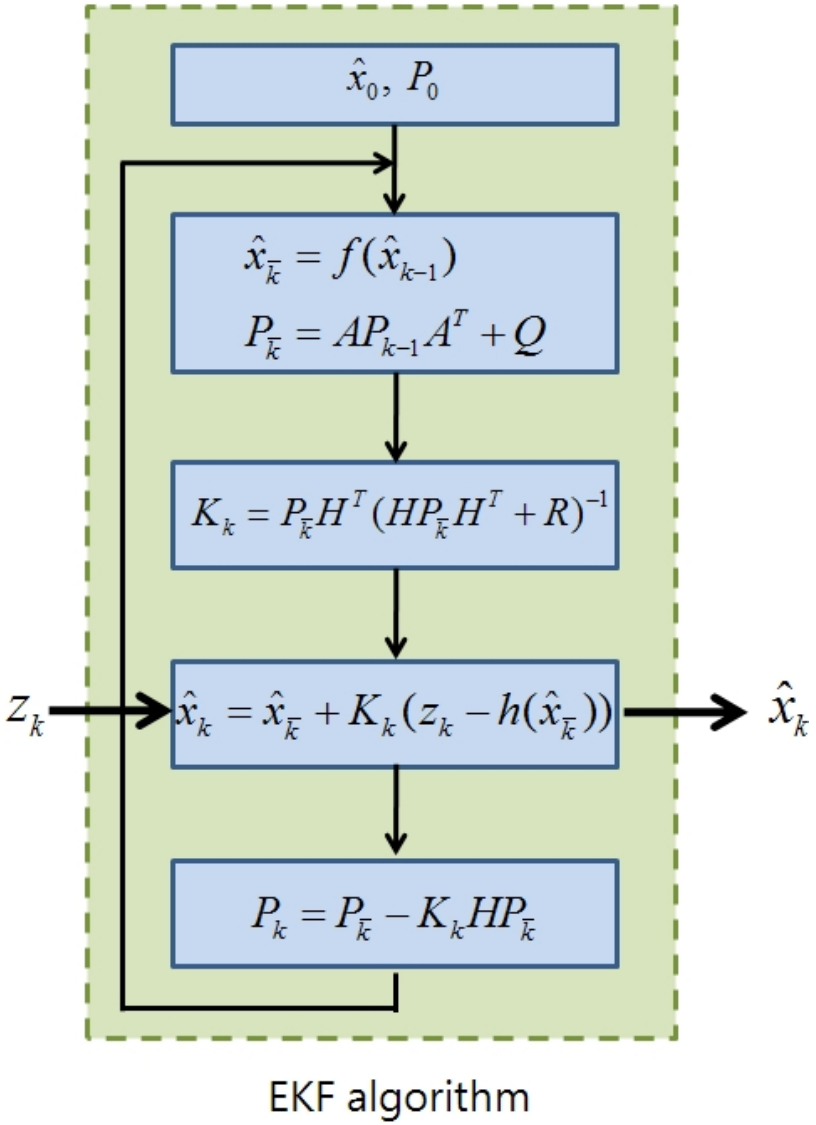
$$\begin{Bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{Bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{Bmatrix}x_1\\ x_2\\ x_3\end{Bmatrix} + \begin{Bmatrix}0\\ w_1\\ w_2\end{Bmatrix} \equiv Ax + w$$

$$A = I + \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \times dt$$

$$r = \sqrt{x_1^2 + x_3^2} + v \equiv h(x) + v$$

$$H = \begin{bmatrix}\frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} & \frac{\partial h}{\partial x_3}\end{bmatrix}$$

4. RADAR Program using EKF



EKF algorithm

$x_{k+1} = f(x_k) + w_k$ $\quad Q = w_k$의 공분산 행렬

$z_k = h(x_k) + v_k$ $\quad R = v_k$의 공분산 행렬

$f(\hat{x}_{k-1}) \Leftrightarrow A\hat{x}_{k-1}$

$h(\hat{x}_k^-) \Leftrightarrow H\hat{x}_k^-$

x_k는 상태 변수

z_k는 측정값

A는 상태전이행렬

H는 mXn 행렬

w_k는 잡음

w_k는 측정 잡음

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{Bmatrix} 0 \\ w_1 \\ w_2 \end{Bmatrix} \equiv Ax + w$$

$$A = I + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times dt$$

$$r = \sqrt{x_1^2 + x_3^2} + v \equiv h(x) + v$$

$$H = \begin{bmatrix} \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} & \frac{\partial h}{\partial x_3} \end{bmatrix} = \begin{bmatrix} \frac{x_1}{\sqrt{x_1^2 + x_3^2}} & 0 & \frac{x_3}{\sqrt{x_1^2 + x_3^2}} \end{bmatrix}$$

4. RADAR Program using EKF

```
clear all

dt = 0.05;
t  = 0:dt:20;

Nsamples = length(t);

Xsaved = zeros(Nsamples, 3);
Zsaved = zeros(Nsamples, 1);

for k=1:Nsamples
  r = GetRadar(dt);

  [pos vel alt] = RadarEKF(r, dt);

  Xsaved(k, :) = [pos vel alt];
  Zsaved(k)    = r;
end
```

```
PosSaved = Xsaved(:, 1);
VelSaved = Xsaved(:, 2);
AltSaved = Xsaved(:, 3);

t = 0:dt:Nsamples*dt-dt;

figure
plot(t, PosSaved)

figure
plot(t, VelSaved)

figure
plot(t, AltSaved)
```

4. RADAR Program using EKF

```
function r = GetRadar(dt)

persistent posp

if isempty(posp)
  posp = 0; % 초기값
end


vel = 100  +  5*randn; % 100m/s의 속도 + 오차범위 5%
alt = 1000 + 10*randn; % 1Km의 고도 + 오차범위 10%

pos = posp + vel*dt;

v = 0 + pos*0.05*randn;

r = sqrt(pos^2 + alt^2) + v;

posp = pos;
```

4. RADAR Program using EKF

```
function [pos vel alt] = RadarEKF(z, dt)
persistent A Q R
persistent x P
persistent firstRun

if isempty(firstRun)
  A = eye(3) + dt*[ 0 1 0;
                    0 0 0;
                    0 0 0 ];
  Q = [ 0  0      0;
        0  0.001  0;
        0  0      0.001 ];
  R = 10;
  x = [0 90 1100]';
  P = 10*eye(3);

  firstRun = 1;
end

H = Hjacob(x);
xp = A*x;
Pp = A*P*A' + Q;
K = Pp*H'*inv(H*Pp*H' + R);
x = xp + K*(z - hx(xp));
P = Pp - K*H*Pp;

pos = x(1);
vel = x(2);
alt = x(3);
%------------------------------
function zp = hx(xhat)

x1 = xhat(1);
x3 = xhat(3);

zp = sqrt(x1^2 + x3^2);
%------------------------------
function H = Hjacob(xp)

H = zeros(1, 3);

x1 = xp(1);
x3 = xp(3);

H(1) = x1 / sqrt(x1^2 + x3^2);
H(2) = 0;
H(3) = x3 / sqrt(x1^2 + x3^2);
```

4. RADAR Program using EKF

```
function [pos vel alt] = RadarEKF(z, dt)
persistent A Q R
persistent x P
persistent firstRun

if isempty(firstRun)
  A = eye(3) + dt*[ 0 1 0;
                    0 0 0;
                    0 0 0 ];
  Q = [ 0  0      0;
        0  0.001  0;
        0  0      0.001 ];
  R = 10;
  x = [0 90 1100]';
  P = 10*eye(3);

  firstRun = 1;
end

H = Hjacob(x);
xp = A*x;
Pp = A*P*A' + Q;
K = Pp*H'*inv(H*Pp*H' + R);
```

$$A = I + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times dt$$

```
x = xp + K*(z - hx(xp));
P = Pp - K*H*Pp;

pos = x(1);
vel = x(2);
alt = x(3);
%-----------------------------
function zp = hx(xhat)

x1 = xhat(1);
x3 = xhat(3);

zp = sqrt(x1^2 + x3^2);
%-----------------------------
function H = Hjacob(xp)

H = zeros(1, 3);

x1 = xp(1);
x3 = xp(3);

H(1) = x1 / sqrt(x1^2 + x3^2);
H(2) = 0;
H(3) = x3 / sqrt(x1^2 + x3^2);
```

4. RADAR Program using EKF

```
function [pos vel alt] = RadarEKF(z, dt)
persistent A Q R
persistent x P
persistent firstRun

if isempty(firstRun)
  A = eye(3) + dt*[ 0 1 0;
                    0 0 0;
                    0 0 0 ];
  Q = [ 0  0      0;
        0  0.001  0;
        0  0      0.001 ];
  R = 10;
  x = [0 90 1100]';
  P = 10*eye(3);

  firstRun = 1;
end

H = Hjacob(x);
xp = A*x;
Pp = A*P*A' + Q;
K = Pp*H'*inv(H*Pp*H' + R);
```

$$A = I + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times dt$$

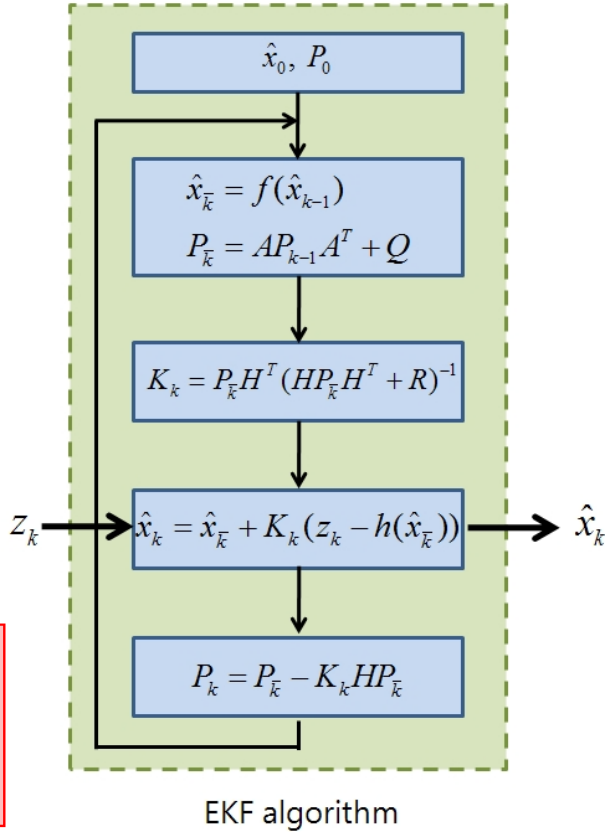
측정값의 영향을 덜 받고 변화가 완만한 추정값을 얻고 싶다면 행렬 R을 키우고 Q를 줄려야한다.

```
x = xp + K*(z - hx(xp));
P = Pp - K*H*Pp;

pos = x(1);
vel = x(2);
alt = x(3);
%------------------------------
function zp = hx(xhat)

x1 = xhat(1);
x3 = xhat(3);

zp = sqrt(x1^2 + x3^2);
%------------------------------
function H = Hjacob(xp)

H = zeros(1, 3);

x1 = xp(1);
x3 = xp(3);

H(1) = x1 / sqrt(x1^2 + x3^2);
H(2) = 0;
H(3) = x3 / sqrt(x1^2 + x3^2);
```

4. RADAR Program using EKF

```
function [pos vel alt] = RadarEKF(z, dt)
persistent A Q R
persistent x P
persistent firstRun

if isempty(firstRun)
  A = eye(3) + dt*[ 0 1 0;
                    0 0 0;
                    0 0 0 ]
  Q = [ 0  0      0;
        0  0.001  0;
        0  0      0.001 ];
  R = 10;
  x = [0 90 1100]';
  P = 10*eye(3);

  firstRun = 1;
end
```

```
H = Hjacob(x);
xp = A*x;
Pp = A*P*A' + Q;
K = Pp*H'*inv(H*Pp*H' + R);
```



EKF algorithm

```
x = xp + K*(z - hx(xp));
P = Pp - K*H*Pp;
```

```
pos = x(1);
vel = x(2);
alt = x(3);
%-----------------------------
function zp = hx(xhat)

x1 = xhat(1);
x3 = xhat(3);

zp = sqrt(x1^2 + x3^2);
%-----------------------------
function H = Hjacob(xp)

H = zeros(1, 3);

x1 = xp(1);
x3 = xp(3);

H(1) = x1 / sqrt(x1^2 + x3^2);
H(2) = 0;
H(3) = x3 / sqrt(x1^2 + x3^2);
```

4. RADAR Program using EKF

```
function [pos vel alt] = RadarEKF(z, dt)
persistent A Q R
persistent x P
persistent firstRun

if isempty(firstRun)
  A = eye(3) + dt*[ 0 1 0;
                    0 0 0;
                    0 0 0 ];
  Q = [ 0  0      0;
        0  0.001  0;
        0  0      0.001 ];
  R = 10;
  x = [0 90 1100]';
  P = 10*eye(3);

  firstRun = 1;
end

H = Hjacob(x);
xp = A*x;
Pp = A*P*A' + Q;
K = Pp*H'*inv(H*Pp*H' + R);
x = xp + K*(z - hx(xp));
P = Pp - K*H*Pp;

pos = x(1);
vel = x(2);
alt = x(3);
%-----------------------------
function zp = hx(xhat)

x1 = xhat(1);
x3 = xhat(3);

zp = sqrt(x1^2 + x3^2);
%-----------------------------
function H = Hjacob(xp)

H = zeros(1, 3);

x1 = xp(1);
x3 = xp(3);

H(1) = x1 / sqrt(x1^2 + x3^2);
H(2) = 0;
H(3) = x3 / sqrt(x1^2 + x3^2);
```

$$r = \sqrt{x_1^2 + x_3^2} + v$$

4. RADAR Program using EKF

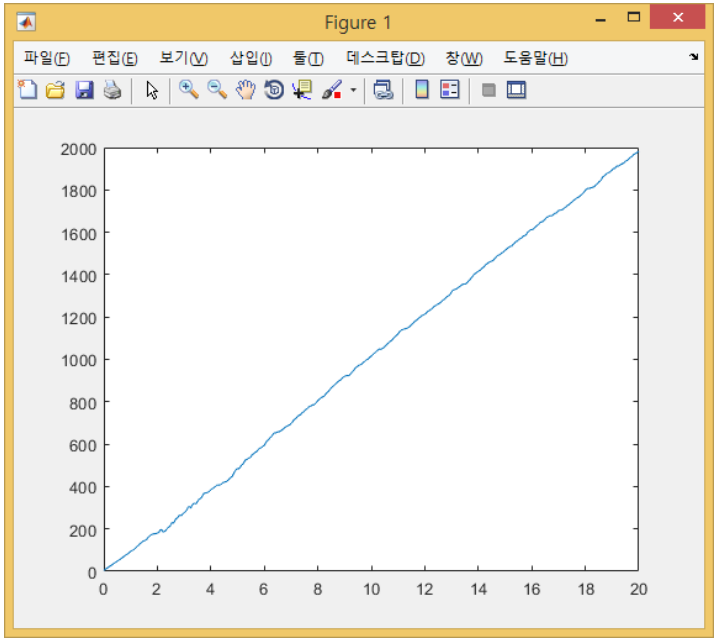
```
function [pos vel alt] = RadarEKF(z, dt)
persistent A Q R
persistent x P
persistent firstRun

if isempty(firstRun)
  A = eye(3) + dt*[ 0 1 0;
                    0 0 0;
                    0 0 0 ];
  Q = [ 0  0      0;
        0  0.001  0;
        0  0      0.001 ];
  R = 10;
  x = [0 90 1100]';
  P = 10*eye(3);

  firstRun = 1;
end

H = Hjacob(x);
xp = A*x;
Pp = A*P*A' + Q;
K = Pp*H'*inv(H*Pp*H' + R);
x = xp + K*(z - hx(xp));
P = Pp - K*H*Pp;

pos = x(1);
vel = x(2);
alt = x(3);
%------------------------------
function zp = hx(xhat)

x1 = xhat(1);
x3 = xhat(3);

zp = sqrt(x1^2 + x3^2);
%------------------------------
function H = Hjacob(xp)

H = zeros(1, 3);

x1 = xp(1);
x3 = xp(3);

H(1) = x1 / sqrt(x1^2 + x3^2);
H(2) = 0;
H(3) = x3 / sqrt(x1^2 + x3^2);
```

$$r = \sqrt{x_1^2 + x_3^2} + v$$

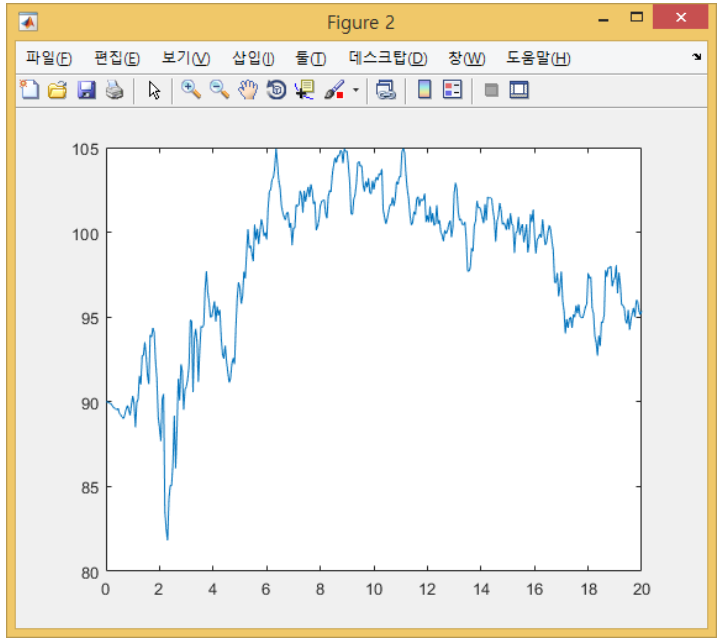
$$H = \begin{bmatrix} \frac{x_1}{\sqrt{x_1^2 + x_3^2}} & 0 & \frac{x_3}{\sqrt{x_1^2 + x_3^2}} \end{bmatrix}$$

4. RADAR Program using EKF

거리



속도



고도