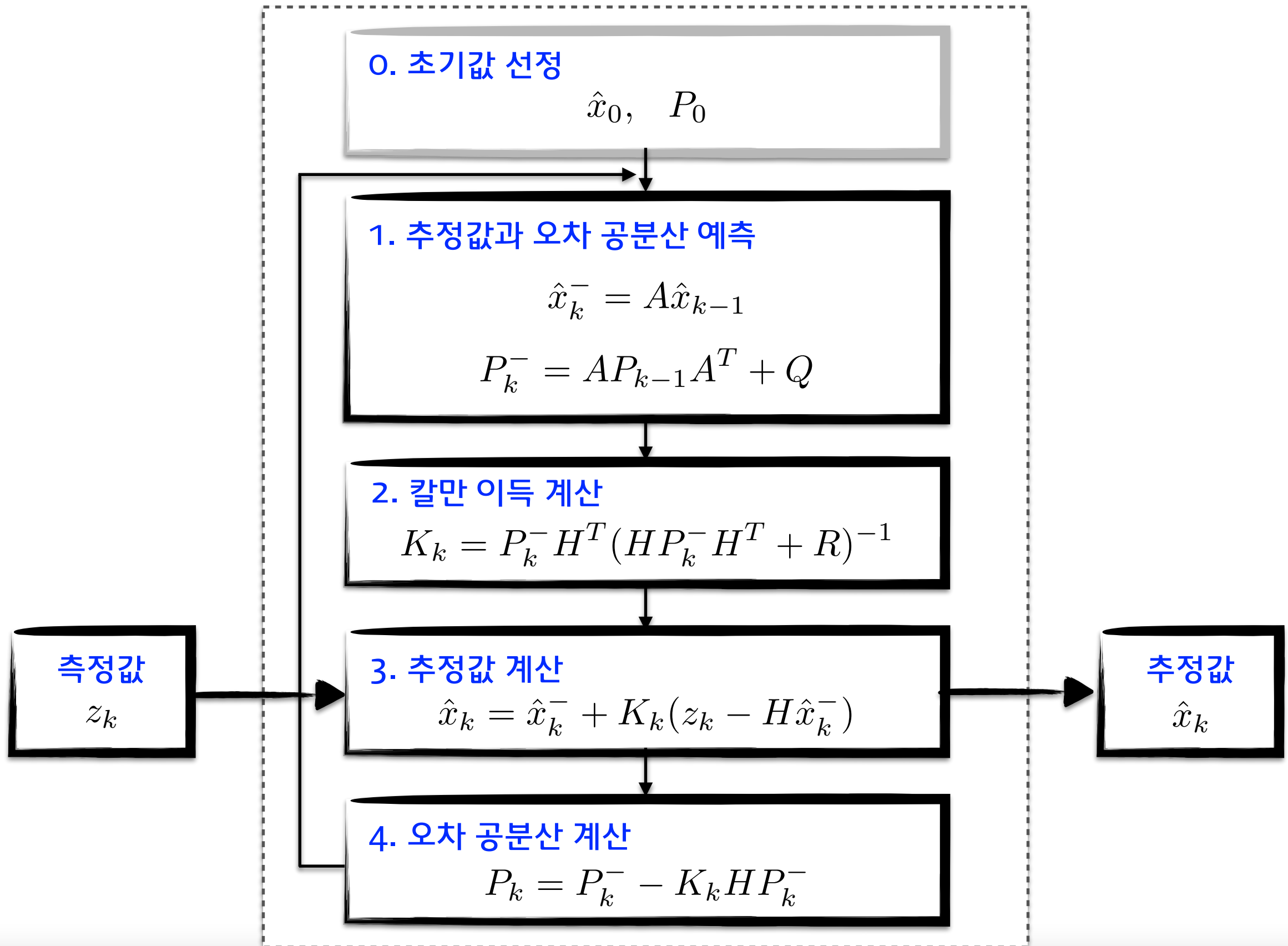


# 위치 속도 추정

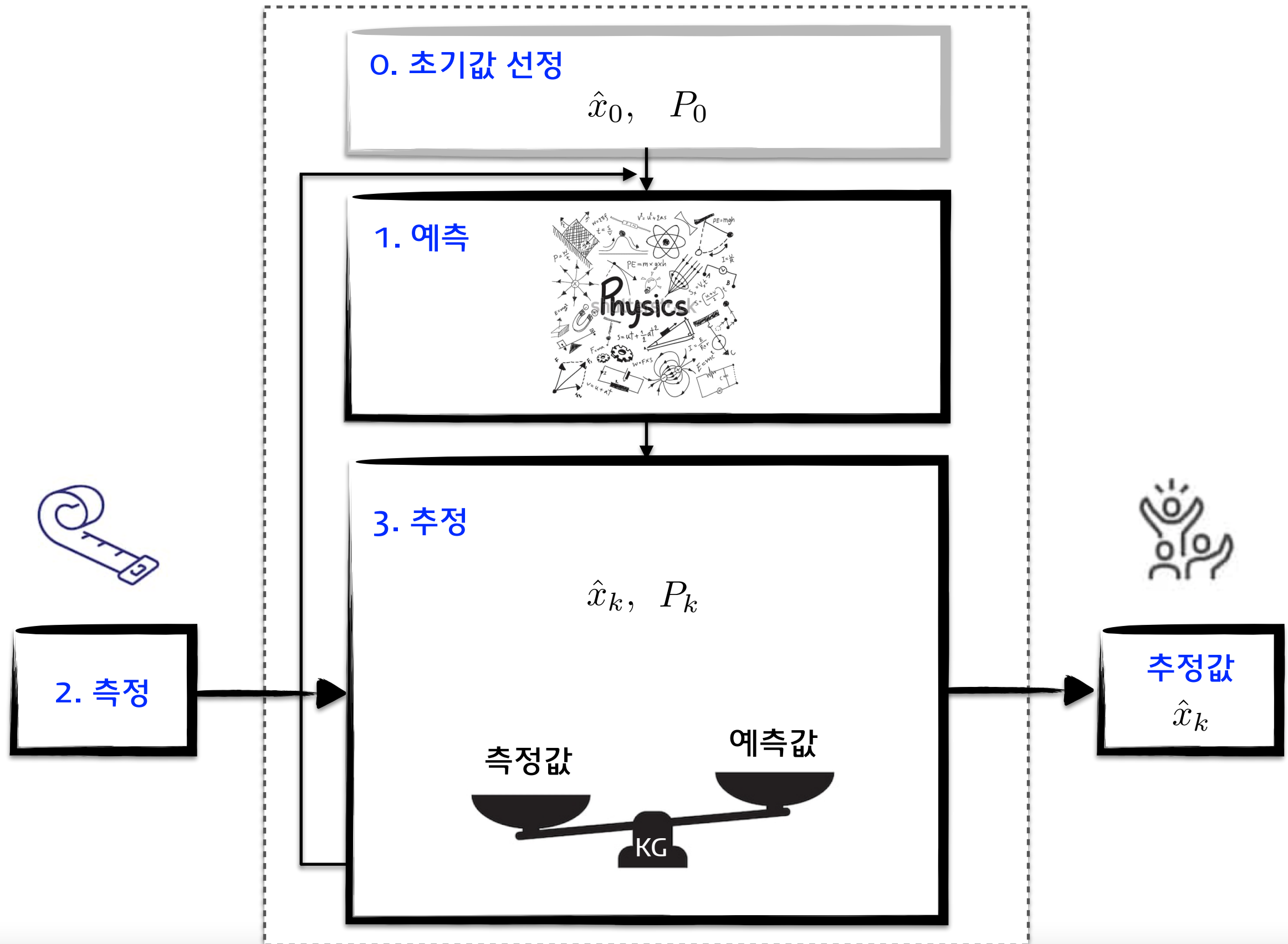
문태봉

2020.01.22 (수)

# 칼만 필터 알고리즘

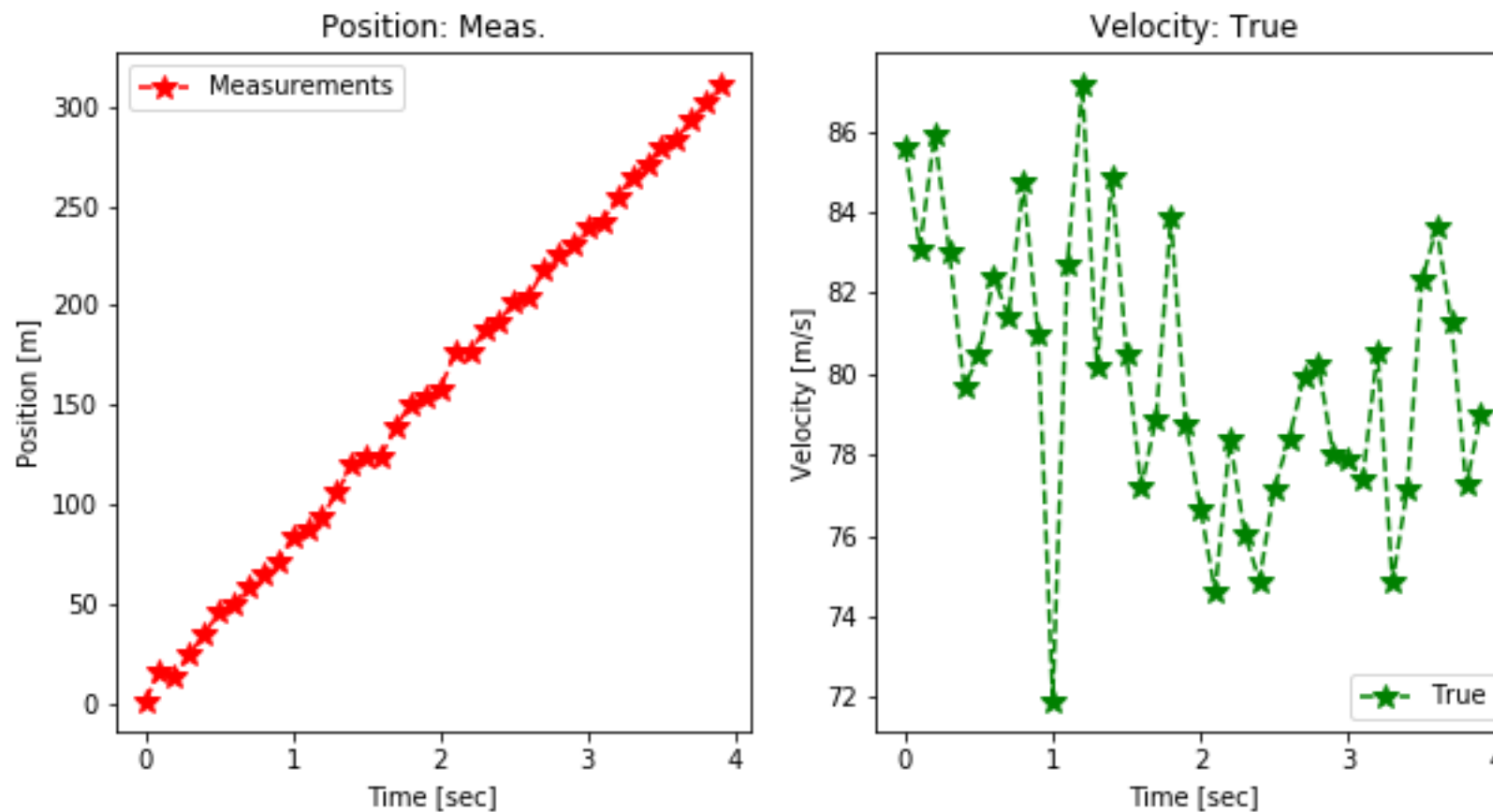


# 칼만 필터 알고리즘



# 칼만 필터: 위치로 속도 추정하기 예제

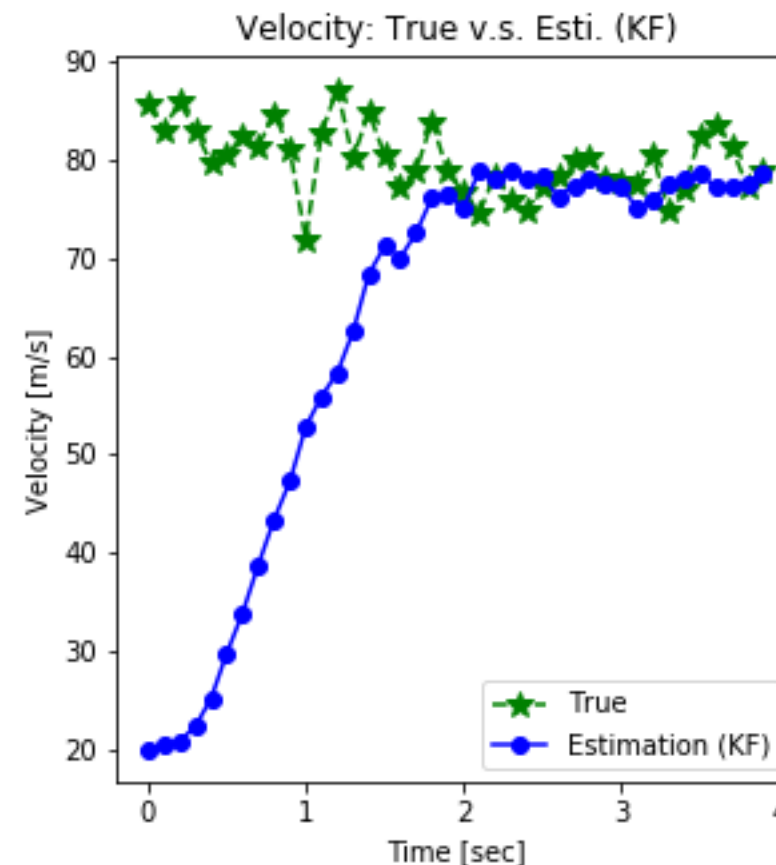
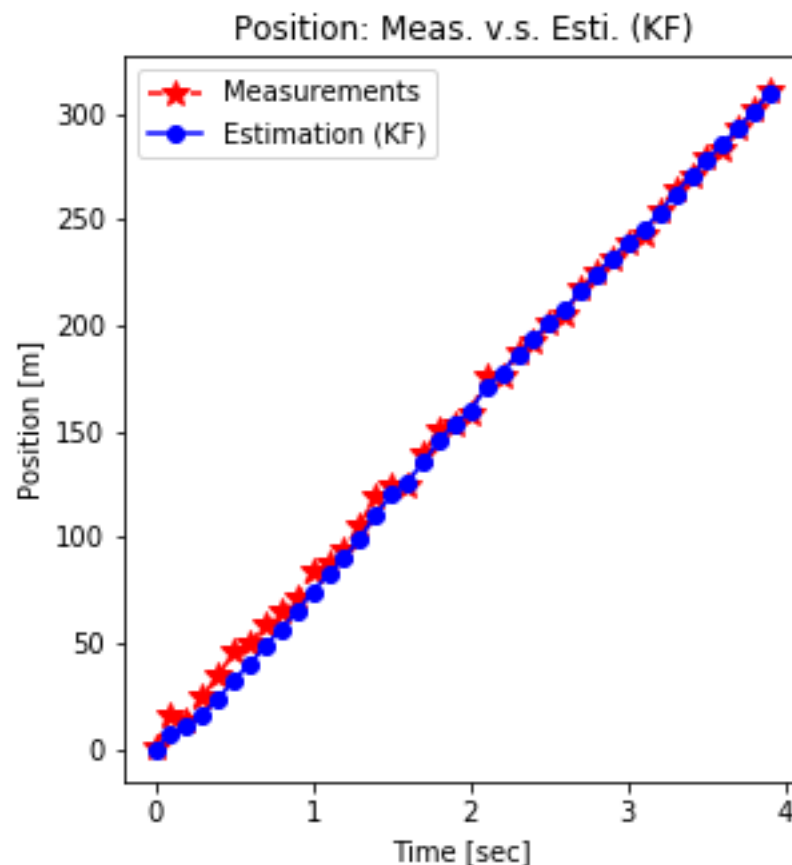
- 4초 동안 0.1초 간격으로 열차의 “위치 측정”
  - 열차의 명목 속도 = 80 m/s
  - 시스템 위치 잡음  $\sim 0$
  - 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
  - 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$
- 칼만 필터를 사용하여 열차의 위치와 “속도를 추정”하자!



[https://github.com/tbmoon/kalman\\_filter/blob/master/Ch09.Pos2VelKF/pos2vel\\_kf.ipynb](https://github.com/tbmoon/kalman_filter/blob/master/Ch09.Pos2VelKF/pos2vel_kf.ipynb)

# 칼만 필터: 위치로 속도 추정하기 예제

- 4초 동안 0.1초 간격으로 열차의 “위치 측정”
  - 열차의 명목 속도 = 80 m/s
  - 시스템 위치 잡음  $\sim 0$
  - 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
  - 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$
- 칼만 필터를 사용하여 열차의 위치와 “속도를 추정”하자!



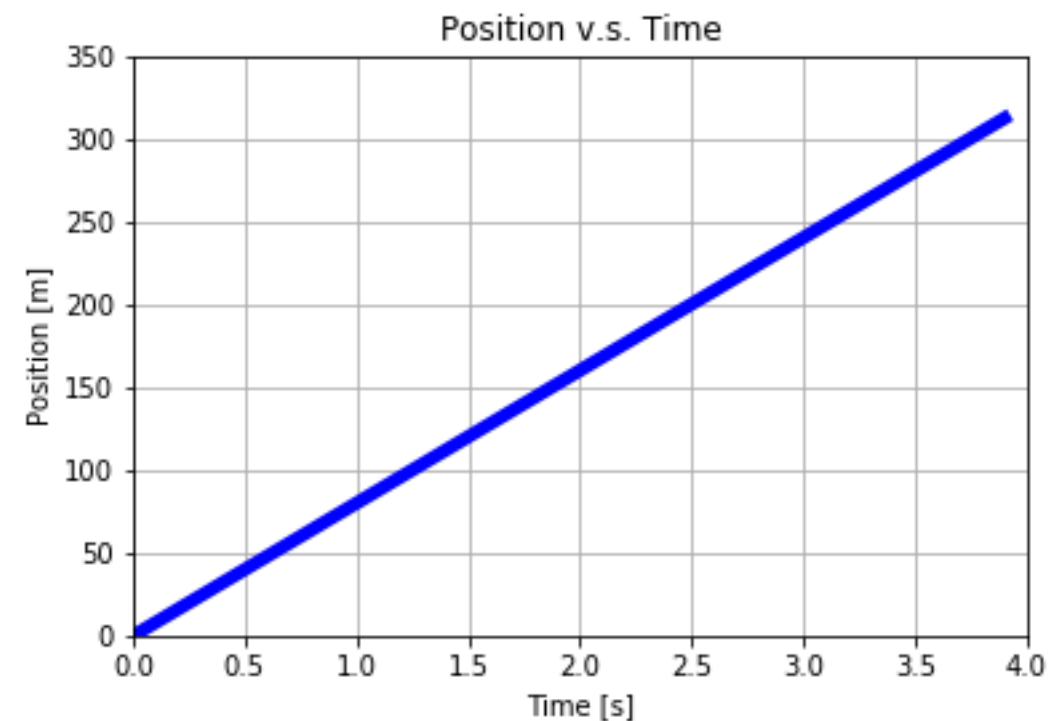
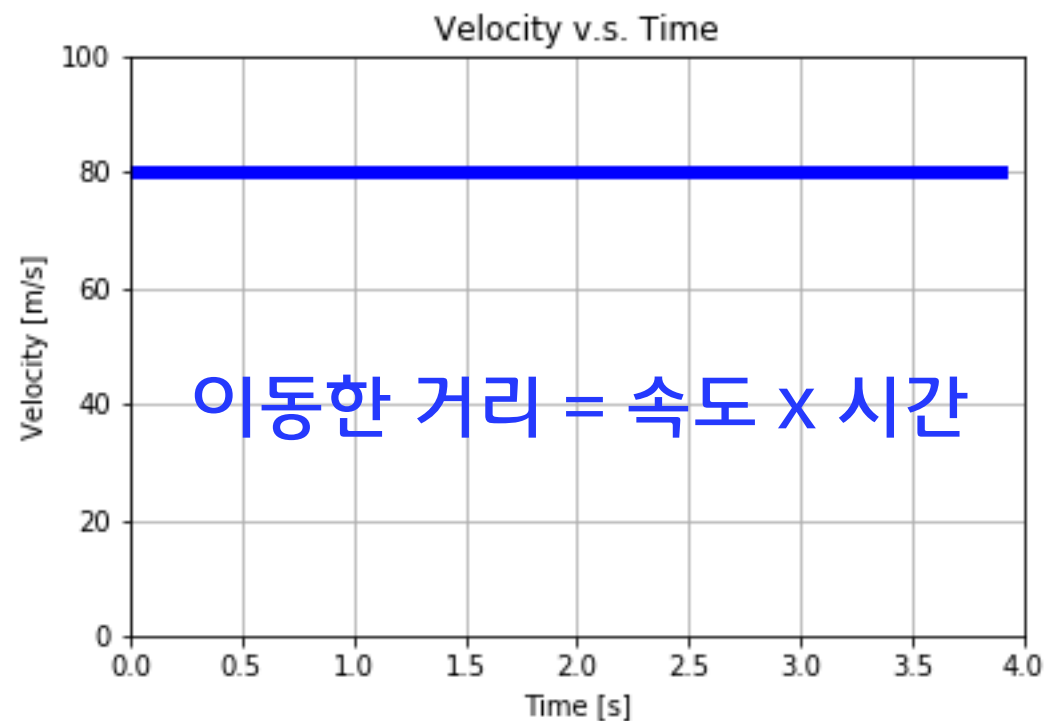
[https://github.com/tbmoon/kalman\\_filter/blob/master/Ch09.Pos2VelKF/pos2vel\\_kf.ipynb](https://github.com/tbmoon/kalman_filter/blob/master/Ch09.Pos2VelKF/pos2vel_kf.ipynb)

# 칼만 필터: 위치로 속도 추정하기 예제

물리식을 이용하여 상태 & 시스템 모델을 만들어 보아요!

# 상태 모델 & 상태 전이 행렬 A

- 등속 운동 (속도 = 80 m/s)
- 시스템 잡음 무시

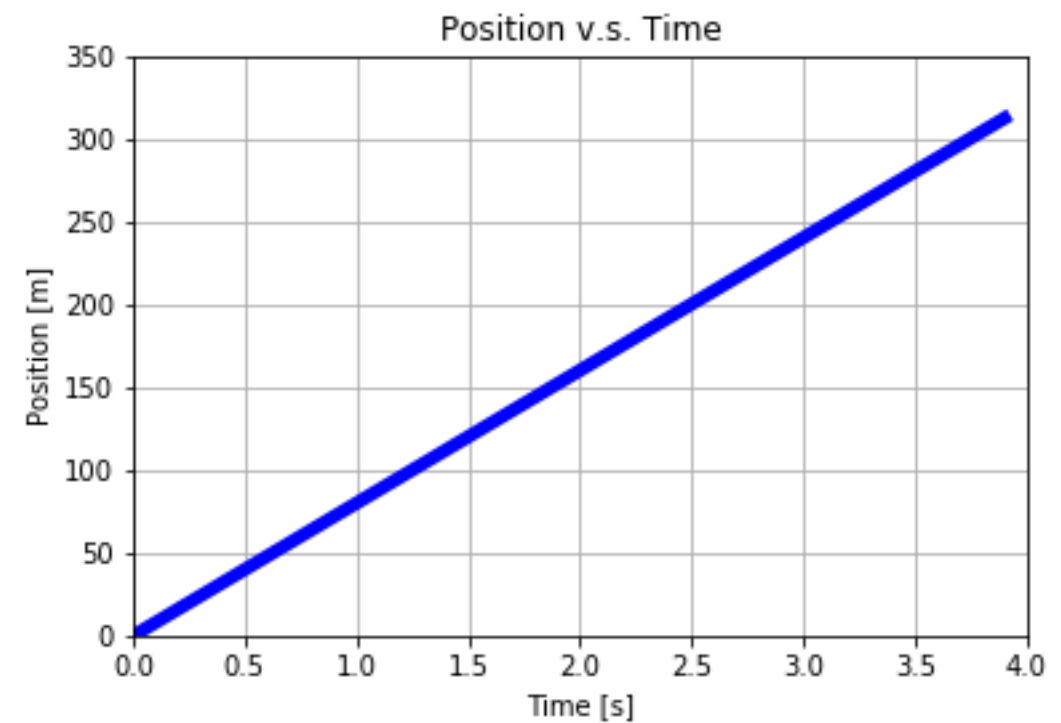
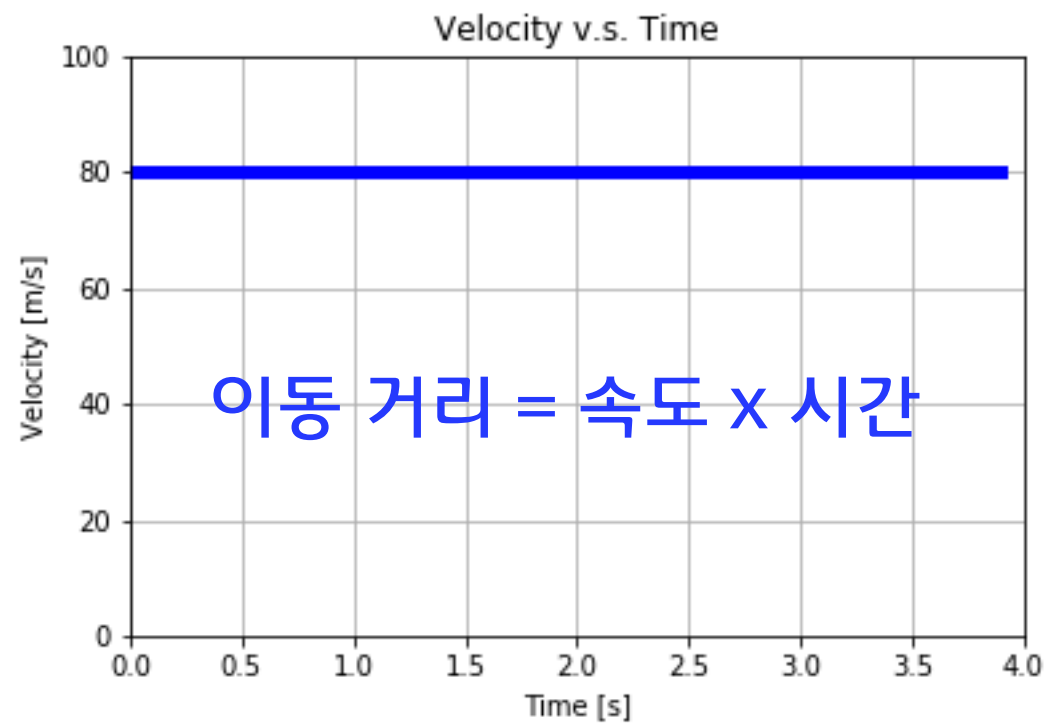


$$\text{위치}_{k+1} = \text{위치}_k + \text{속도}_k \times \Delta t$$

$$\text{속도}_{k+1} = \text{속도}_k$$

# 상태 모델 & 상태 전이 행렬 A

- 등속 운동 (속도 = 80 m/s)
- 시스템 잡음 무시

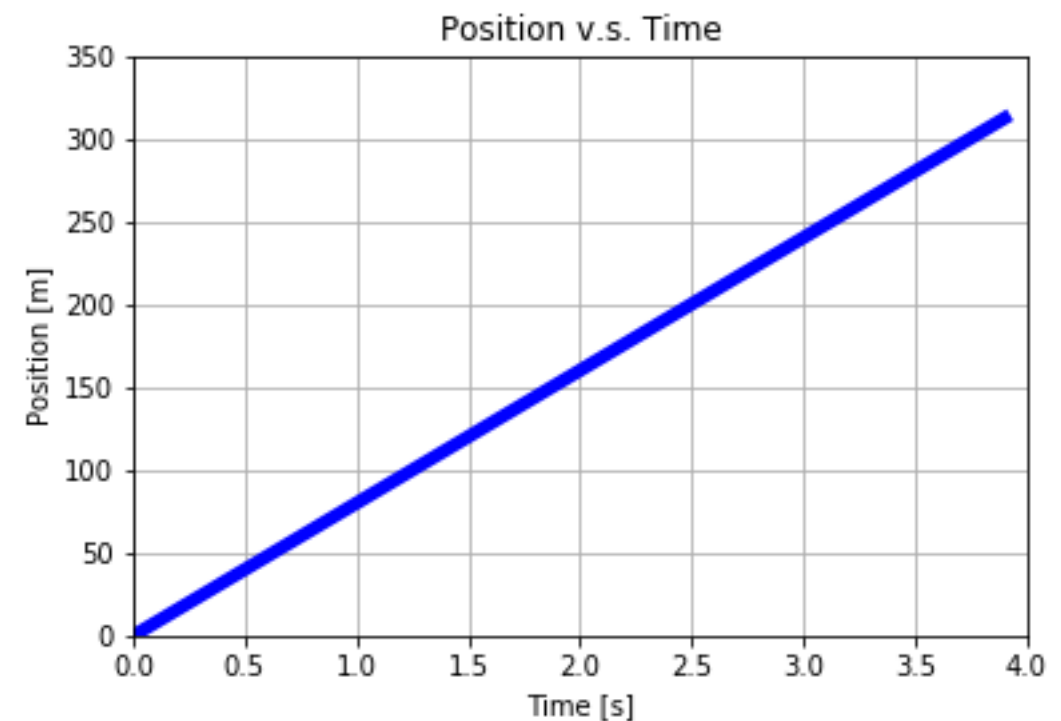
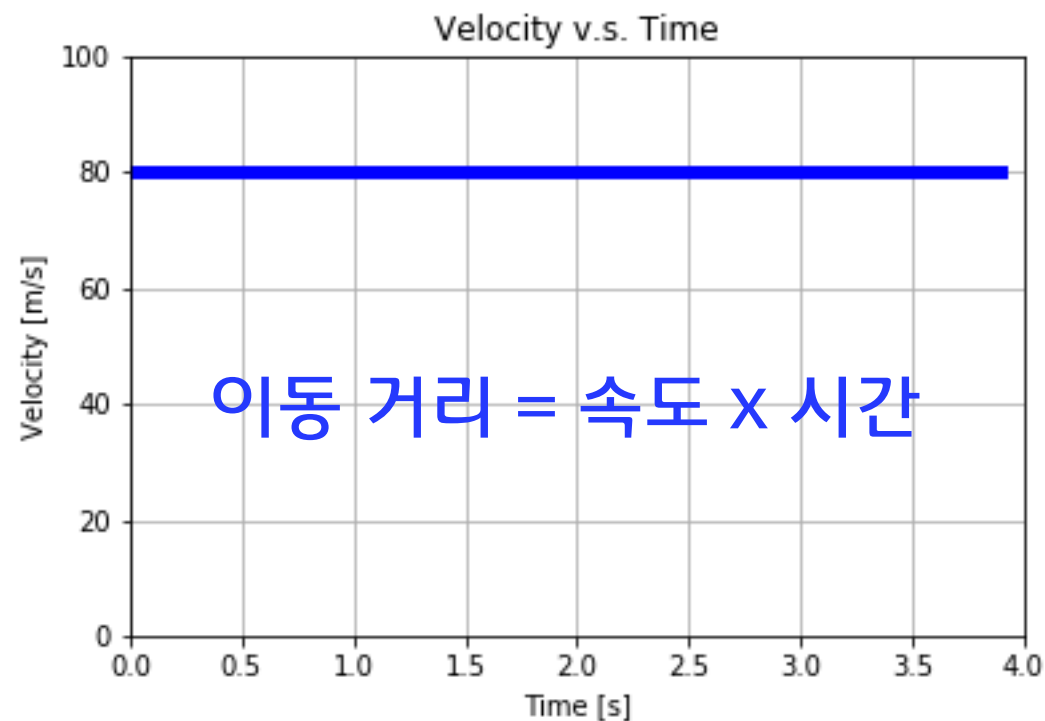


$$\begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_k$$



# 상태 모델 & 상태 전이 행렬 A

- 등속 운동 (속도 = 80 m/s)
- 시스템 잡음 고려



$$\begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_k + \begin{bmatrix} 0 \\ w_k \end{bmatrix}$$

A

# 상태 모델 & 출력 행렬 H

- 4초 동안 0.1초 간격으로 열차의 “위치 측정”
  - 열차의 명목 속도 = 80 m/s
  - 시스템 위치 잡음  $\sim 0$
  - 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
  - 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$
- 칼만 필터를 사용하여 열차의 위치와 “속도를 추정”하자!

측정 변수

상태 변수

# 상태 모델 & 출력 행렬 H

- 4초 동안 0.1초 간격으로 열차의 “위치 측정”

측정 변수

- 열차의 명목 속도 = 80 m/s
- 시스템 위치 잡음  $\sim 0$
- 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
- 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$

- 칼만 필터를 사용하여 열차의 위치와 “속도를 추정”하자!

상태 변수

$$\begin{bmatrix} \text{측정 변수} \end{bmatrix}_k = \begin{bmatrix} \quad \quad \end{bmatrix} \begin{bmatrix} \text{상태 변수} \end{bmatrix}_k$$

# 상태 모델 & 출력 행렬 H

- 4초 동안 0.1초 간격으로 열차의 “위치 측정”

측정 변수

- 열차의 명목 속도 = 80 m/s
- 시스템 위치 잡음  $\sim 0$
- 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
- 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$

- 칼만 필터를 사용하여 열차의 위치와 “속도를 추정”하자!

상태 변수

$$\begin{bmatrix} \text{측정 위치} \end{bmatrix}_k = \begin{bmatrix} \quad \quad \end{bmatrix} \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_k$$

# 상태 모델 & 출력 행렬 H

- 4초 동안 0.1초 간격으로 열차의 “위치 측정”
  - 열차의 명목 속도 = 80 m/s
  - 시스템 위치 잡음  $\sim 0$
  - 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
  - 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$
- 칼만 필터를 사용하여 열차의 위치와 “속도를 추정”하자!

측정 변수

상태 변수

$$\begin{bmatrix} \text{측정 위치} \end{bmatrix}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_k$$

$H$

# 상태 모델 & 출력 행렬 H

- 4초 동안 0.1초 간격으로 열차의 “위치 측정”

측정 변수

- 열차의 명목 속도 = 80 m/s
- 시스템 위치 잡음  $\sim 0$
- 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
- 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$

- 칼만 필터를 사용하여 열차의 위치와 “속도를 추정”하자!

상태 변수

$$\begin{bmatrix} \text{측정 위치} \end{bmatrix}_k = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_H \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_k + v_k$$

# 선형 상태 & 시스템 모델

## 선형 상태 모델

- $x_{k+1} = Ax_k + w_k$   
 $w_k \sim Gauss(0, \sigma_w^2)$
- $z_k = Hx_k + v_k$   
 $v_k \sim Gauss(0, \sigma_v^2)$

## 시스템 모델

- $A$  : 시간에 따른 상태 변환
- $H$  : 상태를 측정값으로 변환
- $Q$  : 시스템 잡음 공분산
- $R$  : 측정 잡음 공분산

# 선형 상태 & 시스템 모델

## 선형 상태 모델

$$\bullet \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_{k+1} = \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_k + \begin{bmatrix} 0 \\ w_k \end{bmatrix}$$

$$\bullet \begin{bmatrix} \text{측정 위치} \end{bmatrix}_k = \begin{bmatrix} H \end{bmatrix} \begin{bmatrix} \text{위치} \\ \text{속도} \end{bmatrix}_k + v_k$$



# 선형 상태 & 시스템 모델

## 시스템 모델

- $A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$

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

- $Q = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$

- $R = \begin{bmatrix} 10 \end{bmatrix}$

# 칼만 필터: 위치로 속도 추정하기 예제

코드로 보아요!

# 코드: 위치 측정 하기

- 4초 동안 0.1초 간격으로 열차의 “위치 측정”

- 열차의 명목 속도 = 80 m/s
- 시스템 위치 잡음  $\sim 0$
- 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
- 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$

- $\text{속도}_k = 80 \text{ m/s} + w_k$

$$\text{위치}_{k+1} = \text{위치}_k + \text{속도}_k \times \Delta t$$

$$\text{측정 위치}_{k+1} = \text{위치}_{k+1} + v_k$$

```
def get_pos_vel(itime, pos_true):  
    """Return Measured Position, True Velocity and True Position."""  
    w = np.random.normal(0, np.sqrt(10))    # w: system noise.  
    v = np.random.normal(0, np.sqrt(10))    # v: measurement noise.  
  
    vel_true = 80 + w                        # nominal velocity = 80 [m/s].  
    if itime == 0:  
        pos_true = pos_true  
    else:  
        pos_true = pos_true + vel_true * dt  
    z_pos_meas = pos_true + v                # z_pos_meas: measured position (observable)  
    return z_pos_meas, vel_true, pos_true
```

# 코드: 시스템 모델 정의 & 추정량 초기화

- $A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$

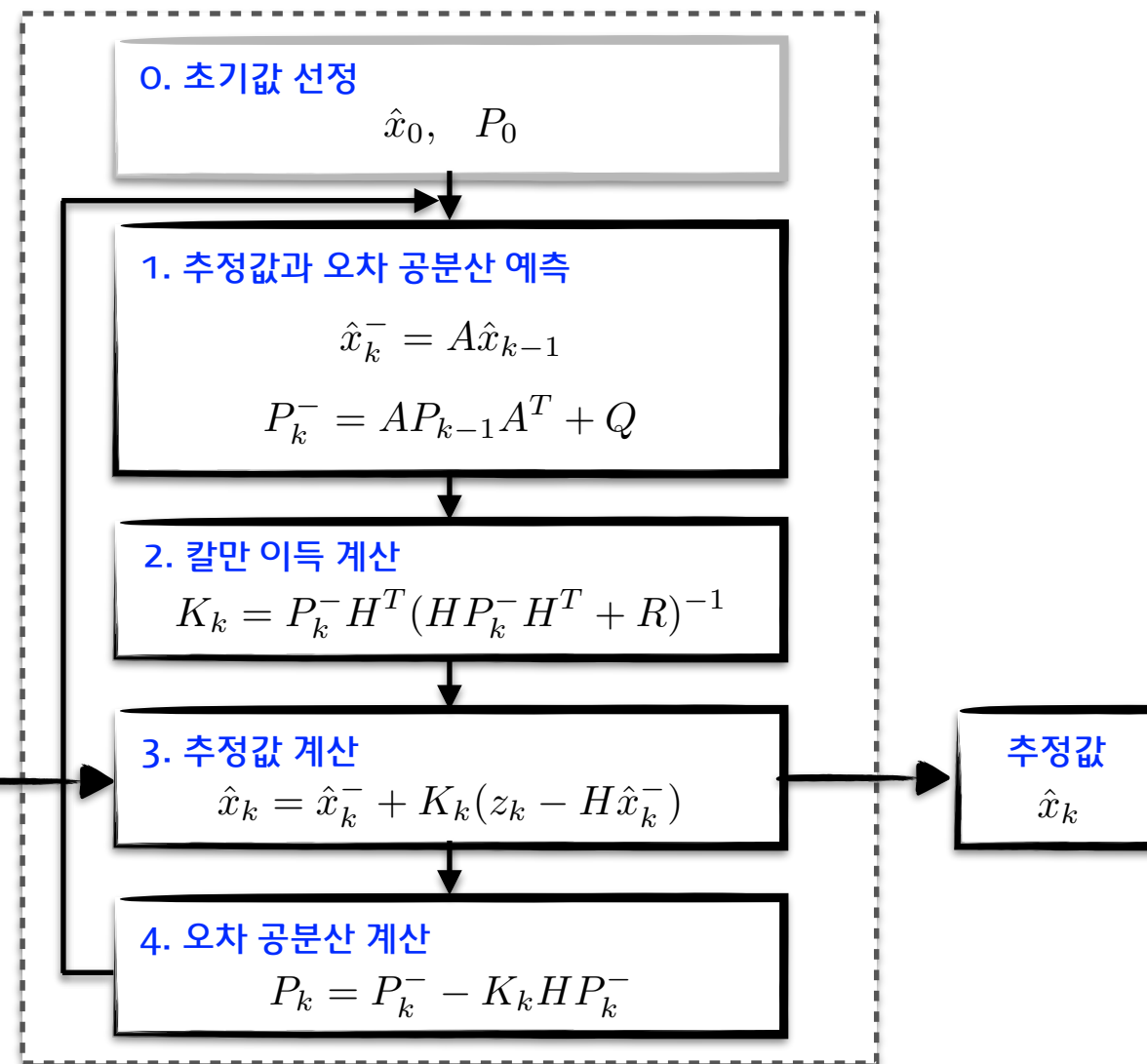
- $Q = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$

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

- $R = \begin{bmatrix} 10 \end{bmatrix}$

```
# Initialization for system model.  
# Define matrix with rank 2 for matrix operation.  
A = np.array([[1, dt],  
              [0, 1]])  
H = np.array([[1, 0]])  
Q = np.array([[1, 0],  
              [0, 3]])  
R = np.array([[10]])  
  
# Initialization for estimation.  
x_0 = np.array([0, 20]) # position and velocity  
P_0 = 5 * np.eye(2)
```

# 코드: 칼만 필터 알고리즘



```
def kalman_filter(z_meas, x_esti, P):  
    """Kalman Filter Algorithm."""  
    # (1) Prediction.  
    x_pred = A @ x_esti  
    P_pred = A @ P @ A.T + Q  
  
    # (2) Kalman Gain.  
    K = P_pred @ H.T @ inv(H @ P_pred @ H.T + R)  
  
    # (3) Estimation.  
    x_esti = x_pred + K @ (z_meas - H @ x_pred)  
  
    # (4) Error Covariance.  
    P = P_pred - K @ H @ P_pred  
  
    return x_esti, P
```

# 코드: 위치, 속도 추정

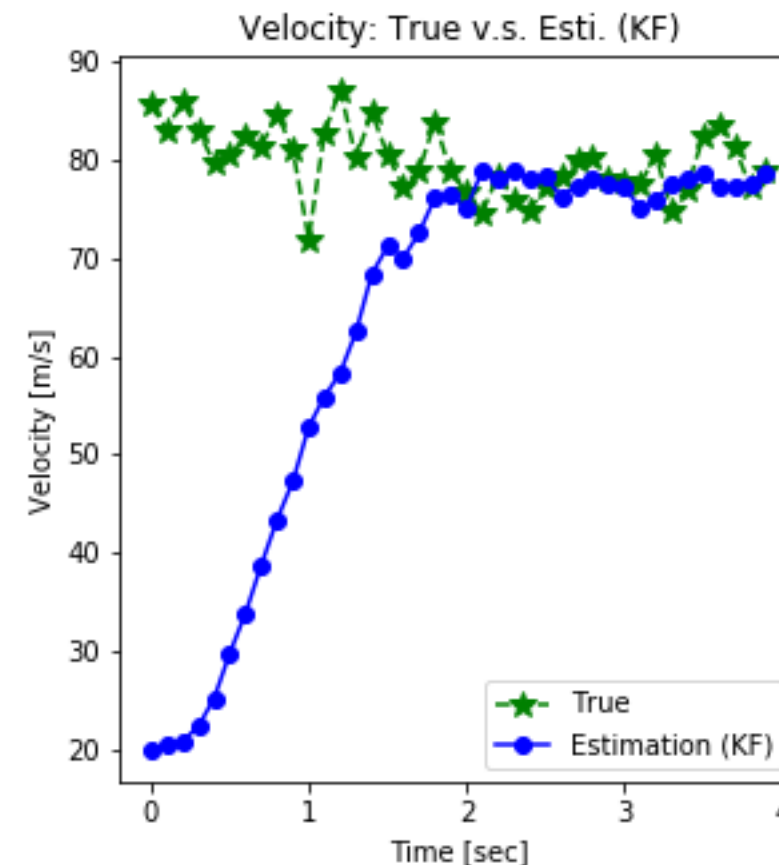
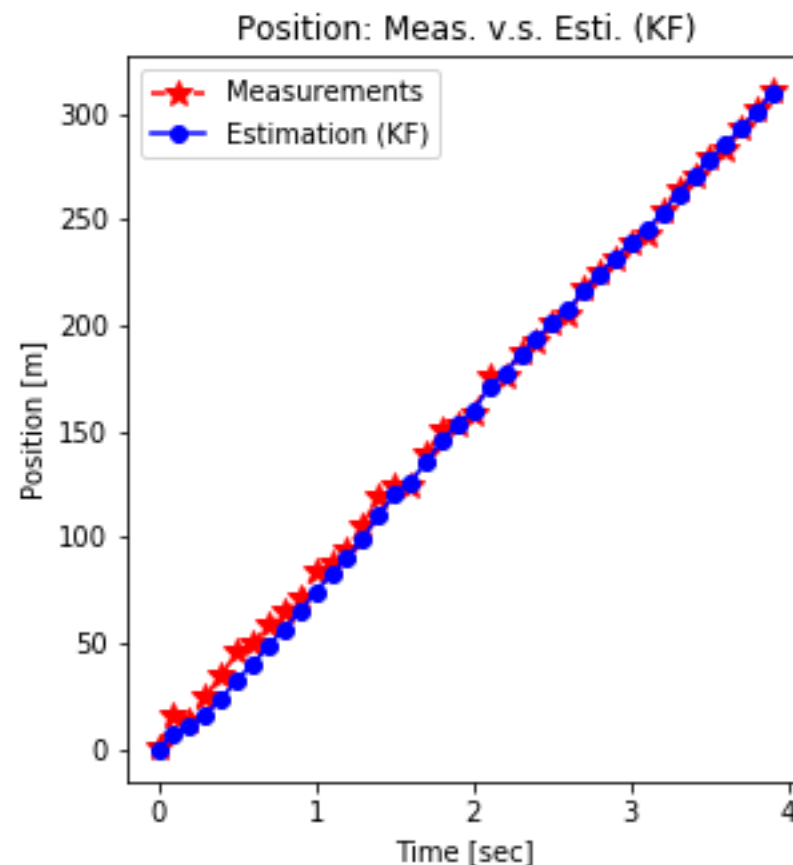
- 4초 동안 0.1초 간격으로 열차의 “위치 측정”
  - n\_samples = 40

```
pos_true = 0
x_esti, P = None, None
for i in range(n_samples):
    z_meas, vel_true, pos_true = get_pos_vel(i, pos_true)
    if i == 0:
        x_esti, P = x_0, P_0
    else:
        x_esti, P = kalman_filter(z_meas, x_esti, P)

    pos_meas_save[i] = z_meas
    vel_true_save[i] = vel_true
    pos_esti_save[i] = x_esti[0]
    vel_esti_save[i] = x_esti[1]
```

# 코드: 위치로 속도 추정하기 결과

- 4초 동안 0.1초 간격으로 열차의 “위치 측정”
  - 열차의 명목 속도 = 80 m/s
  - 시스템 위치 잡음  $\sim 0$
  - 시스템 속도 잡음  $w_k \sim \text{Gauss}(0, 10)$
  - 측정 위치 잡음  $v_k \sim \text{Gauss}(0, 10)$
- 칼만 필터를 사용하여 열차의 위치와 “속도를 추정”하자!

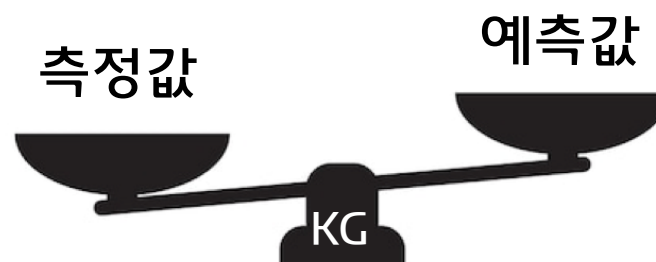


[https://github.com/tbmoon/kalman\\_filter/blob/master/Ch09.Pos2VelKF/pos2vel\\_kf.ipynb](https://github.com/tbmoon/kalman_filter/blob/master/Ch09.Pos2VelKF/pos2vel_kf.ipynb)

# 칼만 필터 파라미터

- 초기 추정량 ( $x_0$ )이 커지면 어떻게 될까?
- 시스템 잡음 공분산 ( $Q$ )이 커지면 어떻게 될까?
- 측정 잡음 공분산 ( $R$ )이 커지면 어떻게 될까?

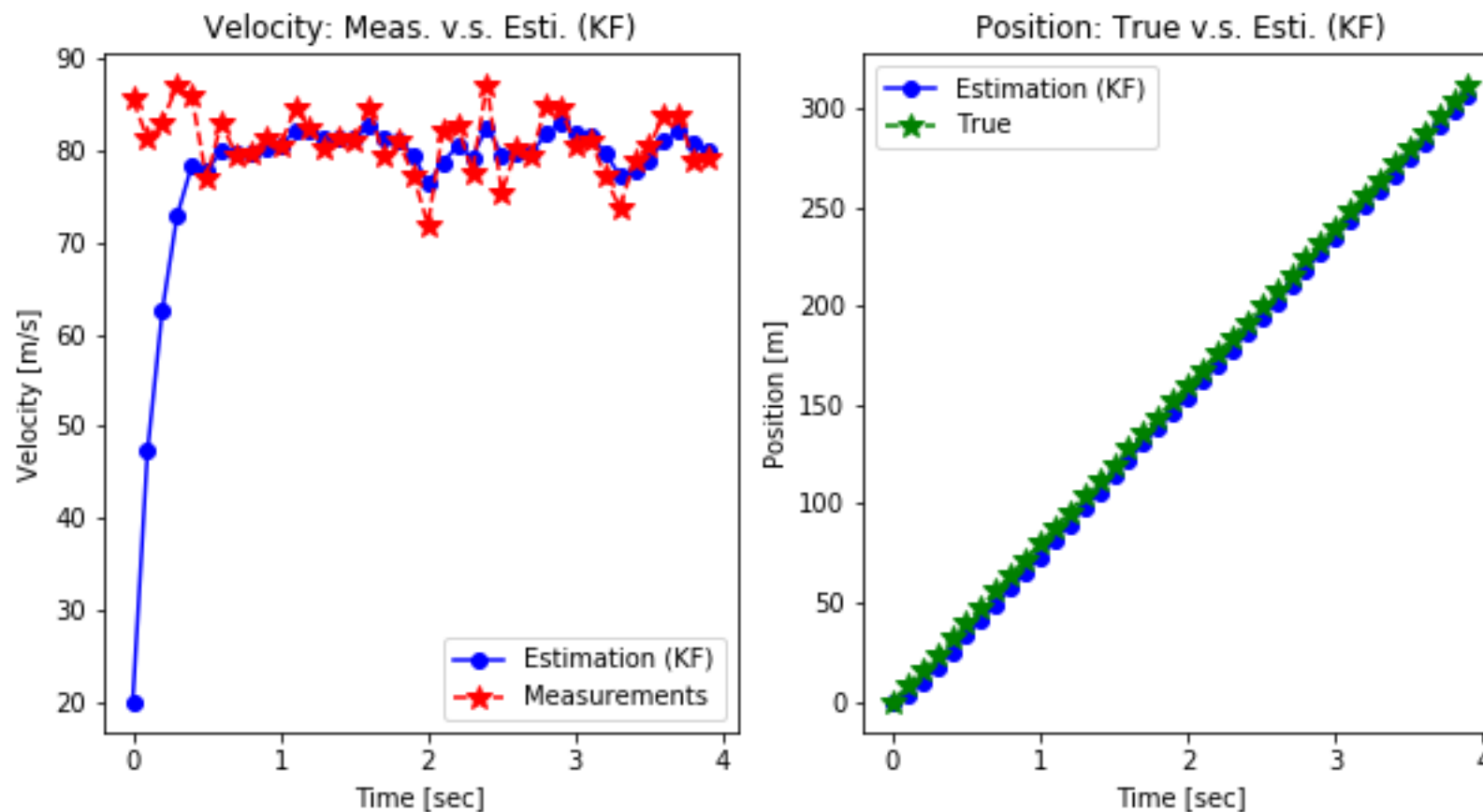
오차 공분산 ( $P$ )를 출력하면서 해보세요!





# 칼만 필터: 속도로 위치 추정하기 예제

- 4초 동안 0.1초 간격으로 열차의 “속도 측정”
  - 열차의 명목 속도 = 80 m/s
  - 시스템 위치 잡음  $\sim 0$
  - 시스템 속도 잡음  $\sim 0$
  - 측정 속도 잡음  $v_k \sim \text{Gauss}(0, 10)$
- 칼만 필터를 사용하여 열차의 속도와 “위치를 추정”하자!



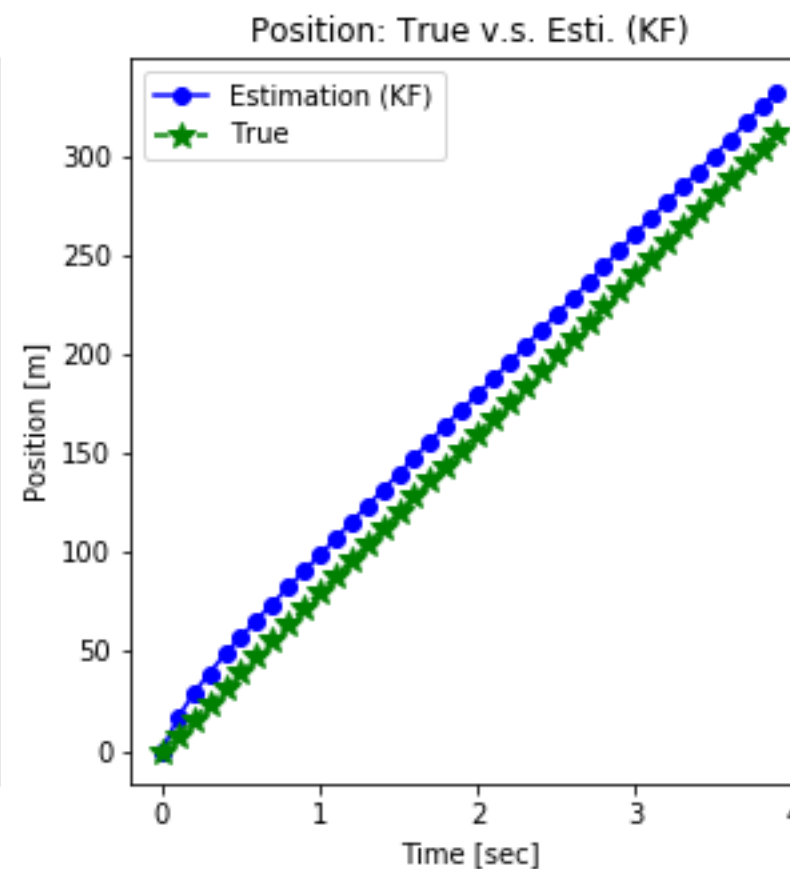
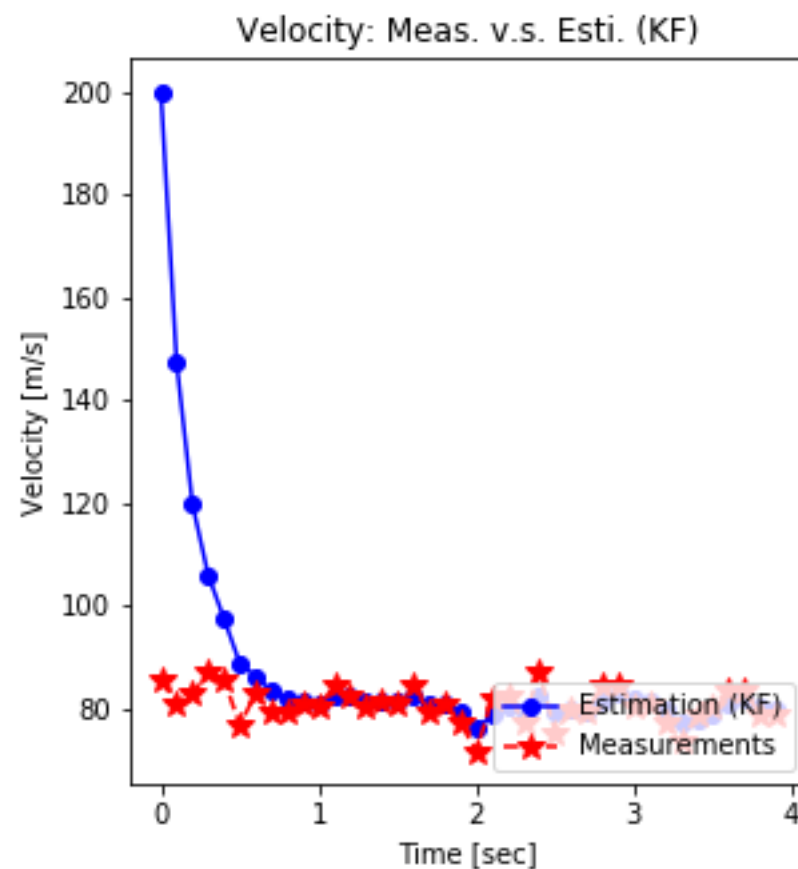
[https://github.com/tbmoon/kalman\\_filter/blob/master/Ch09.Pos2VelKF/vel2pos\\_kf.ipynb](https://github.com/tbmoon/kalman_filter/blob/master/Ch09.Pos2VelKF/vel2pos_kf.ipynb)

# 칼만 필터 파라미터

- 초기 속도 추정량 ( $= 200 \text{ m/s}$ )이 커지면 어떻게 될까?
- 칼만 필터의 단점이 무엇일까?

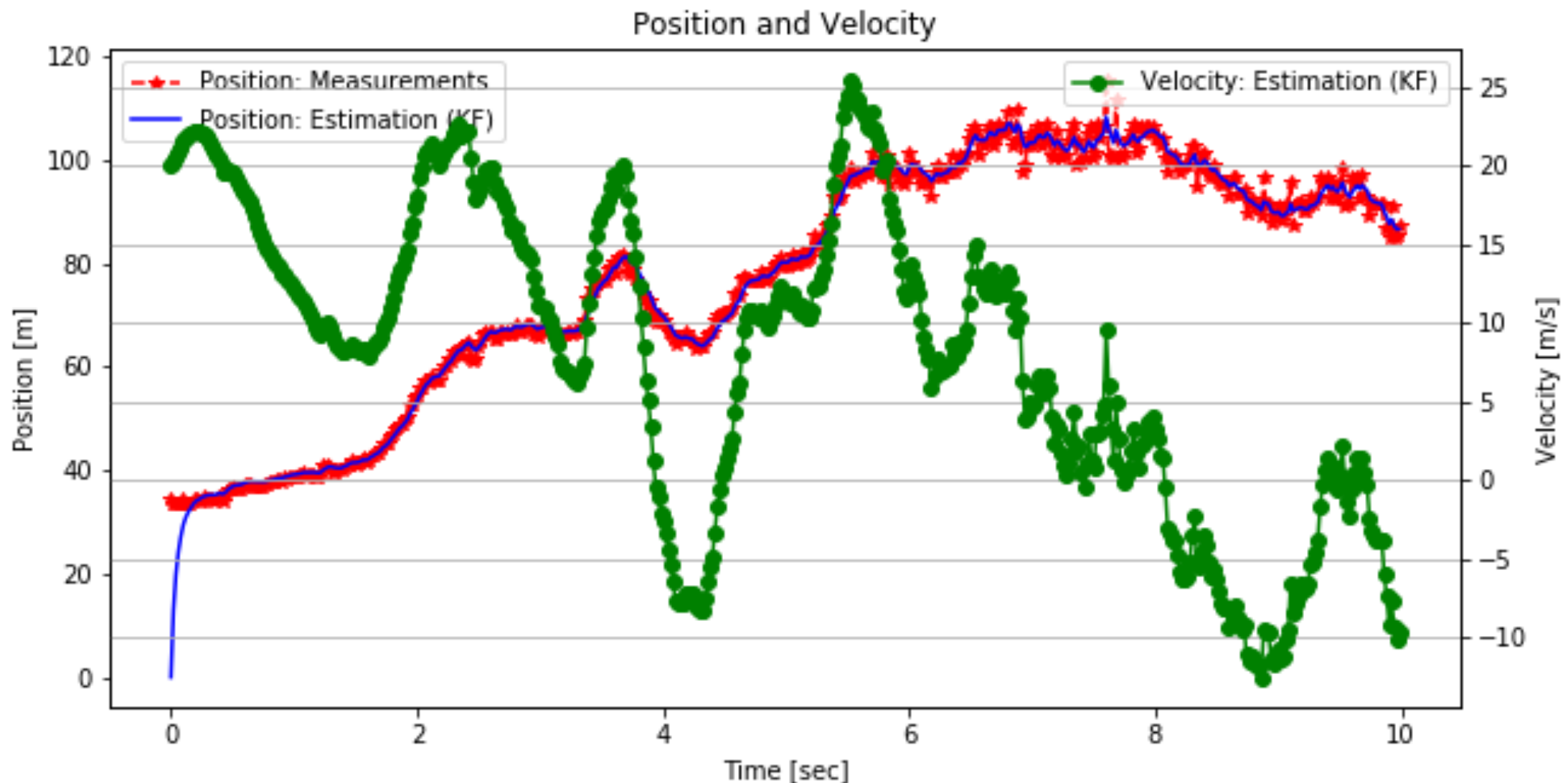
# 칼만 필터 파라미터

- 초기 속도 추정량 ( $= 200 \text{ m/s}$ )이 커지면 어떻게 될까?
- 칼만 필터의 단점이 무엇일까?



# 칼만 필터: 초음파 거리계 측정 예제

- 10초 동안 0.02초 간격으로 초음파 거리계 측정
  - 데이터: 측정 거리 + 잡음
- 칼만 필터를 사용하여 거리와 속도를 추정하자!



[https://github.com/tbmoon/kalman\\_filter/blob/master/Ch09.Pos2VelKF/sonar\\_pos2vel\\_kf.ipynb](https://github.com/tbmoon/kalman_filter/blob/master/Ch09.Pos2VelKF/sonar_pos2vel_kf.ipynb)

- 칼만 필터에서 시스템 모델 정의가 중요하다.
- 물리 모델을 통해 시간에 따라 변하는 물리량을 추정할 수 있다.
  - 전압 측정 예제: 전압 일정.
  - 위치 속도 추정 예제: 위치 변함, 속도 일정.
  - 초음파 거리계 예제: 위치 변함, 속도 변함.
- 잡음 제거에 유용하다.
- 측정하지 않은 물리량을 추정할 수 있다.
- 더 있을까?

- 칼만 필터는 어렵지 않아 (저자: 김성필 님)
- 파이썬으로 구현하는 칼만 필터

# 풀잎 스쿨 11 주간 일정

- ~~01 주차 1월 08일 - 재귀 필터 (퍼실어)~~
- ~~02 주차 1월 15일 - 칼만 필터 기초 & 초간단 칼만 필터 예제 (퍼실어)~~
- ~~03 주차 1월 22일 - 칼만 필터 기초 & 위치로 속도 추정하기 (퍼실어)~~
- 04 주차 1월 29일 - 영상 속의 물체 추정하기 (장창완, 최홍용)
- 05 주차 2월 05일 - 기울기 자세 측정하기 (한상윤, 이창주)
- 06 주차 2월 12일 - 기울기 자세 측정하기 (양덕수, 최홍용)
- 07 주차 2월 19일 - 확장 칼만 필터 (박광수)
- 08 주차 2월 26일 - 확장 칼만 필터 (이찬)
- 09 주차 3월 04일 - 무향 칼만 필터 (이찬)
- 10 주차 3월 11일 - 무향 칼만 필터 (이운전)
- 11 주차 3월 18일 - 파티클 필터 혹은 정리/짬 파티 (한상윤, 이주희)

감사합니다