칼만 필터 & 초간단 예제

문태봉

2020.01.15 (수)

칼만 필터 알고리즘

0. 초기값 선정

$$\hat{x}_0, \quad P_0$$

1. 추정값과 오차 공분산 예측

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

$$P_k^- = AP_{k-1}A^T + Q$$

2. 칼만 이득 계산

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

3. 추정값 계산

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

4. 오차 공분산 계산

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

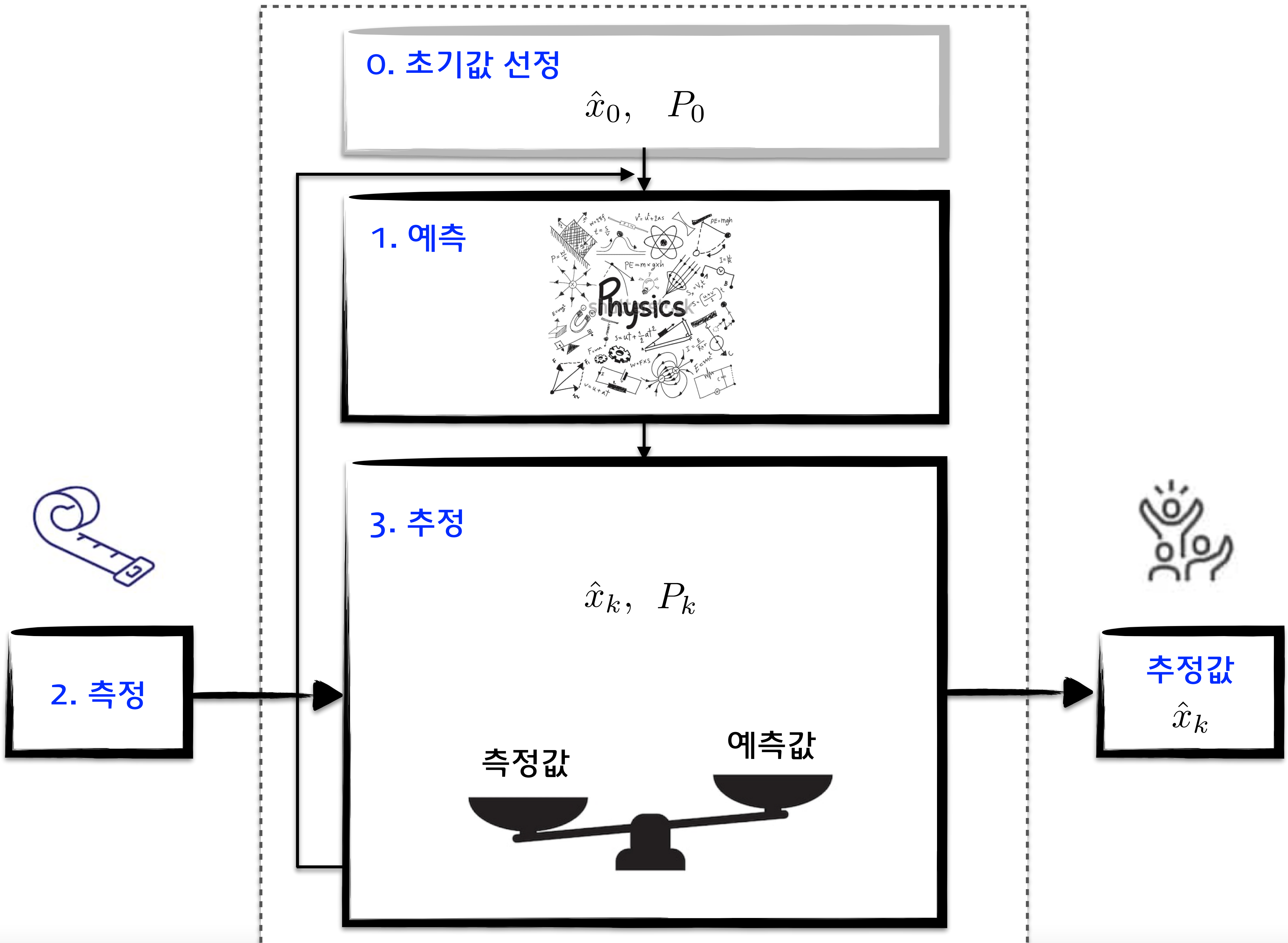
측정값

$$z_k$$

추정값

$$\hat{x}_k$$

칼만 필터 알고리즘

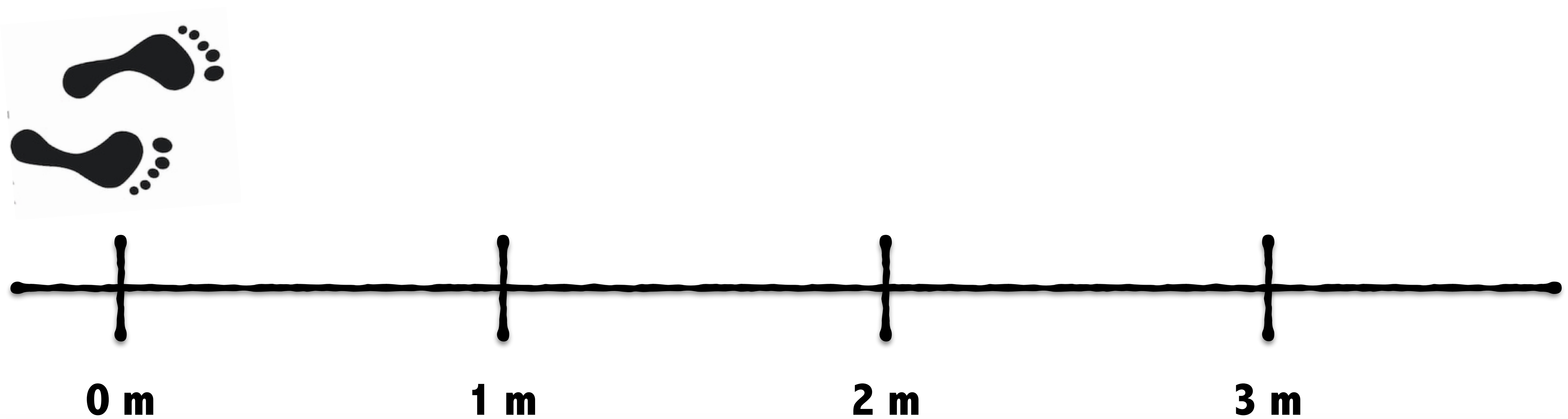

0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$

칼만 필터 흐름 예시

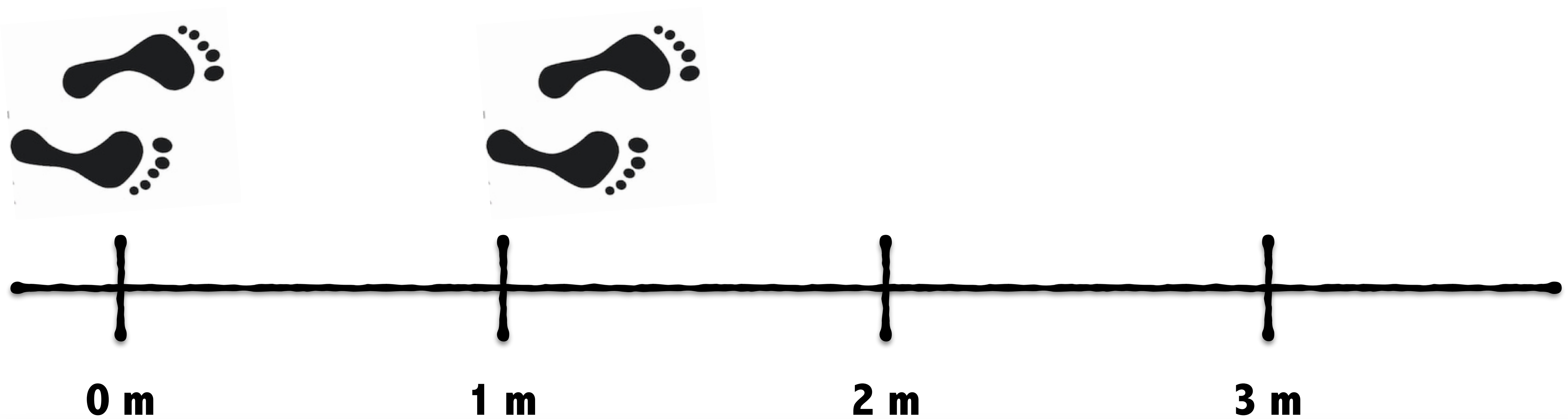
1차원에서 실제 이동 경로 추정을 해봐요!

트랙킹

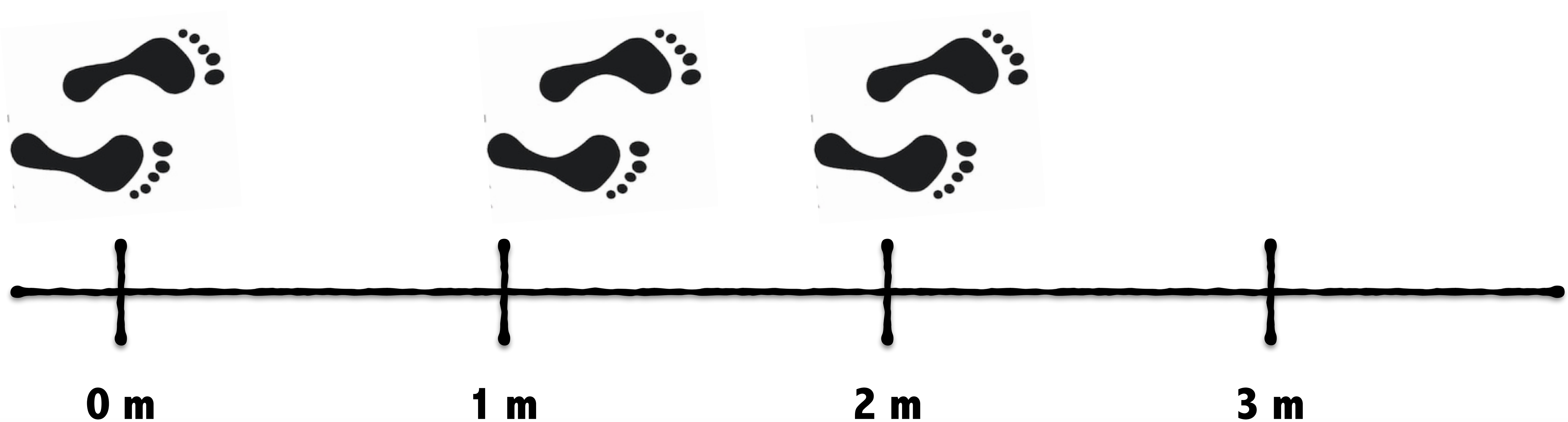
실제 이동 경로



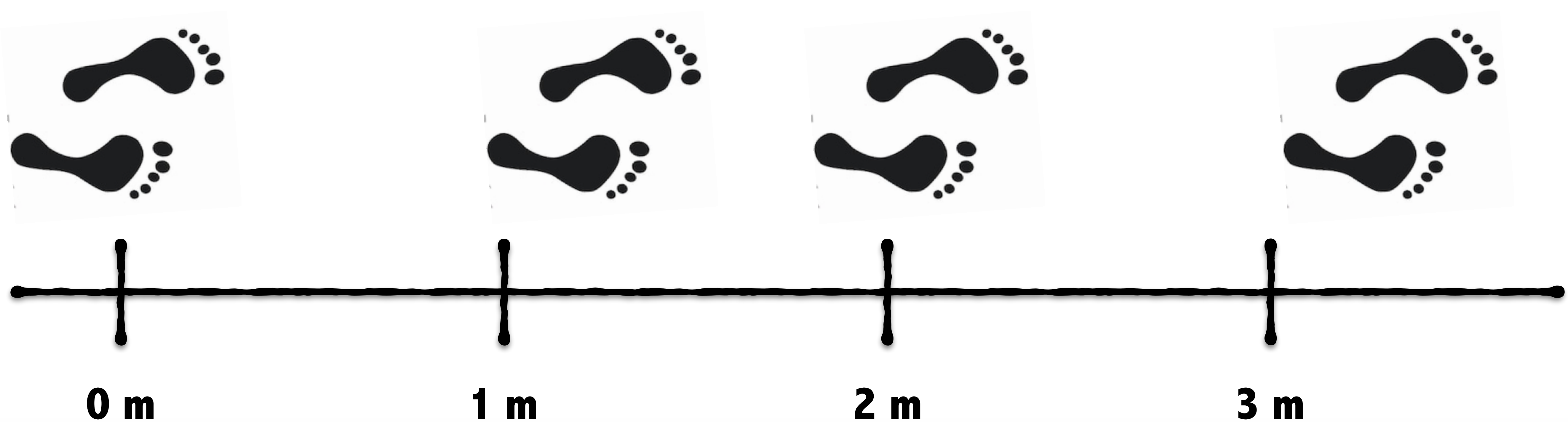
실제 이동 경로



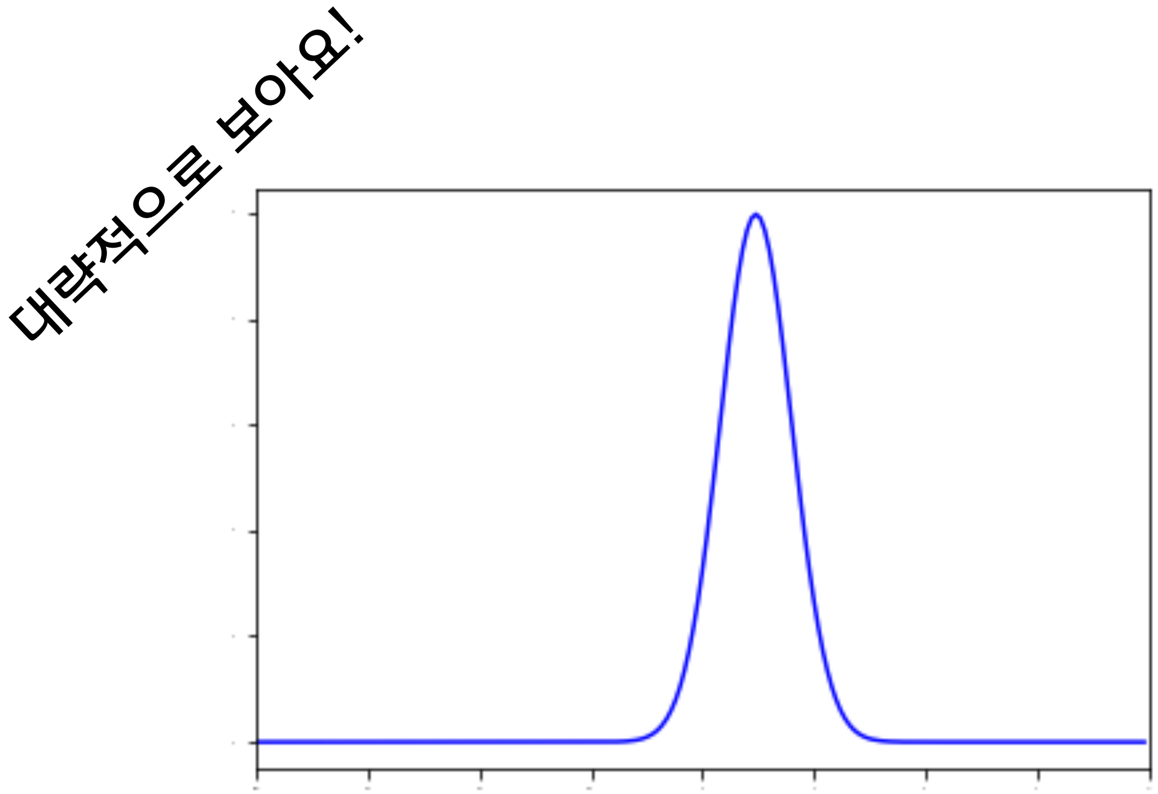
실제 이동 경로

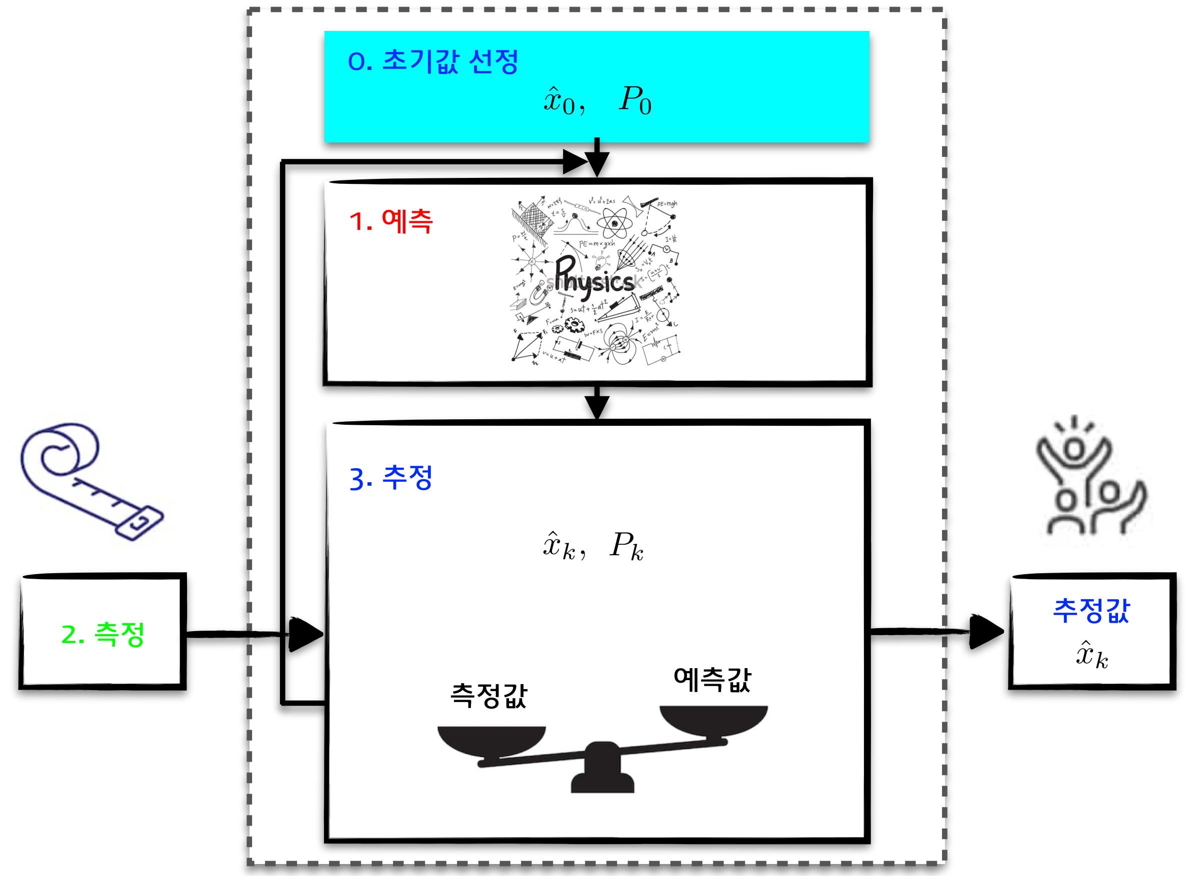


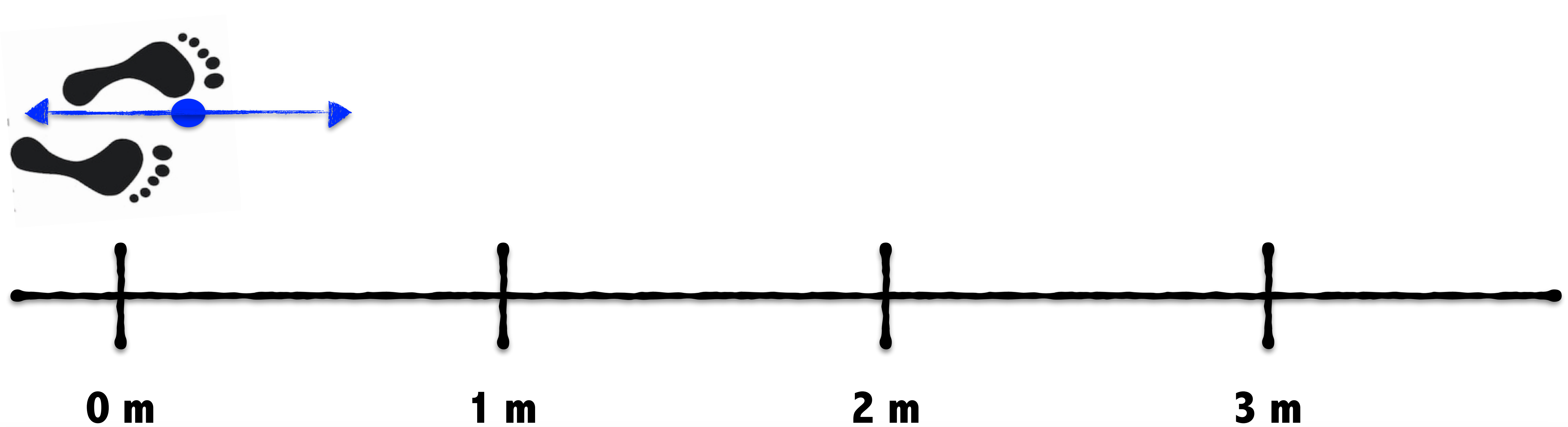
실제 이동 경로



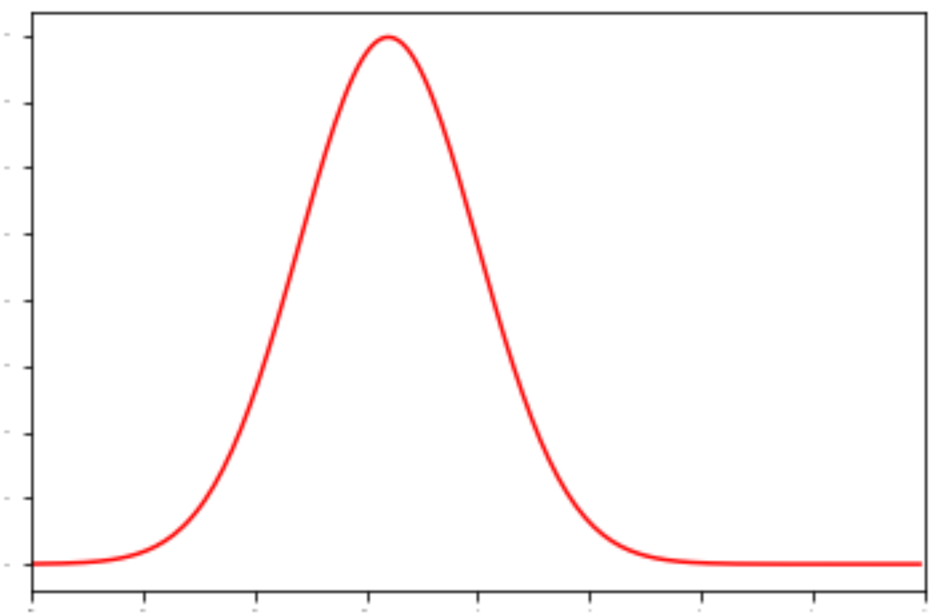
경로 추정 과정

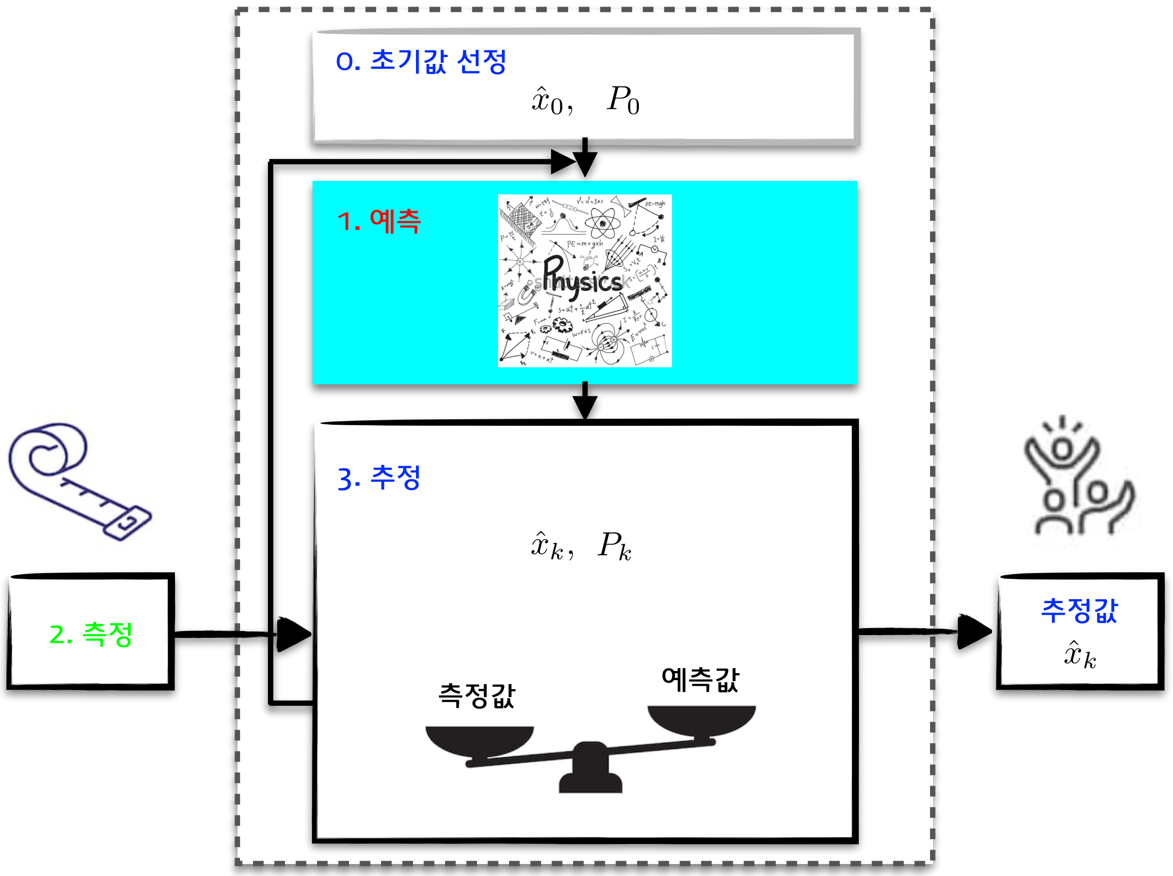

대략적으로 보아요!


0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$

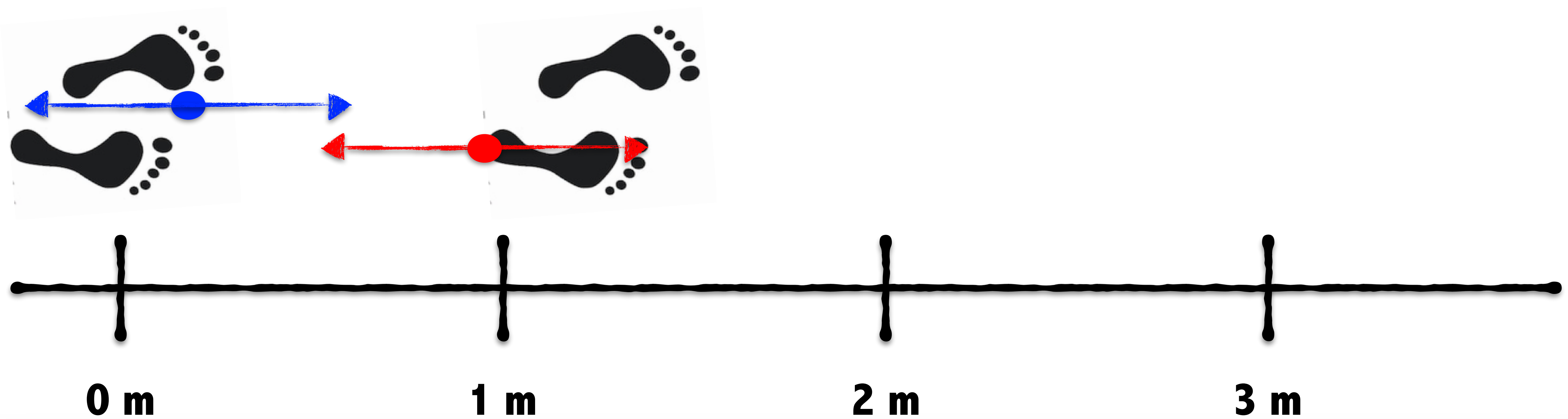

0 m
1 m
2 m
3 m

경로 추정 과정



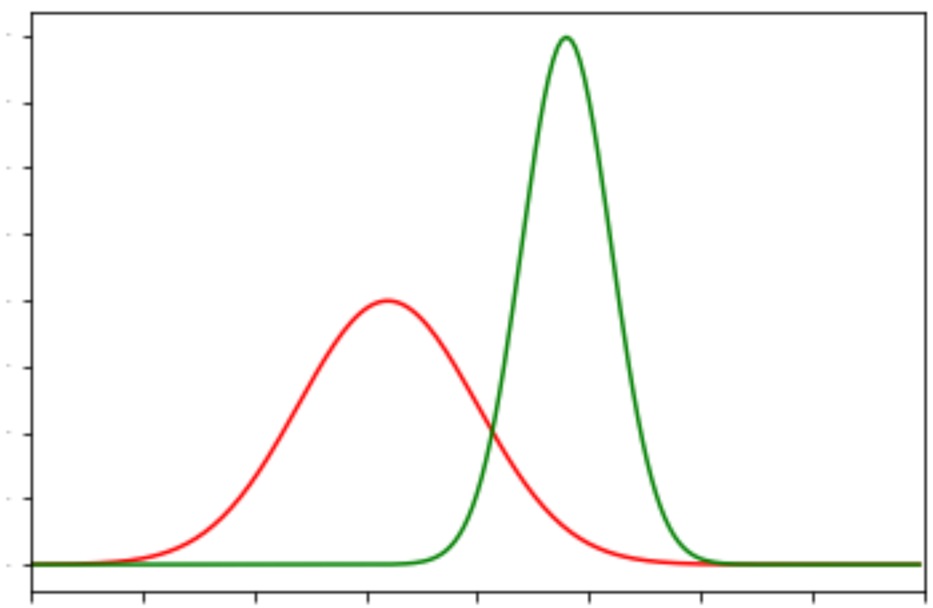


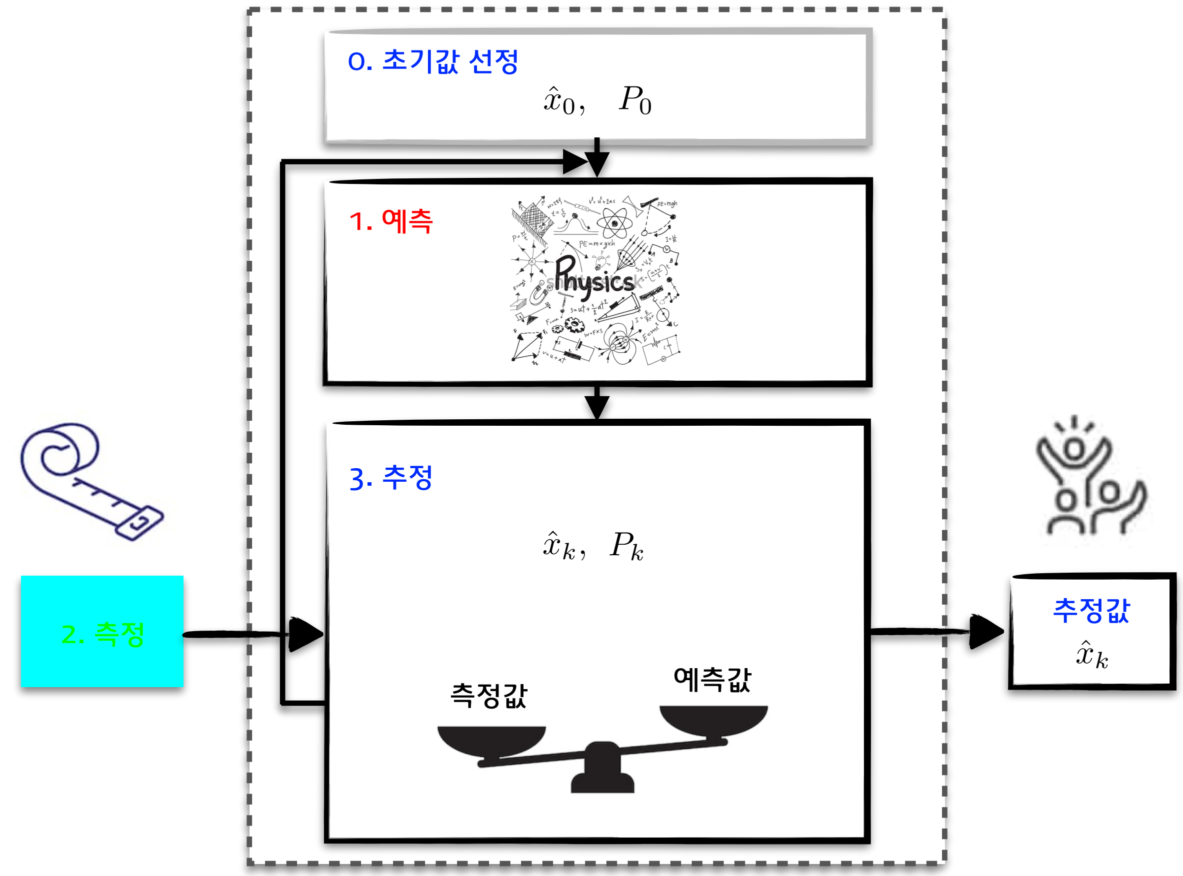
0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$



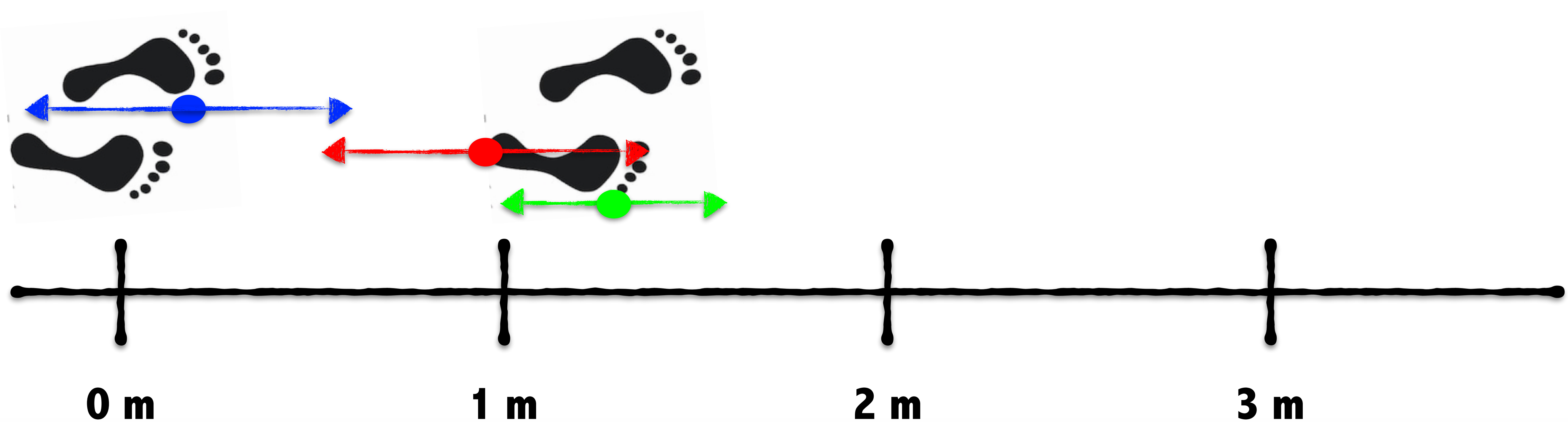
0 m
1 m
2 m
3 m

경로 추정 과정



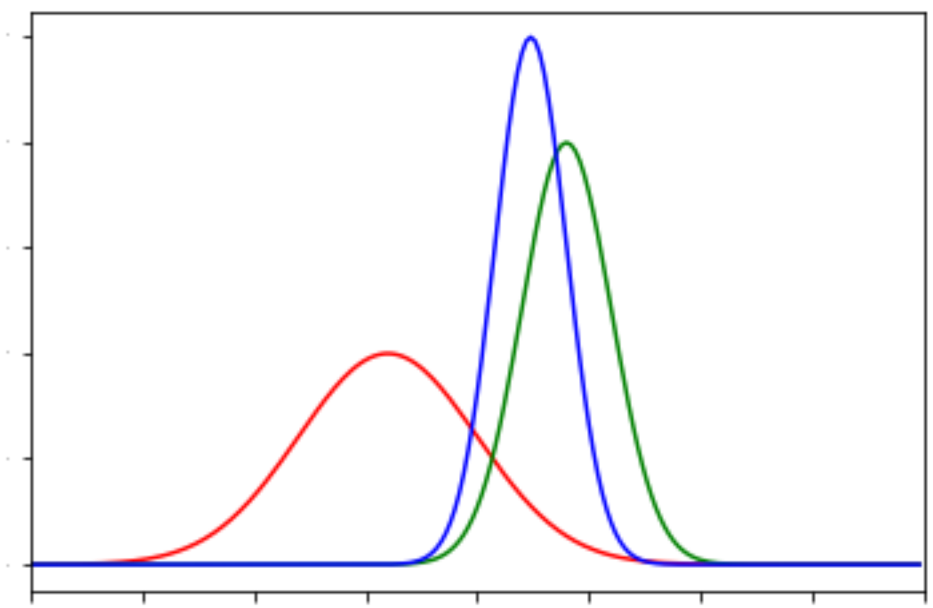


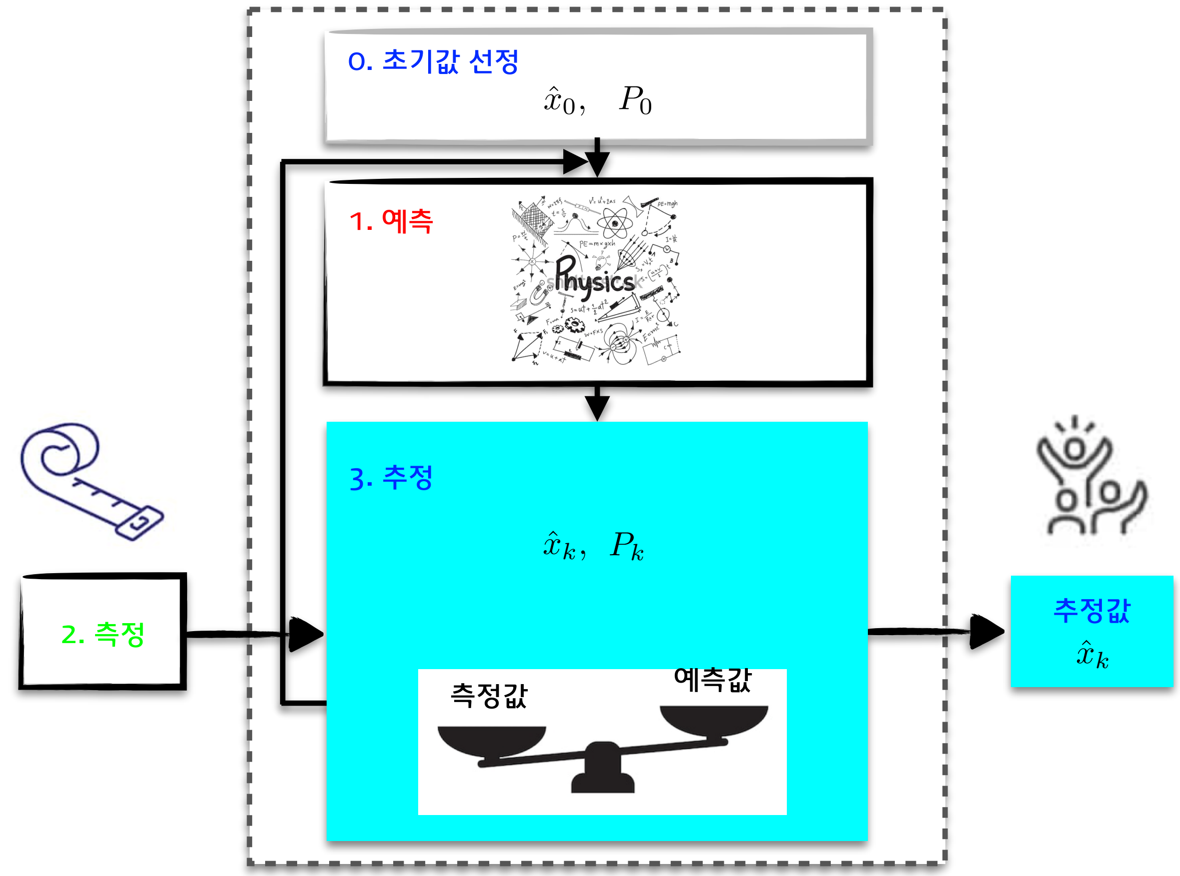
0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
2. 측정
추정값
$\hat{x}_k$



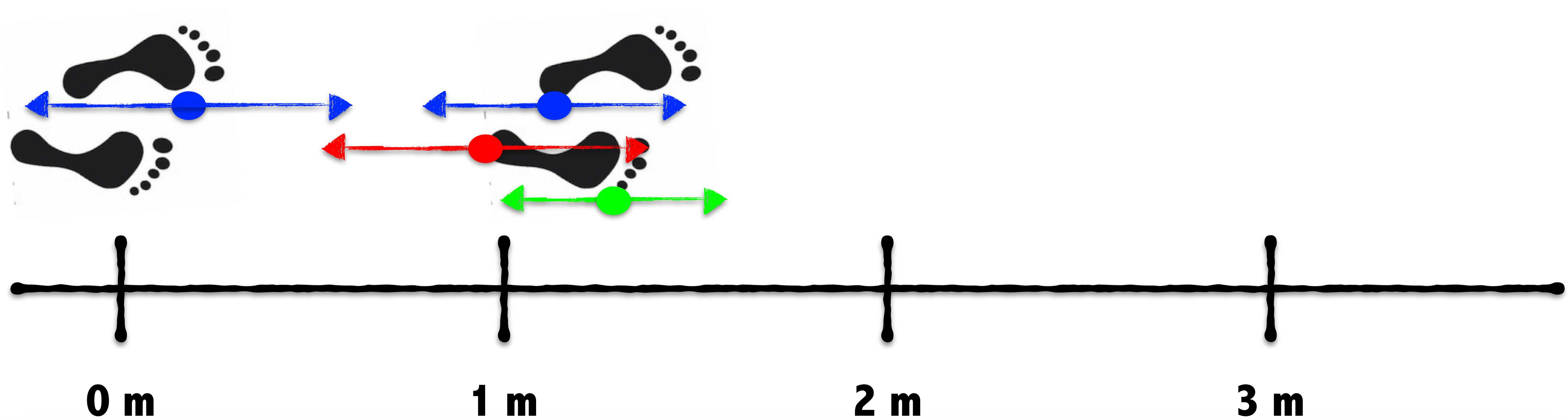
0 m
1 m
2 m
3 m

경로 추정 과정



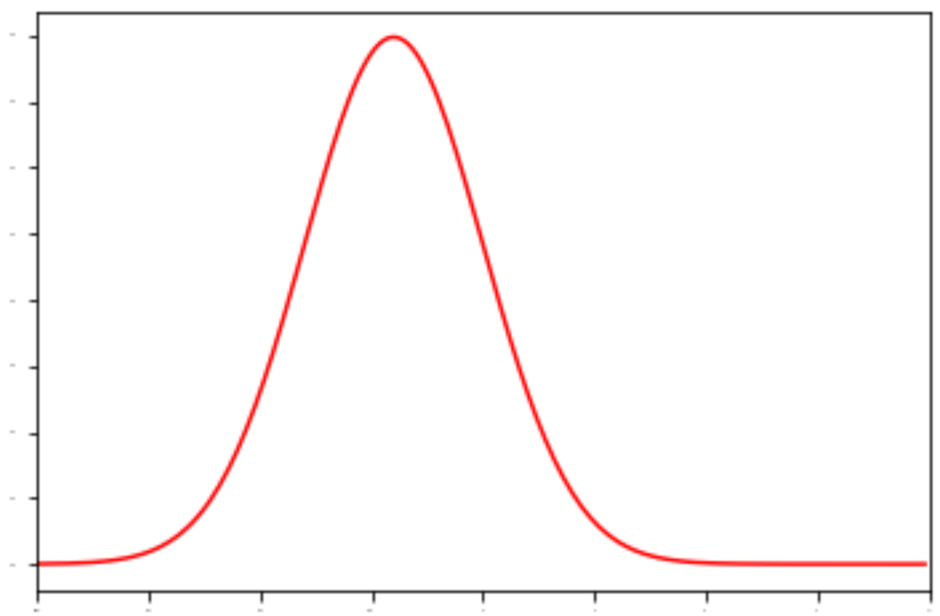


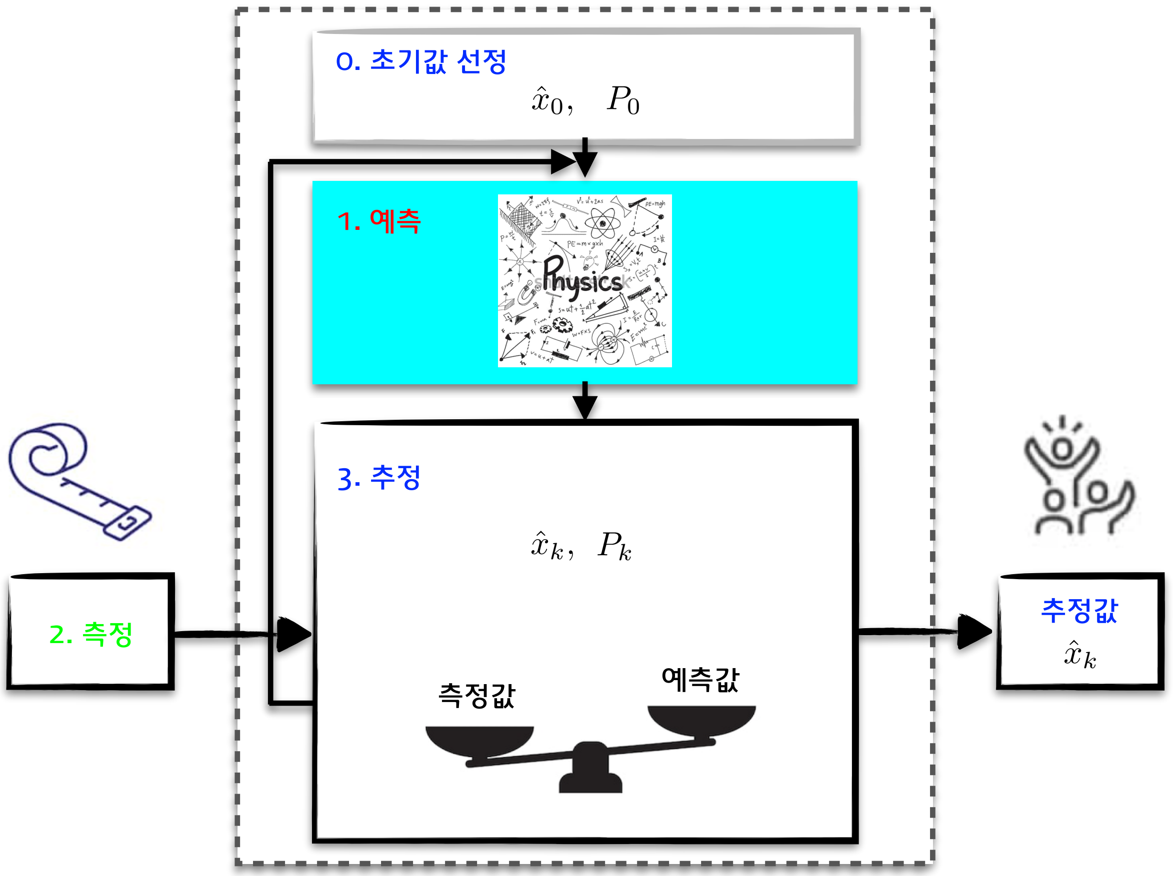
0. 초기값 선정
$\hat{x}_0$, P_0
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k$, P_k
측정값
예측값
추정값
$\hat{x}_k$



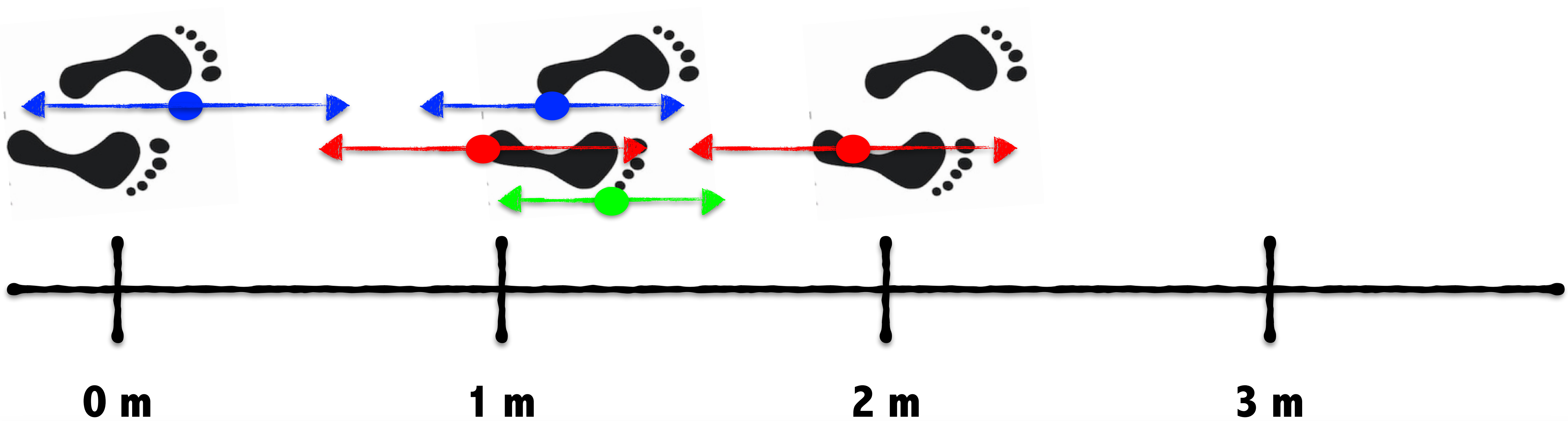
0 m
1 m
2 m
3 m

경로 추정 과정



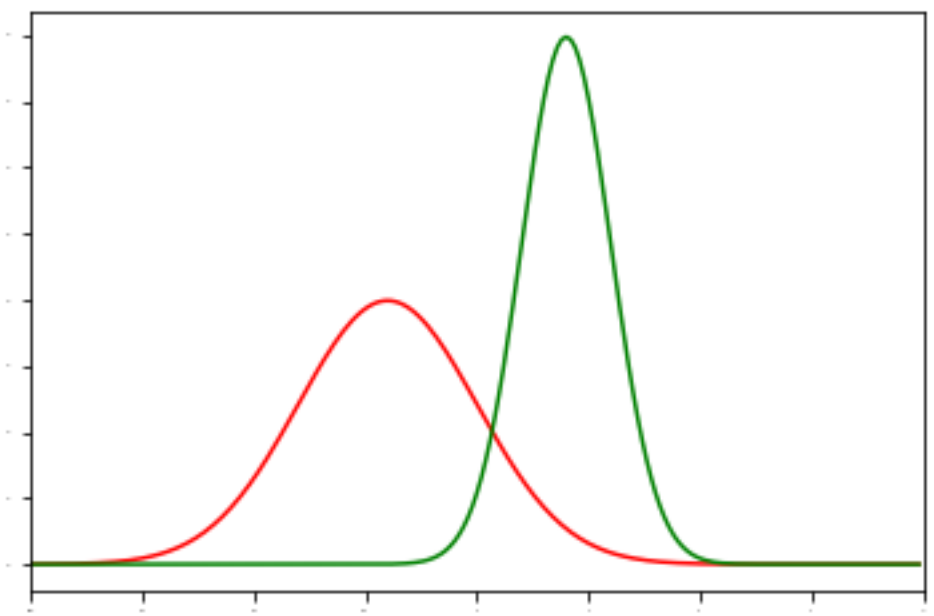


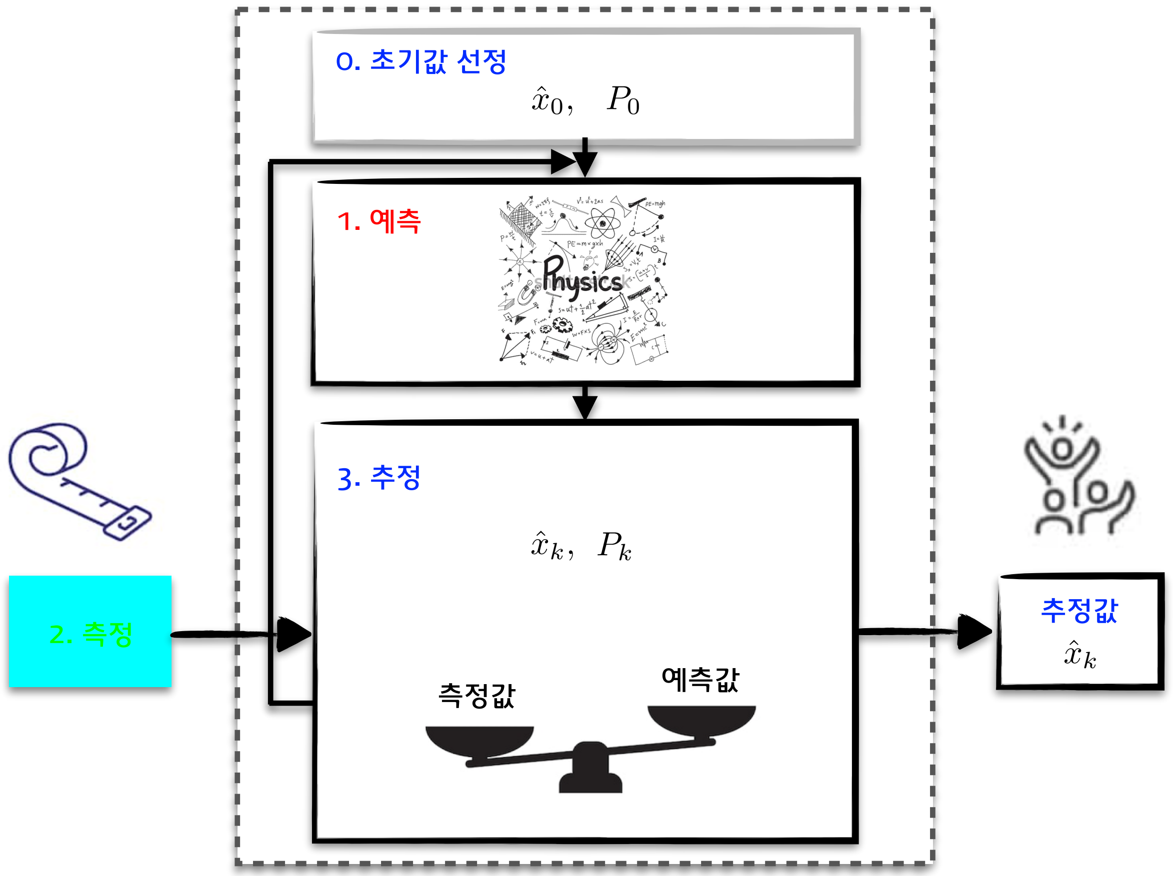
0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$



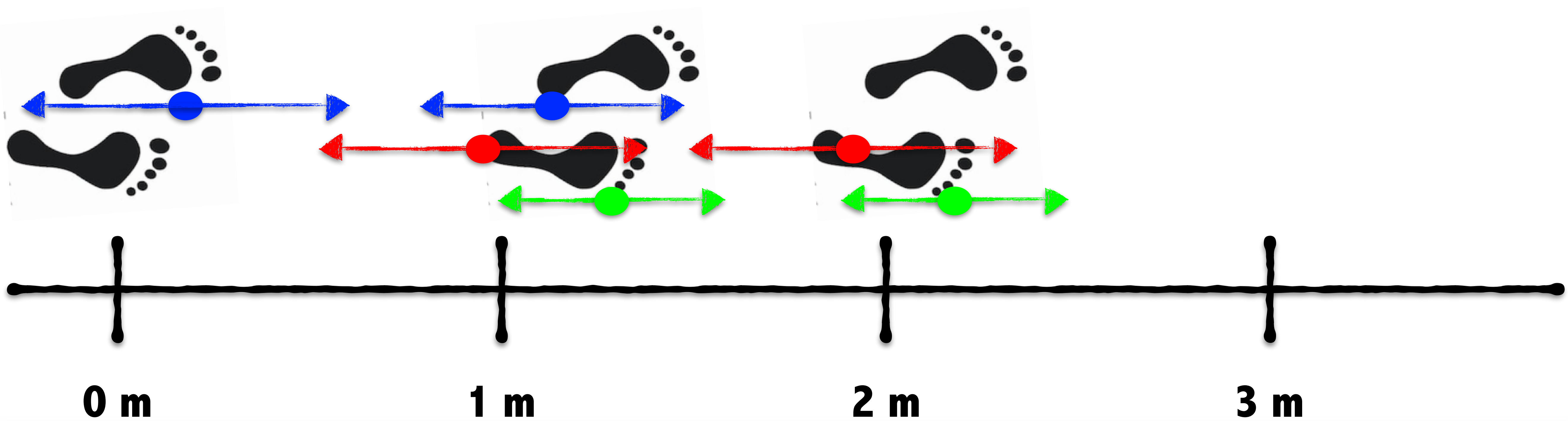
0 m
1 m
2 m
3 m

경로 추정 과정



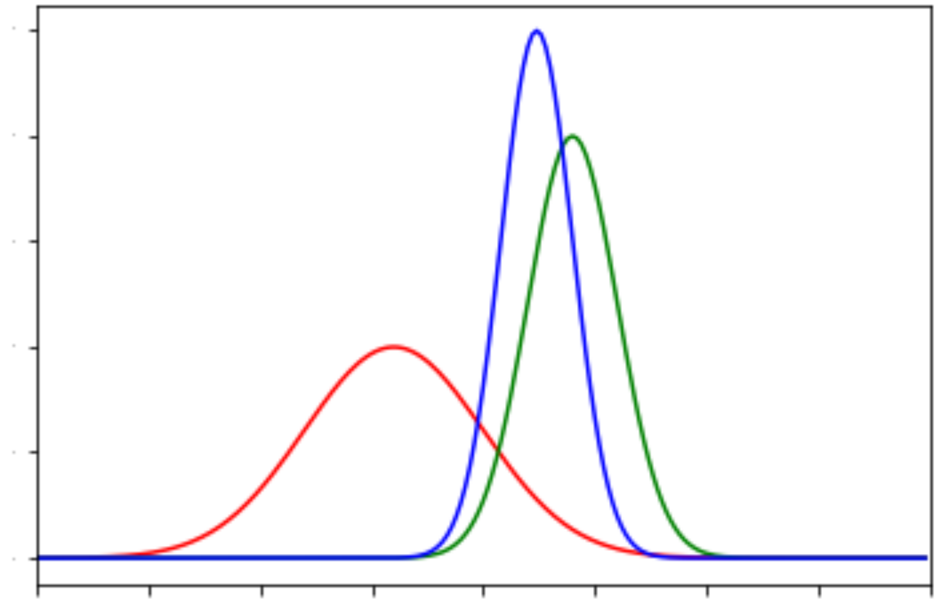


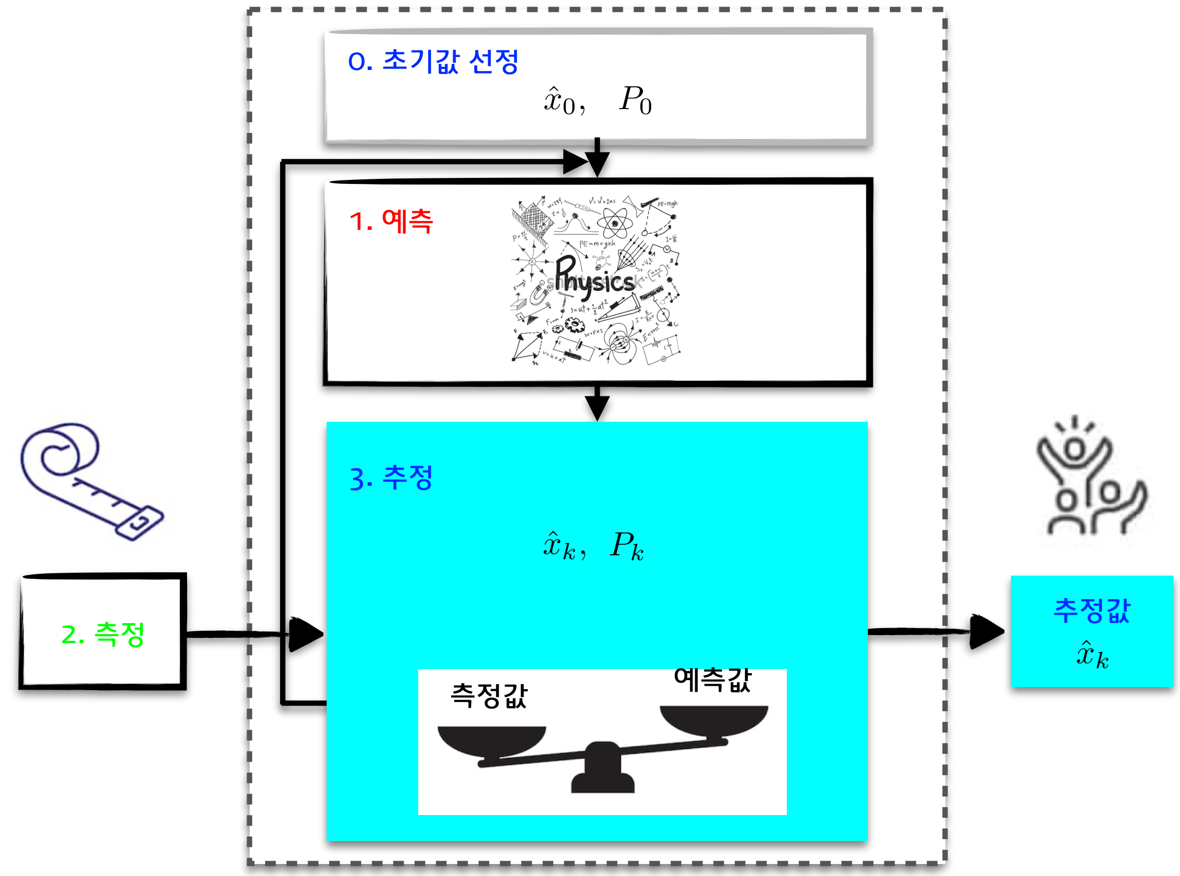
0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$

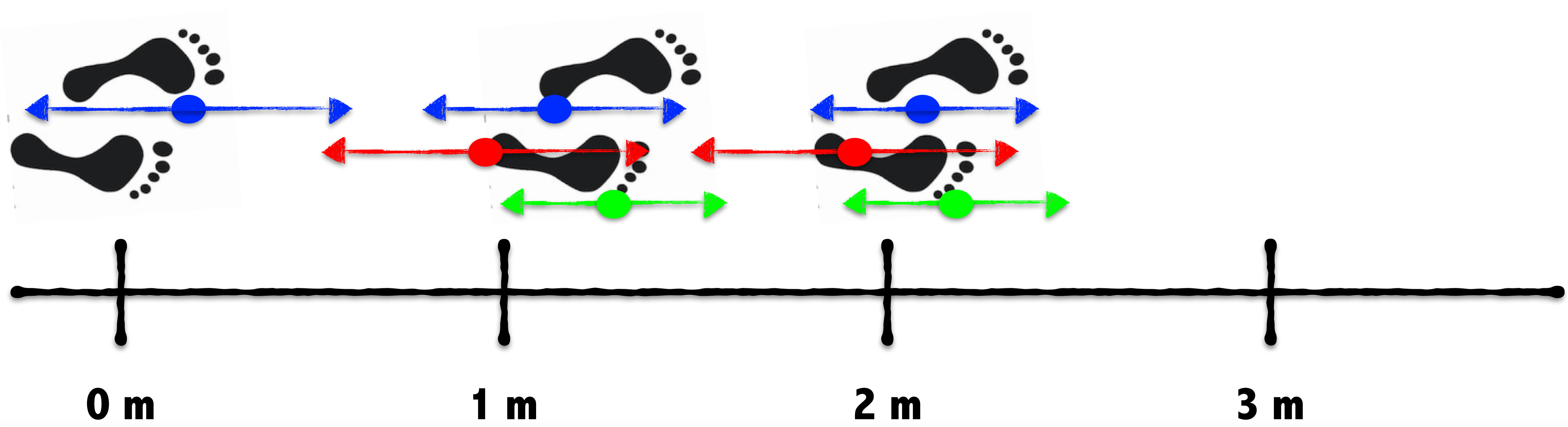


0 m
1 m
2 m
3 m

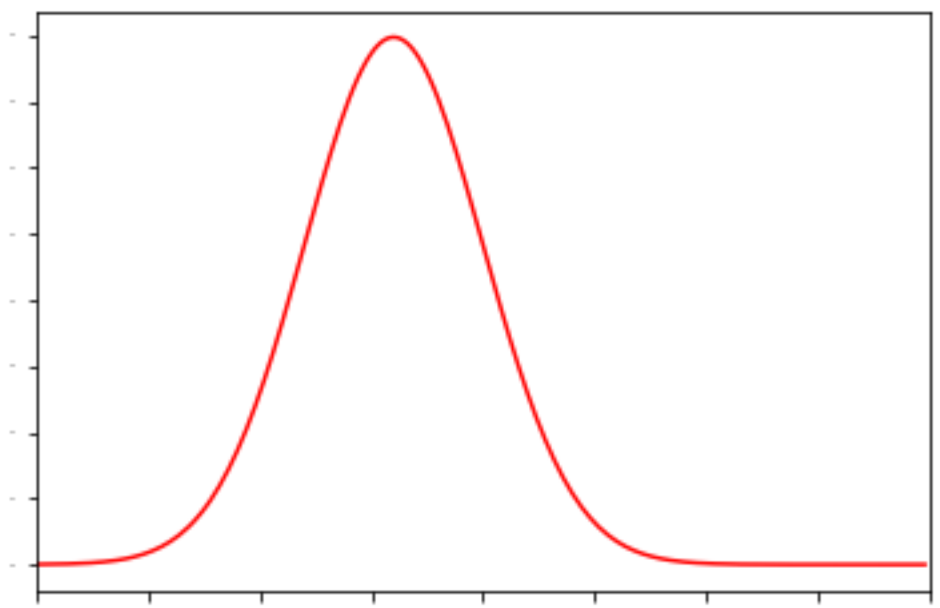
경로 추정 과정

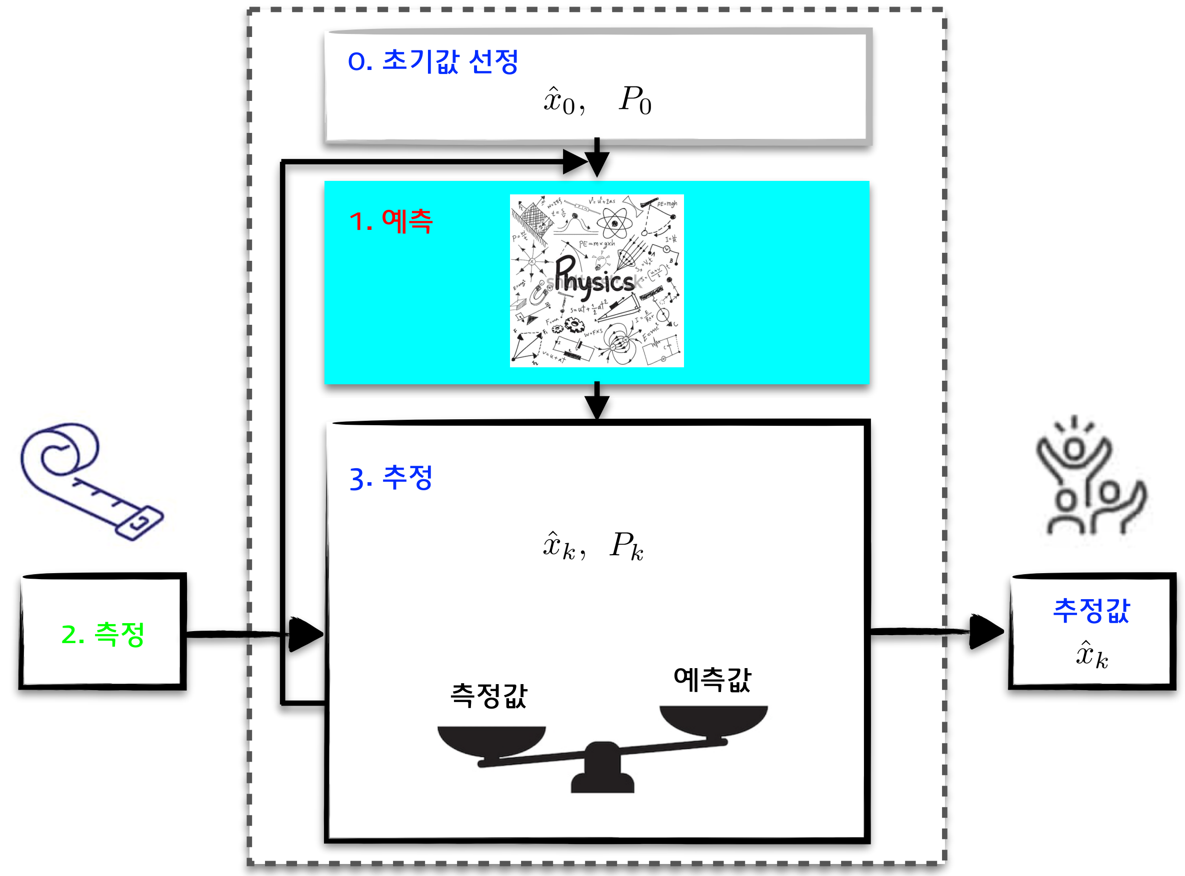


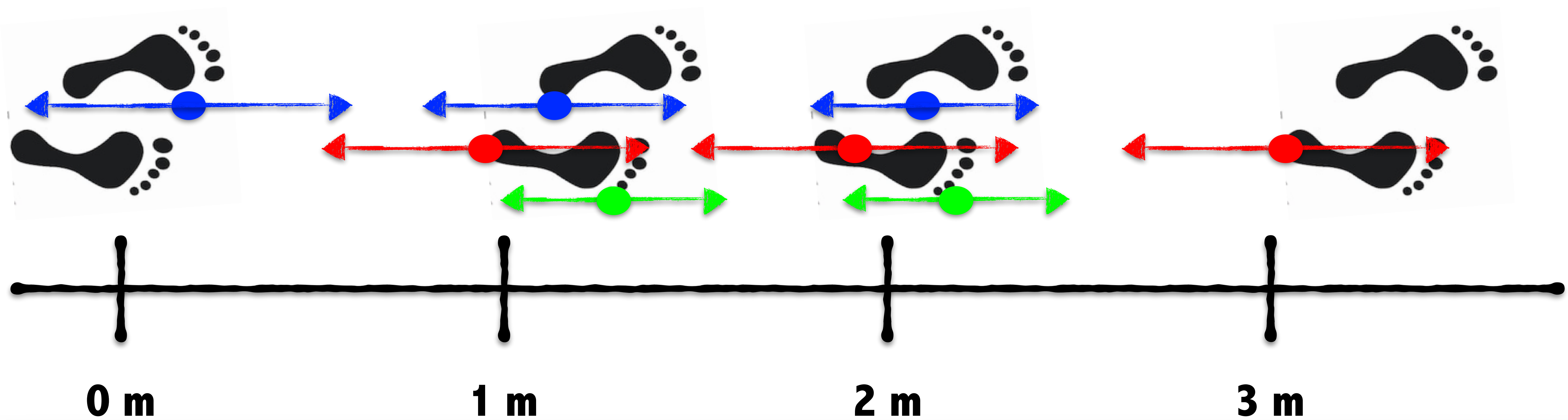

0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$


0 m
1 m
2 m
3 m

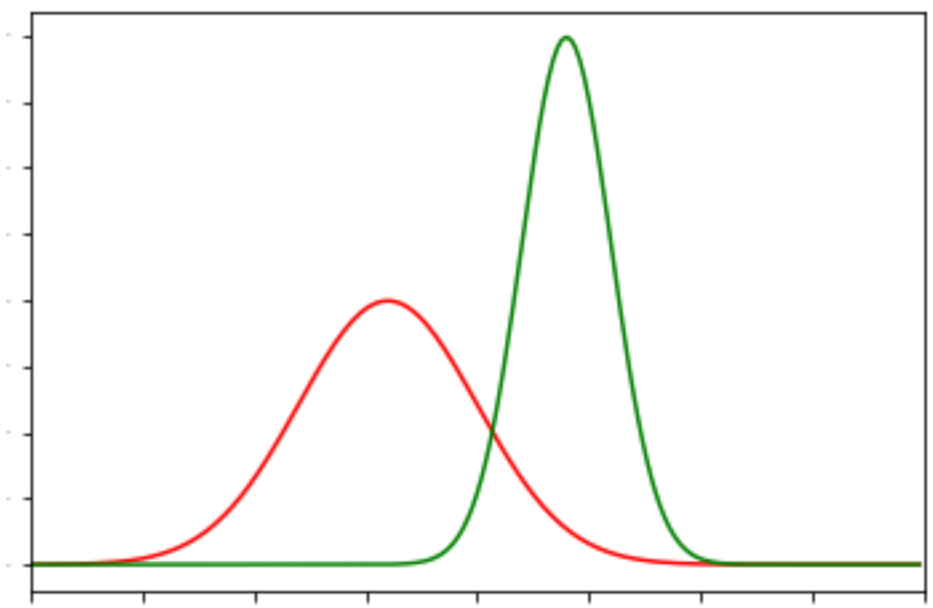
경로 추정 과정

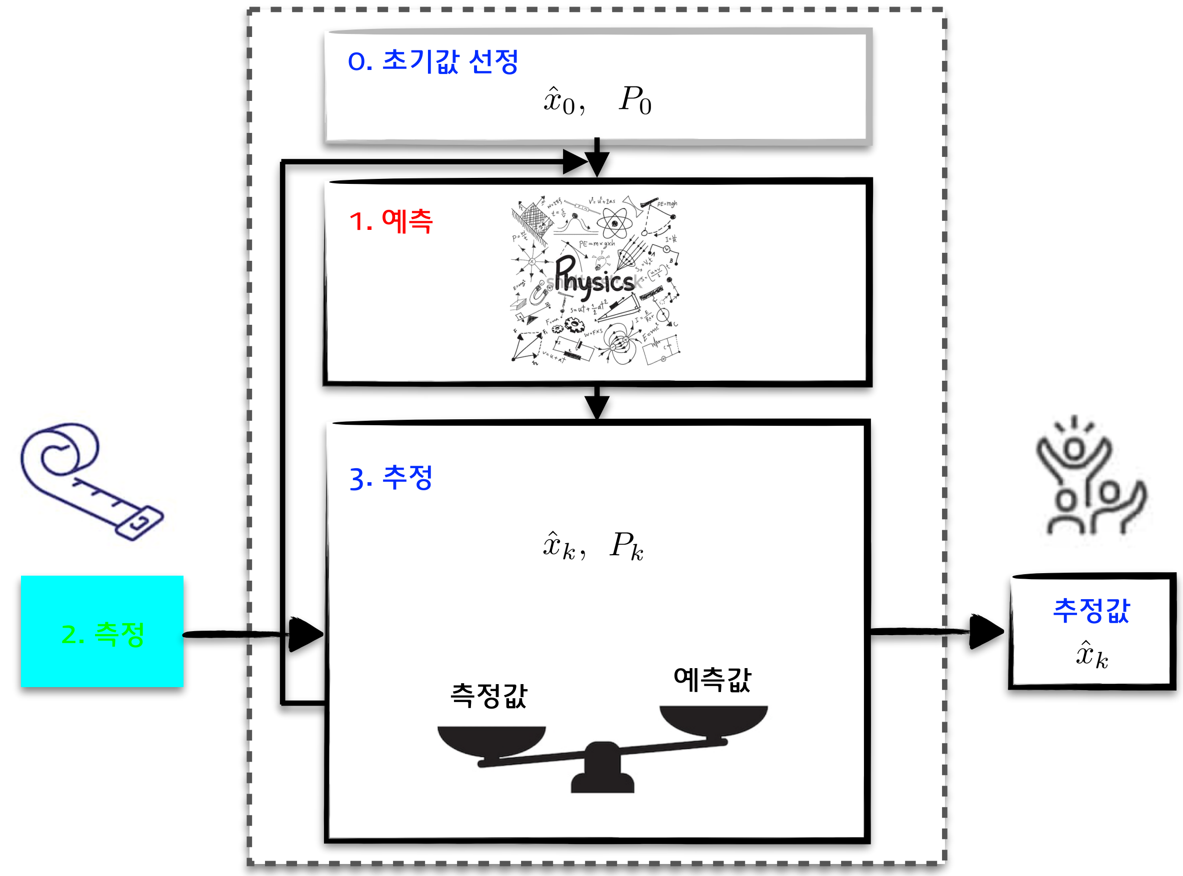


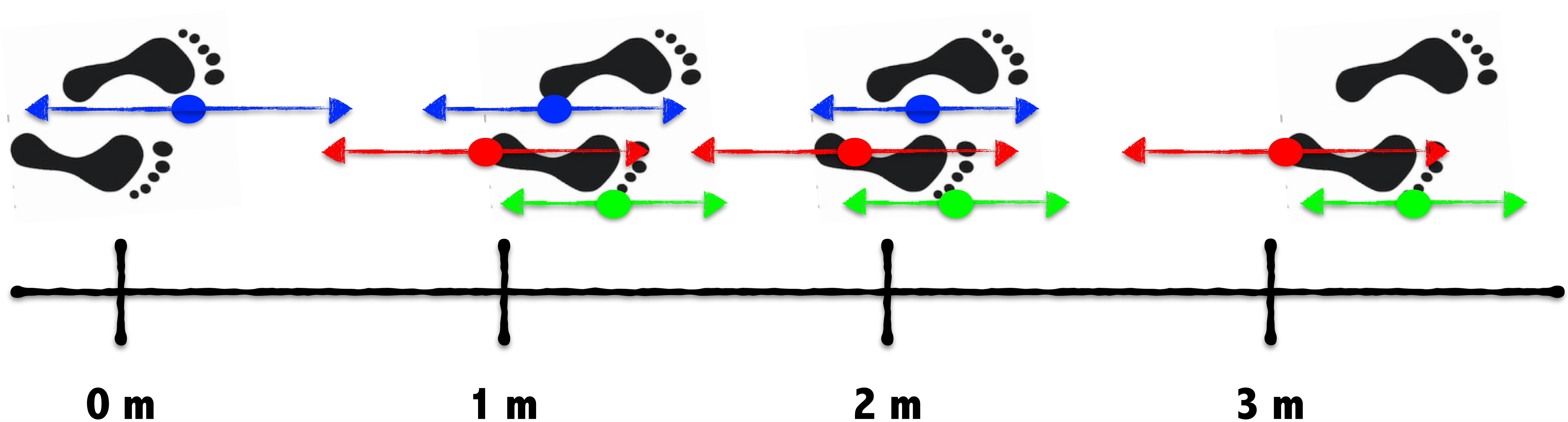

0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$


0 m
1 m
2 m
3 m

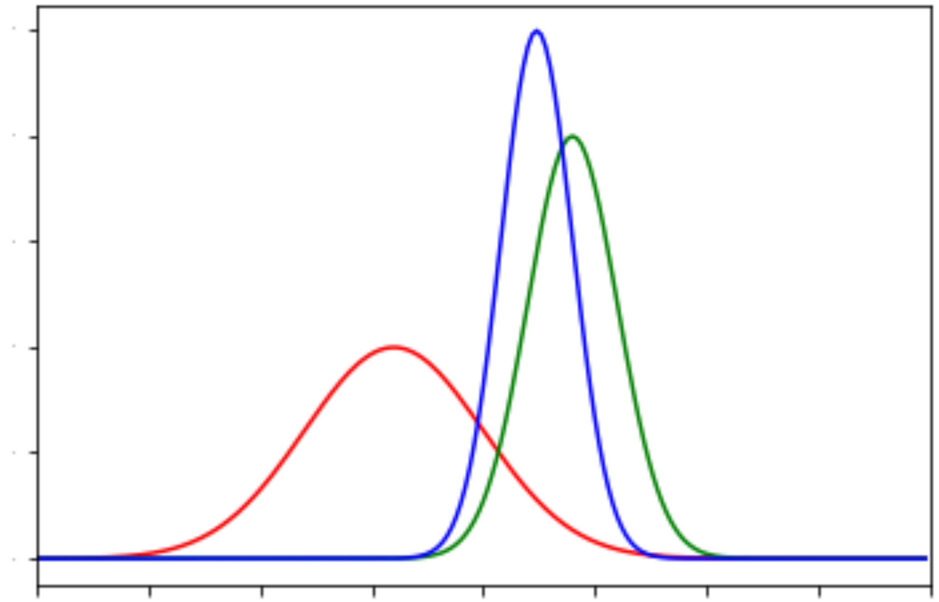
경로 추정 과정

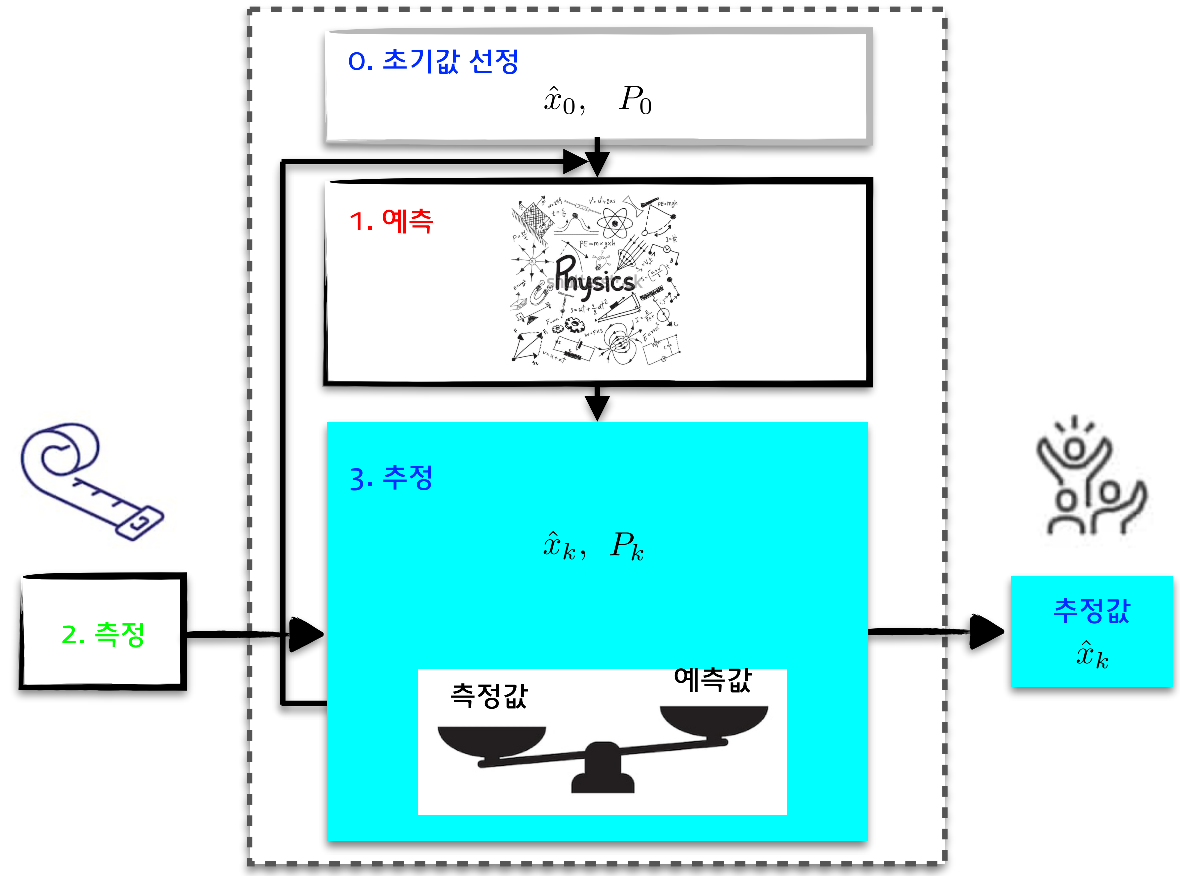



0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$

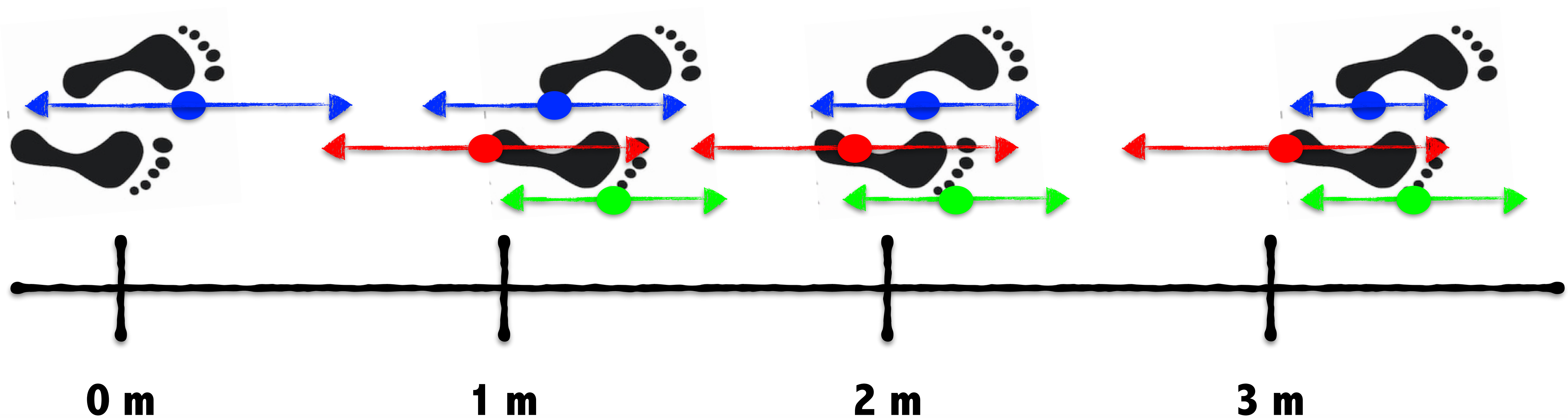

0 m
1 m
2 m
3 m

경로 추정 과정



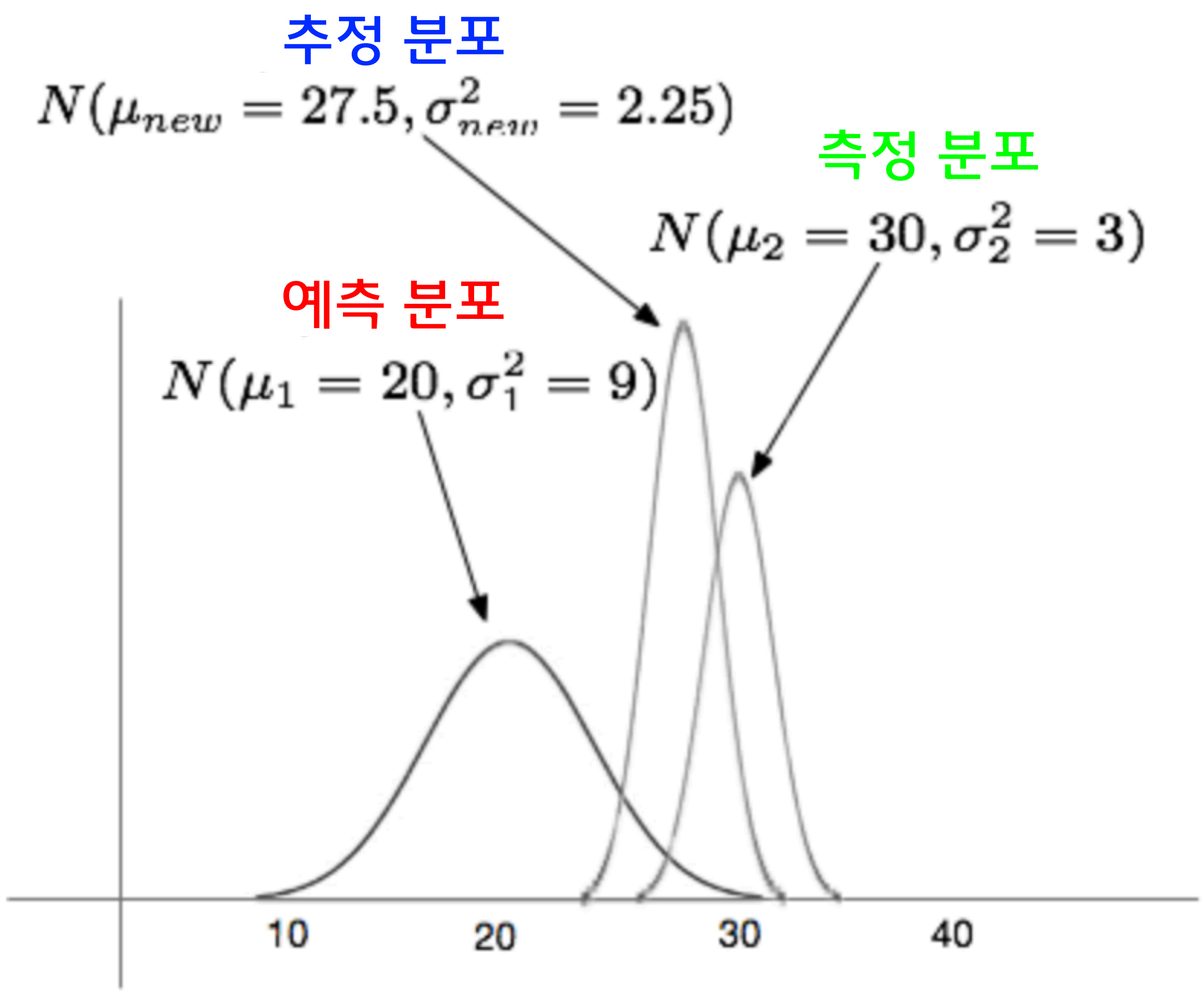


0. 초기값 선정
$\hat{x}_0, \quad P_0$
1. 예측
Physics
2. 측정
3. 추정
$\hat{x}_k, \quad P_k$
측정값
예측값
추정값
$\hat{x}_k$



0 m
1 m
2 m
3 m

위치 추정값의 확률 분포

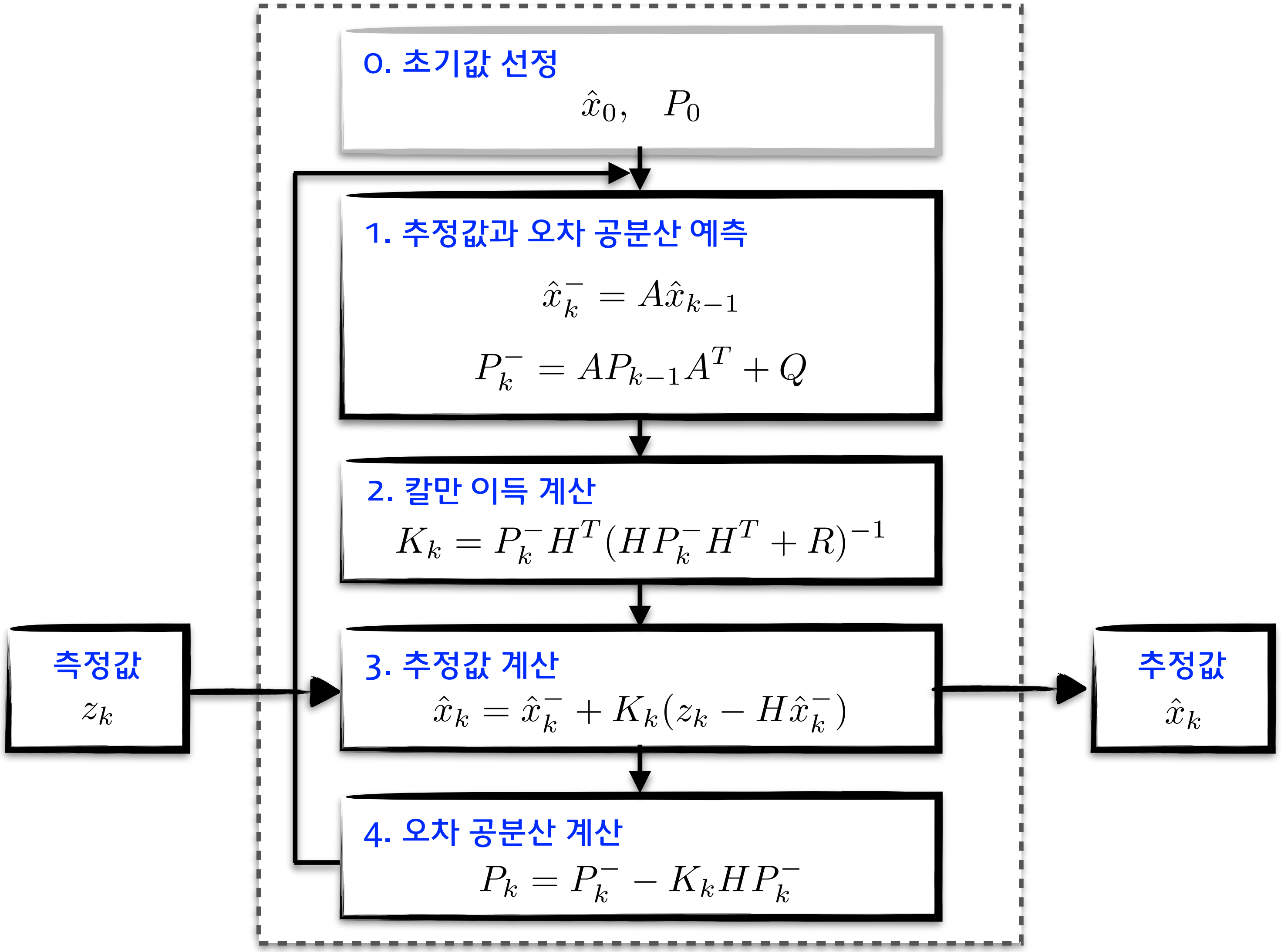


$$\mu_{new} = \frac{\mu_2\sigma_1^2 + \mu_1\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

$$\sigma^2_{new} = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

참고자료: Kalman filter 소개 (신동원님)

칼만 필터 알고리즘 & 코드




```python
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
```

칼만 필터: 전압 측정 예제

- 10초 동안 0.2초 간격으로 배터리의 전압 측정

  - 배터리의 명목 전압 = 14.4 V

  - 측정 잡음 ~ Gauss(0, 2^2)

- 칼만 필터를 사용하여 배터리의 전압을 추정하자!

https://github.com/tbmoon/kalman_filter/blob/master/Ch08.SimpleKalmanFilter/simple_kalman_filter.ipynb

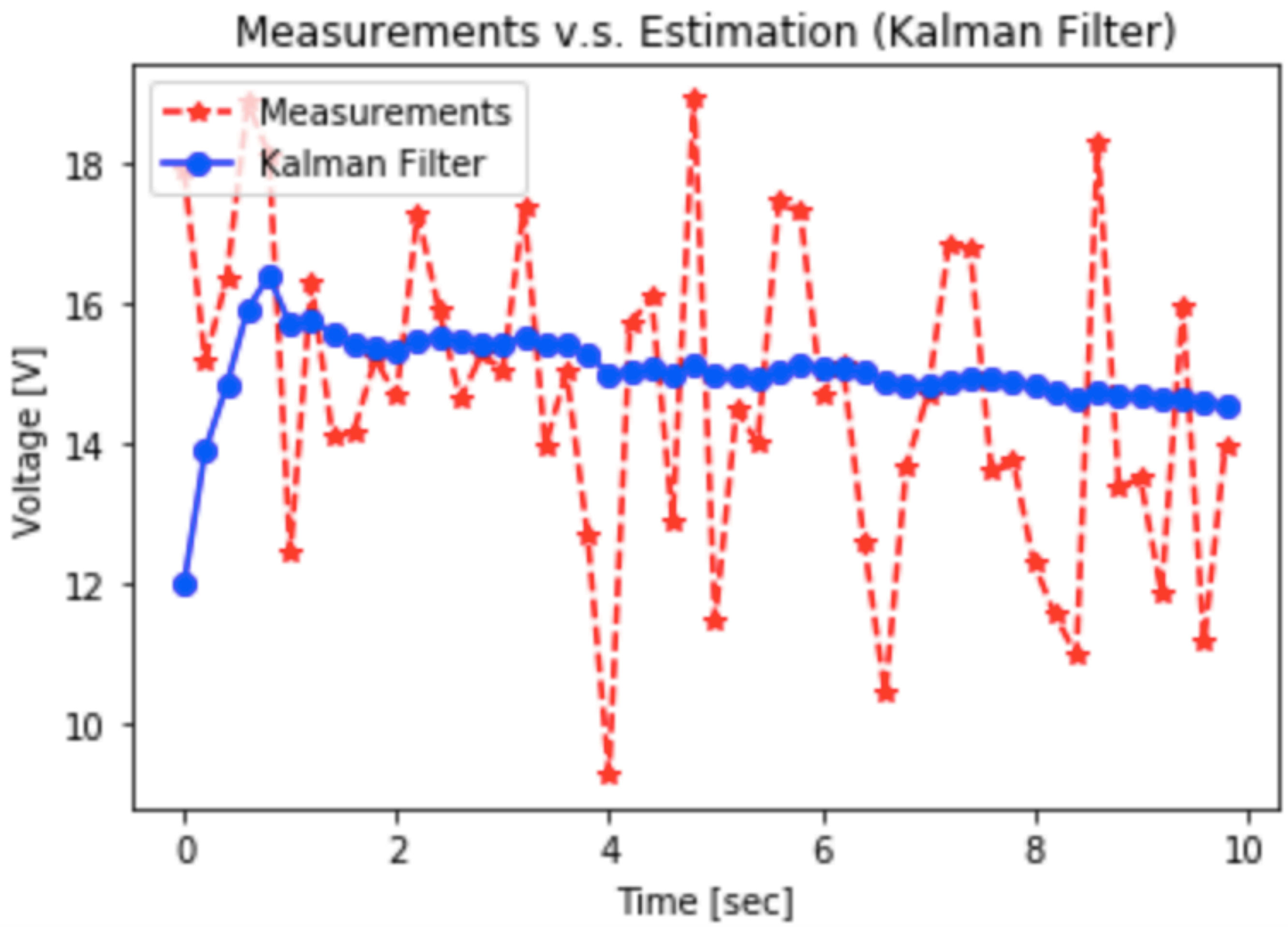
칼만 필터: 전압 측정 예제

- 10초 동안 0.2초 간격으로 배터리의 전압 측정

  - 배터리의 명목 전압 = 14.4 V

  - 측정 잡음 ~ Gauss(0, 2^2)

- 칼만 필터를 사용하여 배터리의 전압을 추정하자!



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 시스템 변화
- H : 측정값과 상태 변수 관계
- $Q = \sigma_w^2$
- $R = \sigma_v^2$

선형 상태 & 시스템 모델

선형 상태 모델

- $x_{k+1} = x_k$

  $w_k \sim 0$

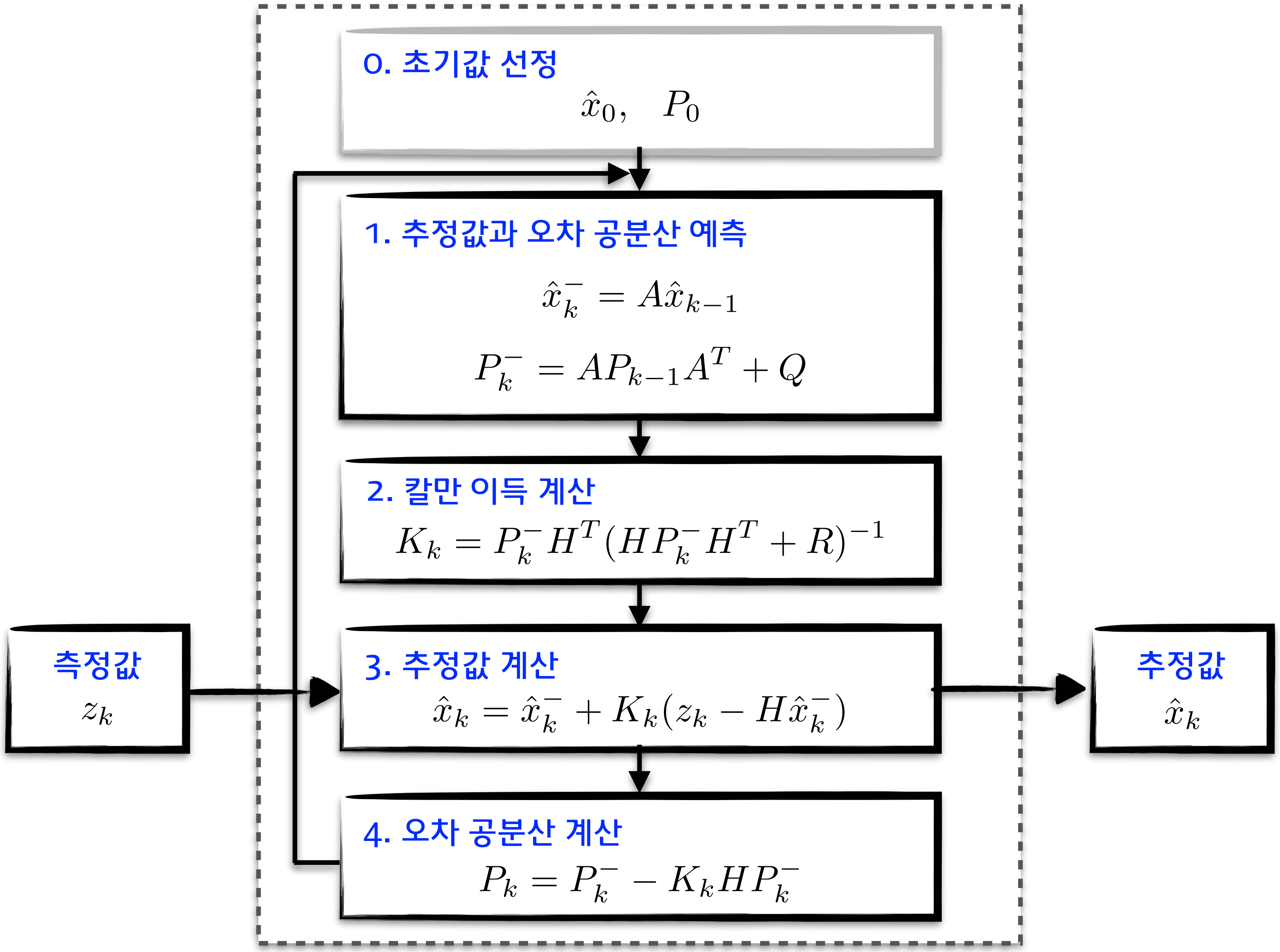
- $z_k = x_k + v_k$

  $v_k \sim Gauss(0, \sigma_v^2)$

시스템 모델

- $A = 1$
- $H = 1$
- $Q = 0$
- $R = 4[V^2]$

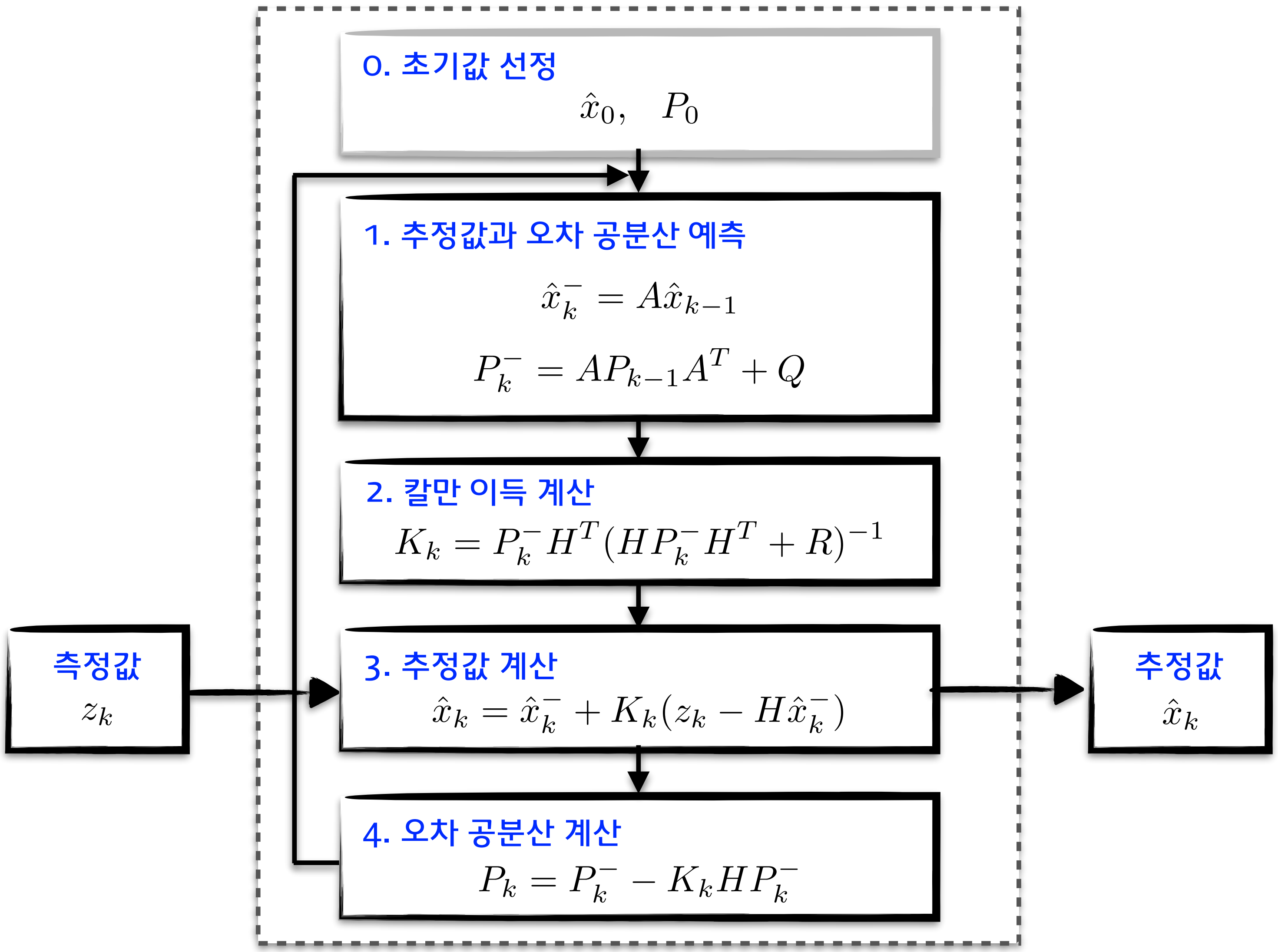
칼만 필터 알고리즘 & 코드




```python
# Initialization for estimation.
x_0 = 12  # 14 for book.
P_0 = 6
```

```python
def kalman_filter(z_meas, x_esti, P):
    """Kalman Filter Algorithm for One Variable."""
    # (1) Prediction.
    x_pred = A * x_esti
    P_pred = A * P * A + Q

    # (2) Kalman Gain.
    K = P_pred * H / (H * P_pred * H + 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
```

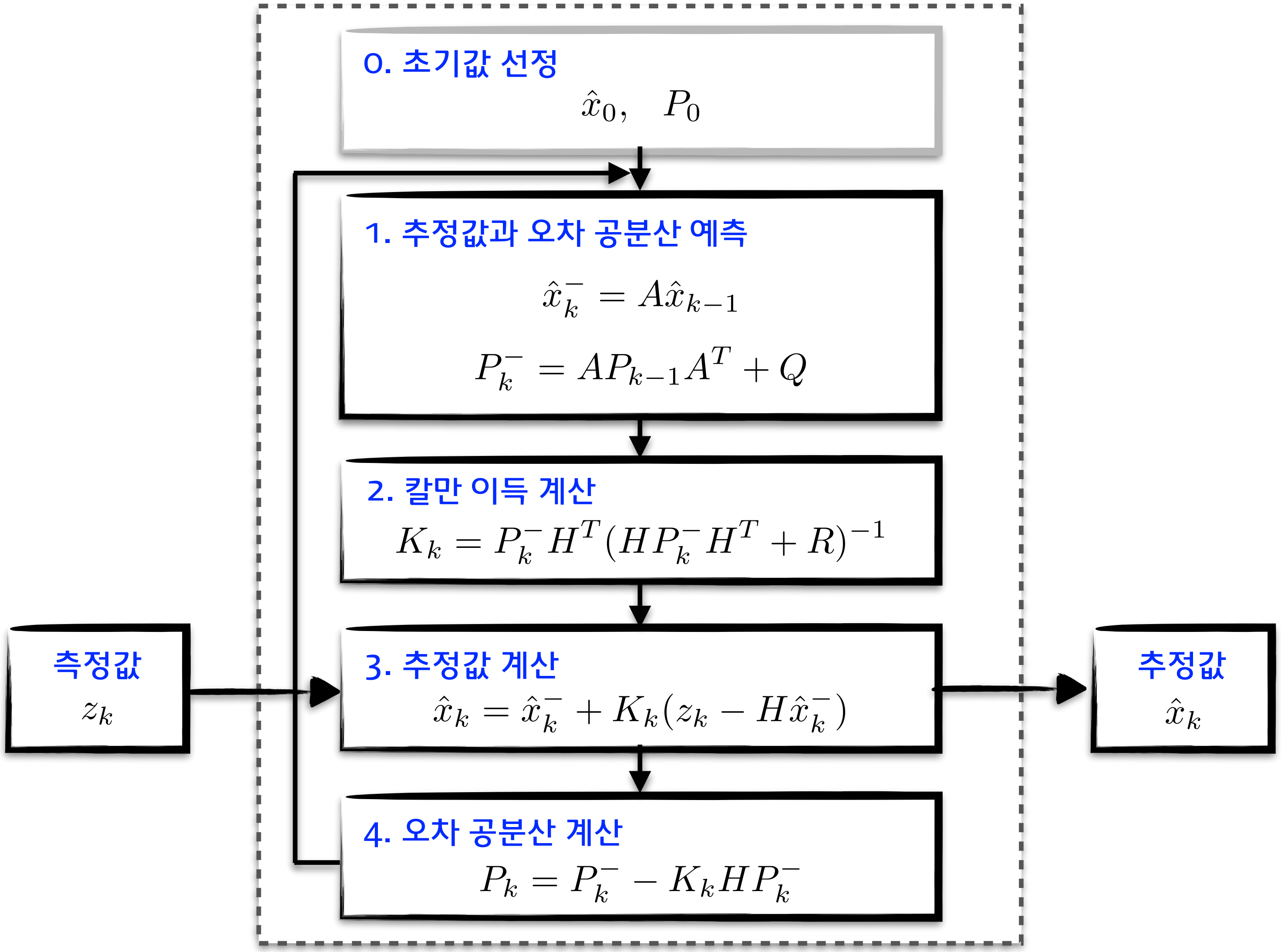
칼만 필터 알고리즘 & 코드




```python
# Initialization for estimation.
x_0 = 12  # 14 for book.
P_0 = 6
```

```python
def kalman_filter(z_meas, x_esti, P):
    """Kalman Filter Algorithm for One Variable."""
    # (1) Prediction.
    x_pred = x_esti
    P_pred = P

    # (2) Kalman Gain.
    K = P_pred / (P_pred + R)

    # (3) Estimation.
    x_esti = x_pred + K * (z_meas - x_pred)

    # (4) Error Covariance.
    P = P_pred - K * P_pred

    return x_esti, P
```

칼만 필터 알고리즘 & 코드




```python
# Initialization for estimation.
x_0 = 12  # 14 for book.
P_0 = 6
```

```python
def kalman_filter(z_meas, x_esti, P):
    """Kalman Filter Algorithm for One Variable."""
    # (1) Prediction.
    x_pred = x_esti
    P_pred = P

    # (2) Kalman Gain.
    K = P_pred / (P_pred + R)

    # (3) Estimation.
    x_esti = (1 - K) * x_pred + K * z_meas

    # (4) Error Covariance.
    P = (1 - K) * P_pred

    return x_esti, P
```

칼만 필터 흐름 확인

```python
# Initialization for system model.
A = 1
H = 1
Q = 0
R = 4
# Initialization for estimation.
x_0 = 12  # 14 for book.
P_0 = 6
```

```python
def kalman_filter(z_meas, x_esti, P):
    """Kalman Filter Algorithm for One Variable."""
    # (1) Prediction.
    x_pred = x_esti
    P_pred = P

    # (2) Kalman Gain.
    K = P_pred / (P_pred + R)

    # (3) Estimation.
    x_esti = (1 - K) * x_pred + K * z_meas

    # (4) Error Covariance.
    P = (1 - K) * P_pred

    return x_esti, P
```

k	x 측정 [V]	x 추정 [V]	P 추정 [V^2]	x 예측 [V]	P 예측 [V^2]	칼만 이득
0		12	6			
1	15.20					
2	16.35					
3	18.88					
4	18.14					

채워 보세요!

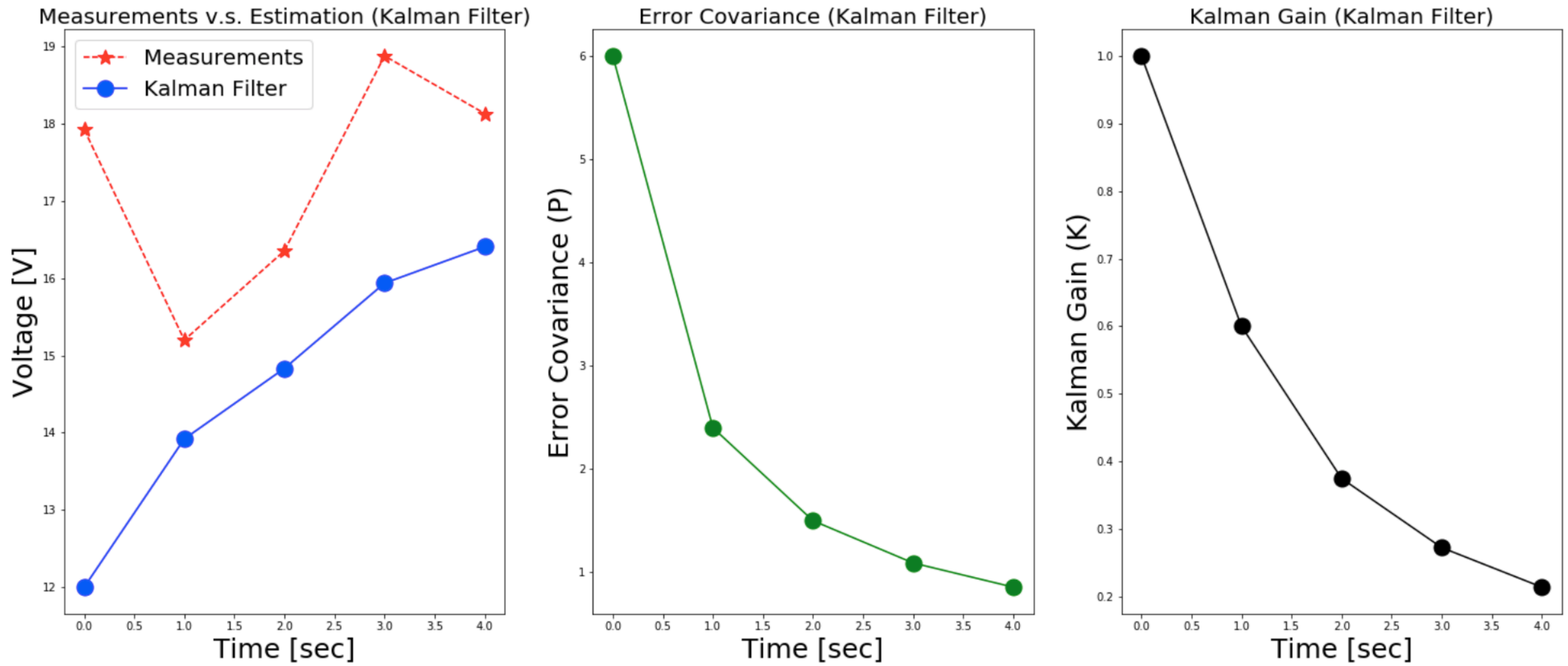
칼만 필터 흐름 확인

```
# Initialization for system model.
A = 1
H = 1
Q = 0
R = 4
# Initialization for estimation.
x_0 = 12  # 14 for book.
P_0 = 6
```

```
def kalman_filter(z_meas, x_esti, P):
    """Kalman Filter Algorithm for One Variable."""
    # (1) Prediction.
    x_pred = x_esti
    P_pred = P

    # (2) Kalman Gain.
    K = P_pred / (P_pred + R)

    # (3) Estimation.
    x_esti = (1 - K) * x_pred + K * z_meas

    # (4) Error Covariance.
    P = (1 - K) * P_pred

    return x_esti, P
```

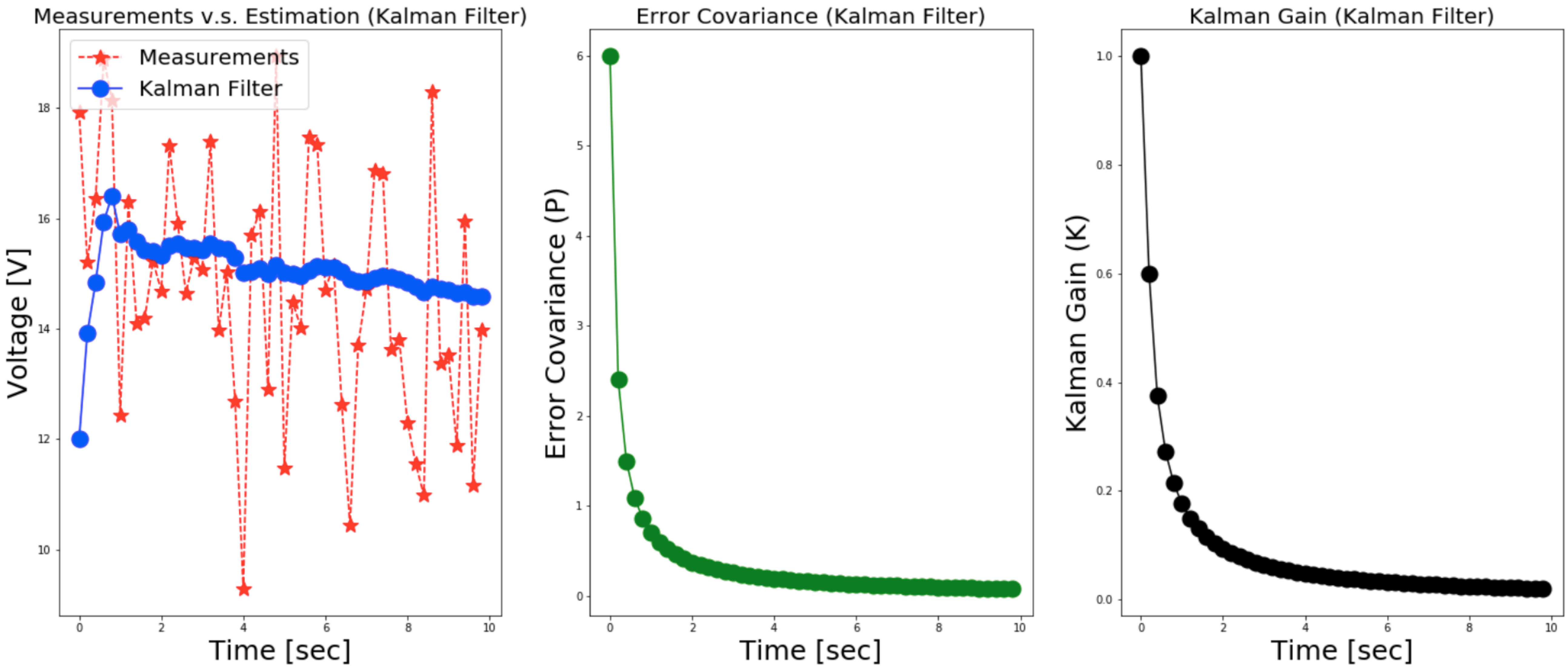
k	x 측정 [V]	x 추정 [V]	P 추정 [V^2]	x 예측 [V]	P 예측 [V^2]	칼만 이득
0		12	6			
1	15.20	13.92	2.40	12	6	0.60
2	16.35	14.83	1.50	13.92	2.40	0.38
3	18.88	15.94	1.09	14.83	1.50	0.27
4	18.14	16.41	0.86	15.94	1.09	0.21

칼만 필터 흐름 확인



k	x 측정 [V]	x 추정 [V]	P 추정 [V^2]	x 예측 [V]	P 예측 [V^2]	칼만 이득
0		12	6			
1	15.20	13.92	2.40	12	6	0.60
2	16.35	14.83	1.50	13.92	2.40	0.38
3	18.88	15.94	1.09	14.83	1.50	0.27
4	18.14	16.41	0.86	15.94	1.09	0.21

칼만 필터: 전압 측정 결과



Measurements v.s. Estimation (Kalman Filter)
Measurements
Kalman Filter
Voltage [V]
Time [sec]
Error Covariance (Kalman Filter)
Error Covariance (P)
Time [sec]
Kalman Gain (Kalman Filter)
Kalman Gain (K)
Time [sec]

칼만 필터 파라미터

- 초기 위치 추정량 (x_0)이 커지면 어떻게 될까?
- 오차 공분산 (P)이 커지면 어떻게 될까?
- 측정 잡음 공분산 (R)이 커지면 어떻게 될까?
- 칼만 이득 (K)이 커지면 어떻게 될까?

참고 자료

- 칼만 필터는 어렵지 않아 (저자: 김성필 님)
- 파이썬으로 구현하는 칼만 필터
- Kalman filter 소개 (신동원 님)
- SLAM: KF & EKF (정진용 님)
- Robot Mapping (Cyrill Stachniss)
- The Kalman Filter (Michel van Biezen)

풀잎 스쿨 11 주간 일정

~~01 주차 1월 08일 - 재귀 필터 (퍼실이)~~

~~02 주차 1월 15일 - 칼만 필터 기초 & 초간단 칼만 필터 예제 (퍼실이)~~

03 주차 1월 22일 - 칼만 필터 기초 & 위치로 속도 추정하기 (퍼실이)

04 주차 1월 29일 - 영상 속의 물체 추정하기

05 주차 2월 05일 - 기울기 자세 측정하기

06 주차 2월 12일 - 기울기 자세 측정하기

07 주차 2월 19일 - 확장 칼만 필터

08 주차 2월 26일 - 확장 칼만 필터

완료

09 주차 3월 04일 - 무향 칼만 필터

10 주차 3월 11일 - 무향 칼만 필터

11 주차 3월 18일 - 파틱클 필터 혹은 정리/쫑 파티

예정

감사합니다