

RSSI를 이용한 거리 측정 최적화 방안 연구

수학과 201621120 최동헌

수학과 201621136 이재협

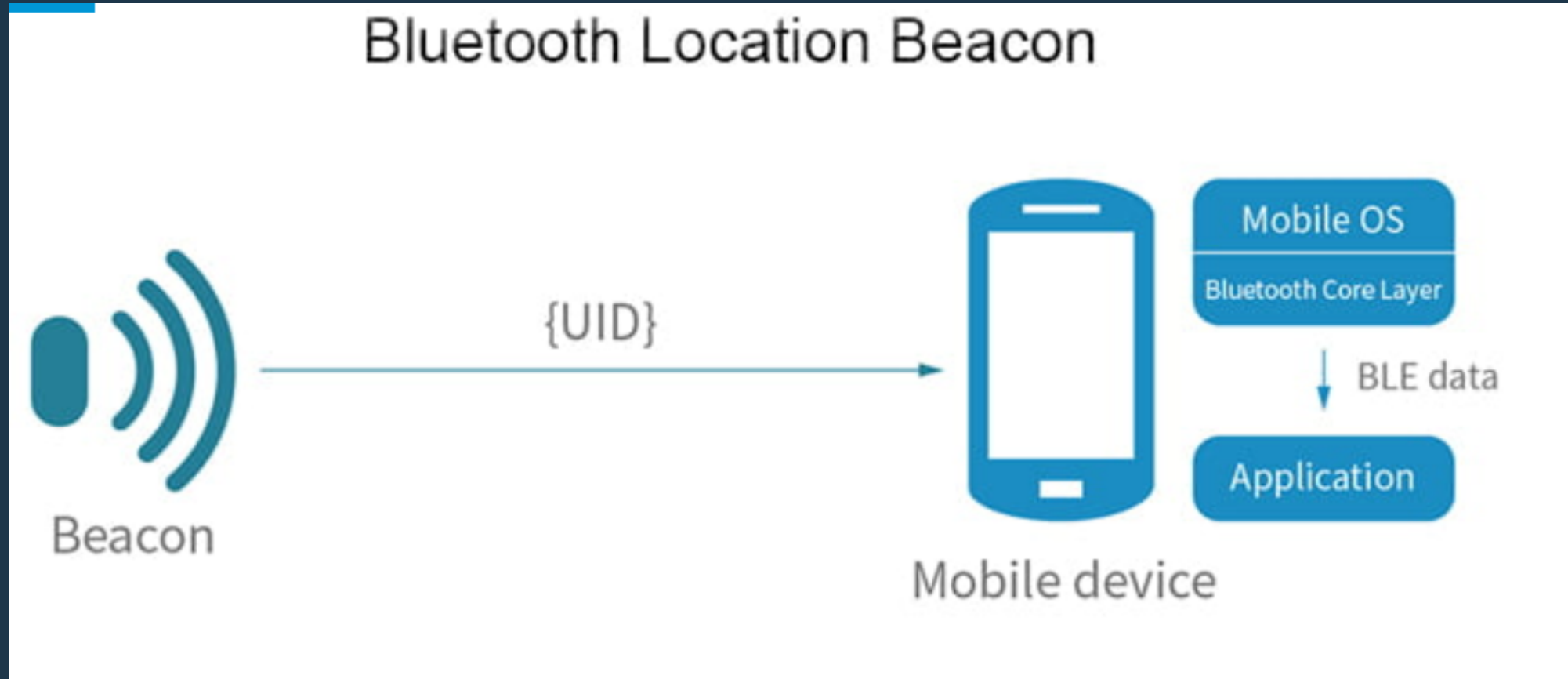
수학과 201621138 허창현

수학과 201621148 백현규

Contents

- 1 Bluetooth Beacon and RSSI
- 2 Calculating RSSI/Distance with Python
- 3 Kalman Filter

1. Bluetooth Beacon and RSSI



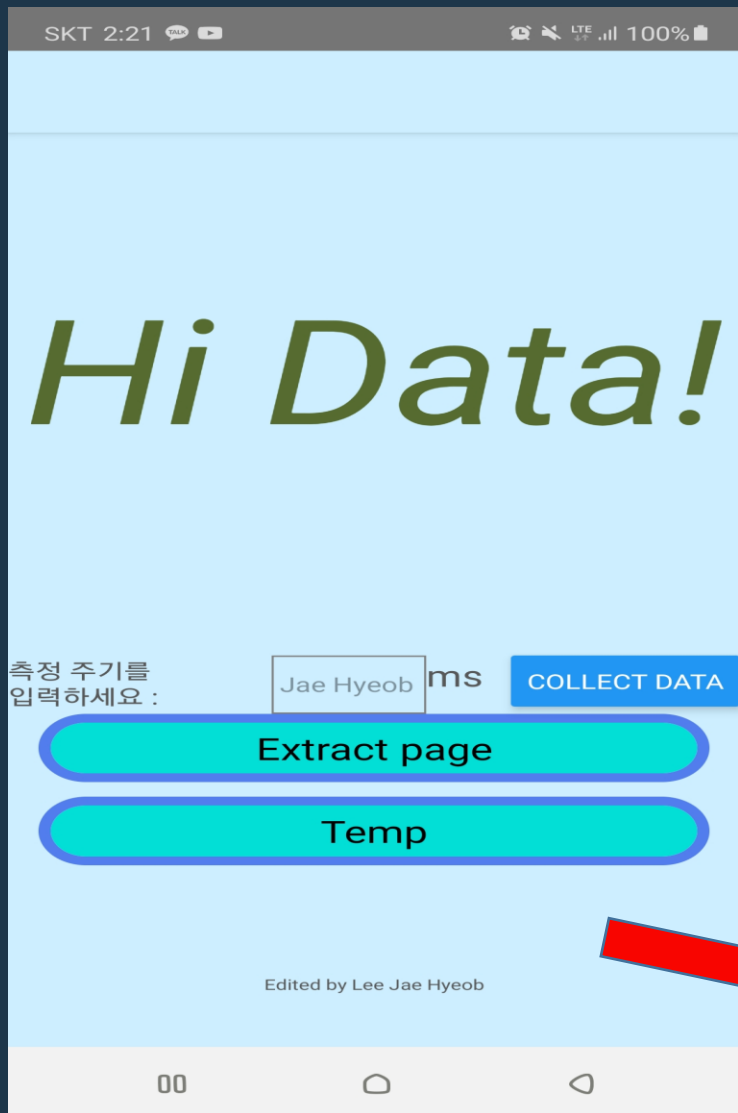
1. Bluetooth Beacon and RSSI

$$RSSI = -10n \log_{10}(d) + A,$$

- RSSI를 통해 거리를 측정하는 데 노이즈로 인해 정확도가 낮음.
- RSSI값이 1만 변해도 거리는 몇 m씩 변함
- 정확도를 높일 수 있는 방법을 연구하고자 함.

2. Calculating RSSI/Distance with Python

측정된 RSSI값



H	I	J	K	L	M	N	O	P	Q	R	S	T
major	100	minor	3	proximity	near	rss	-70	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-73	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	4	proximity	near	rss	-61	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	2	proximity	near	rss	-64	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-66	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-67	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	2	proximity	near	rss	-62	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	2	proximity	near	rss	-61	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	2	proximity	near	rss	-60	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-66	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-67	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-67	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	2	proximity	near	rss	-61	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-65	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	2	proximity	near	rss	-64	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-70	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-70	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-73	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	4	proximity	near	rss	-61	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			
major	100	minor	3	proximity	near	rss	-67	uid	e2c56db5-dffb-48d2-b060-d0f5a71096e0]			

안드로이드 스마트폰이 받은 RSSI 데이터를 정리하기 위해 만들어진 어플리케이션임.

2. Calculating RSSI/Distance with Python

```
class Beacon:
    x, y, N, M_Power = 0, 0, 0, 0
    mu_RSS, sigma_RSS = 0, 0
    dataset = []

    def __init__(self, x, y, N, M_Power):
        self.x = x
        self.y = y
        self.N = N
        self.M_Power = M_Power

    def RSSI_to_distance(self, RSSI):
        return math.pow(10, ((self.M_Power - RSSI) / (10 * self.N)))

    def Dist_to_RSSI(self, Dist):
        return ((-10) * self.N * math.log(Dist, 10) + self.M_Power)

    def set_mu_sigma_RSS(self, a, b, c):
        temp = []
        temp.append(a)
        temp.append(b)
        temp.append(c)
        self.mu_RSS, self.sigma_RSS = np.mean(temp), np.std(temp)

    def add_RSS(self, RSS):
        self.dataset.append(RSS)
        self.mu_RSS = np.mean(self.dataset)
        self.sigma_RSS = np.std(self.dataset)
```

1. RSSI를 통해서 거리를 구하는 함수

2. 거리를 이용해서 RSSI를 구하는 함수

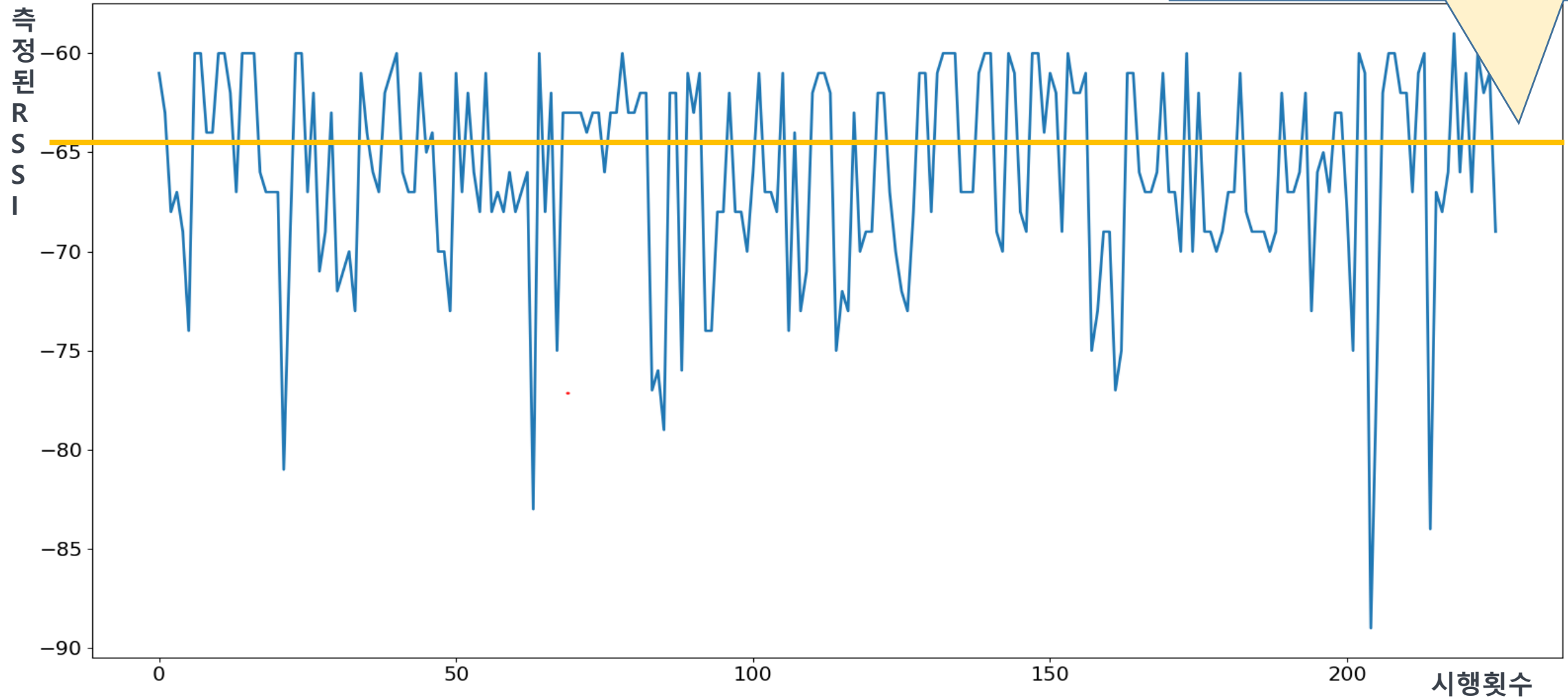
3. 평균, 표준편차

2. Calculating RSSI/Distance with Python

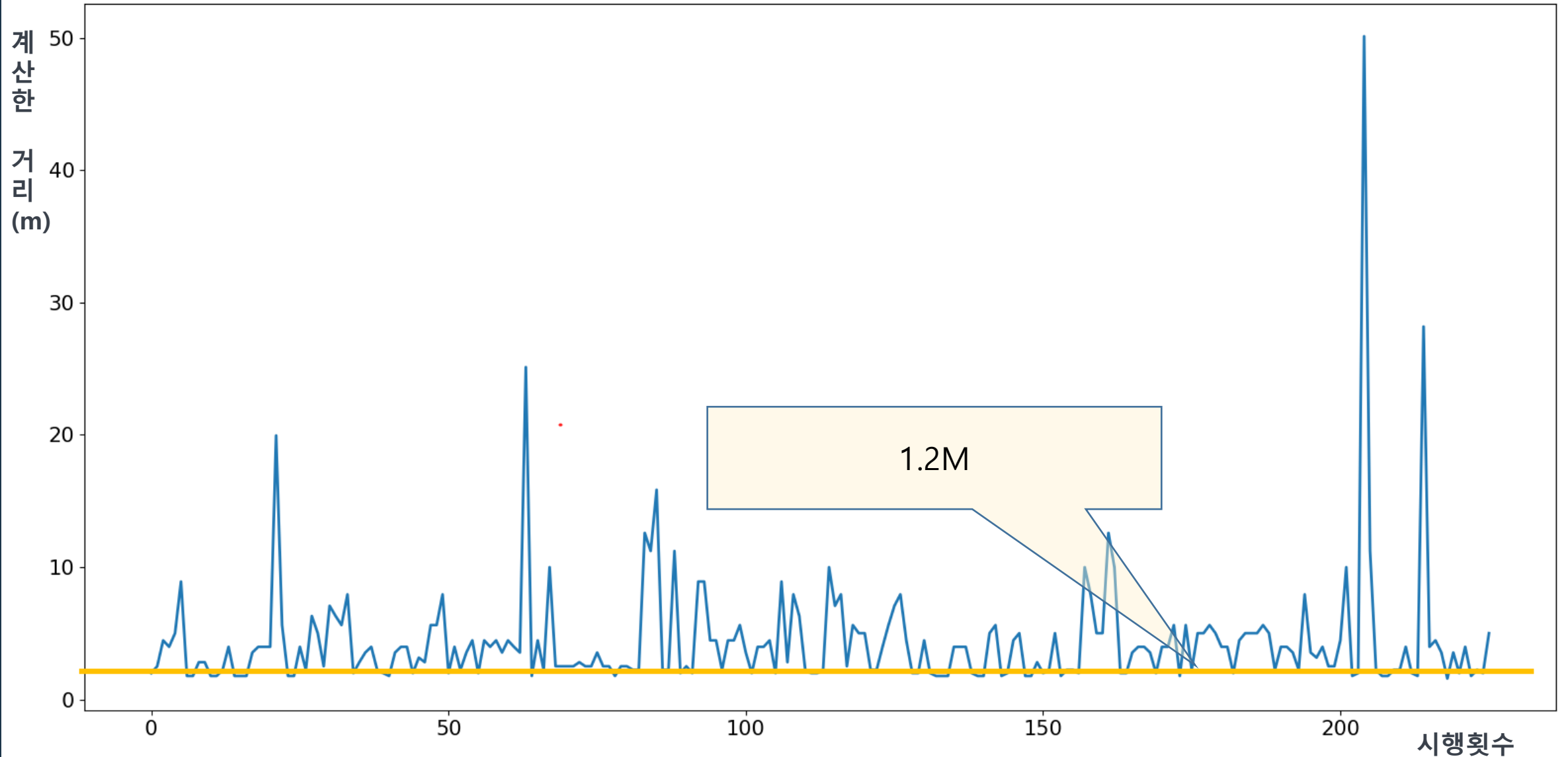
실제 거리	이론상 RSSI(평균)	실제 RSSI(평균)	RSSI로 측정한 거리
0.5m	-57.97	-61.37	0.73
0.8m	-62.06	-63.19	0.91
1.2m	-65.58	-65.98	1.25
1.6m	-68.08	-65.22	1.15
2.0m	-70.02	-65.80	1.23
2.4m	-72.94	-70.29	2.06
2.8m	-75.12	-66.72	1.36

2. Calculating RSSI/Distance with Python(1.2m)

1.2M일 때, 이론 상 RSSI



2. Calculating RSSI/Distance with Python



3. Kalman Filter

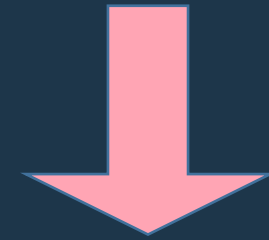
$$\bar{x}_k = \alpha \bar{x}_{k-1} + (1 - \alpha) x_k$$

- 1차 저주파 필터
- α 는 weight를 뜻함.

3. Kalman Filter



- 일정 기간 주가의 산술 평균값을
이어서 만든 선
- 주식에서 일정기간의 평균을
의미 있는 보조 지표로 활용



칼만 필터

3. Kalman Filter

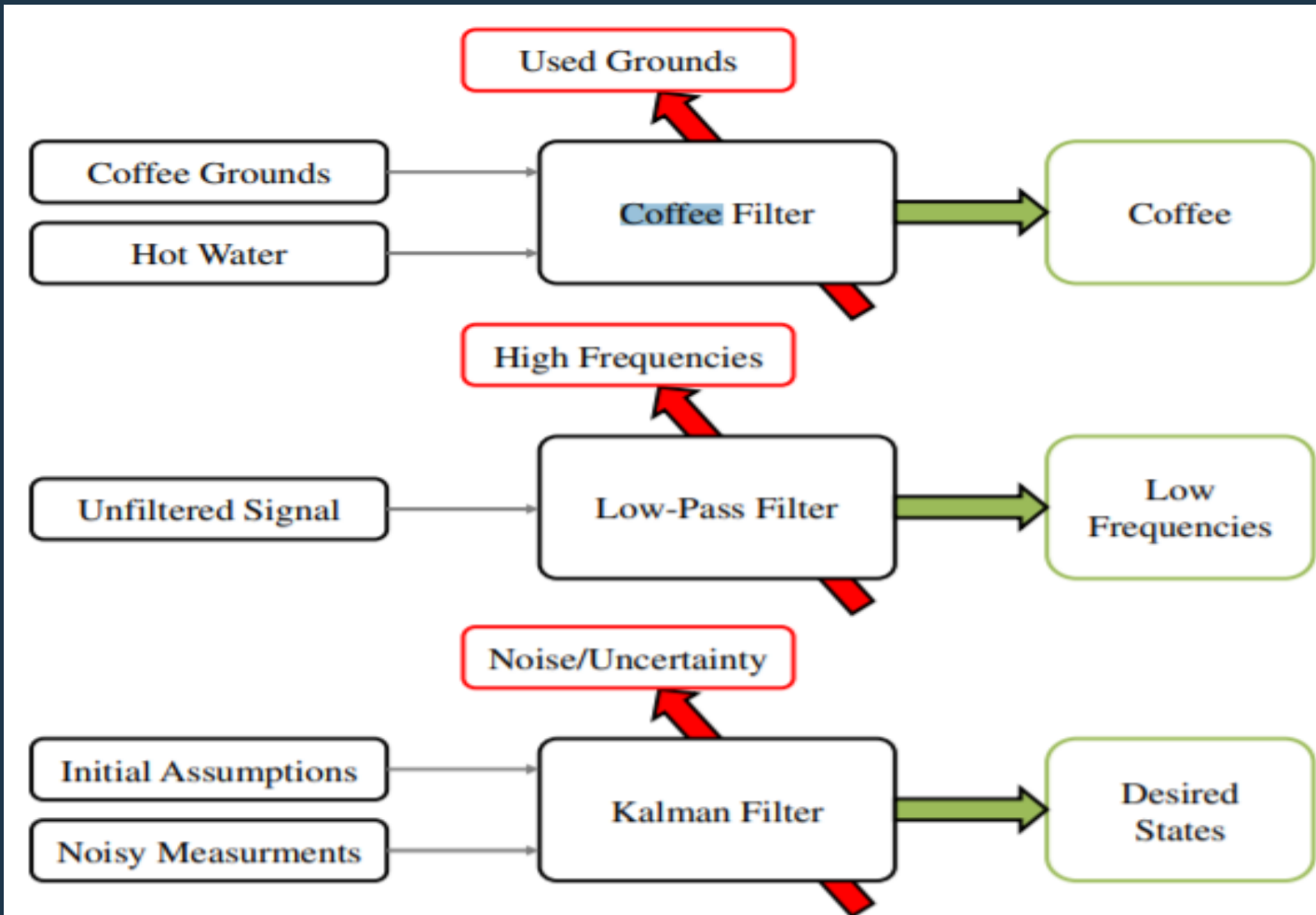
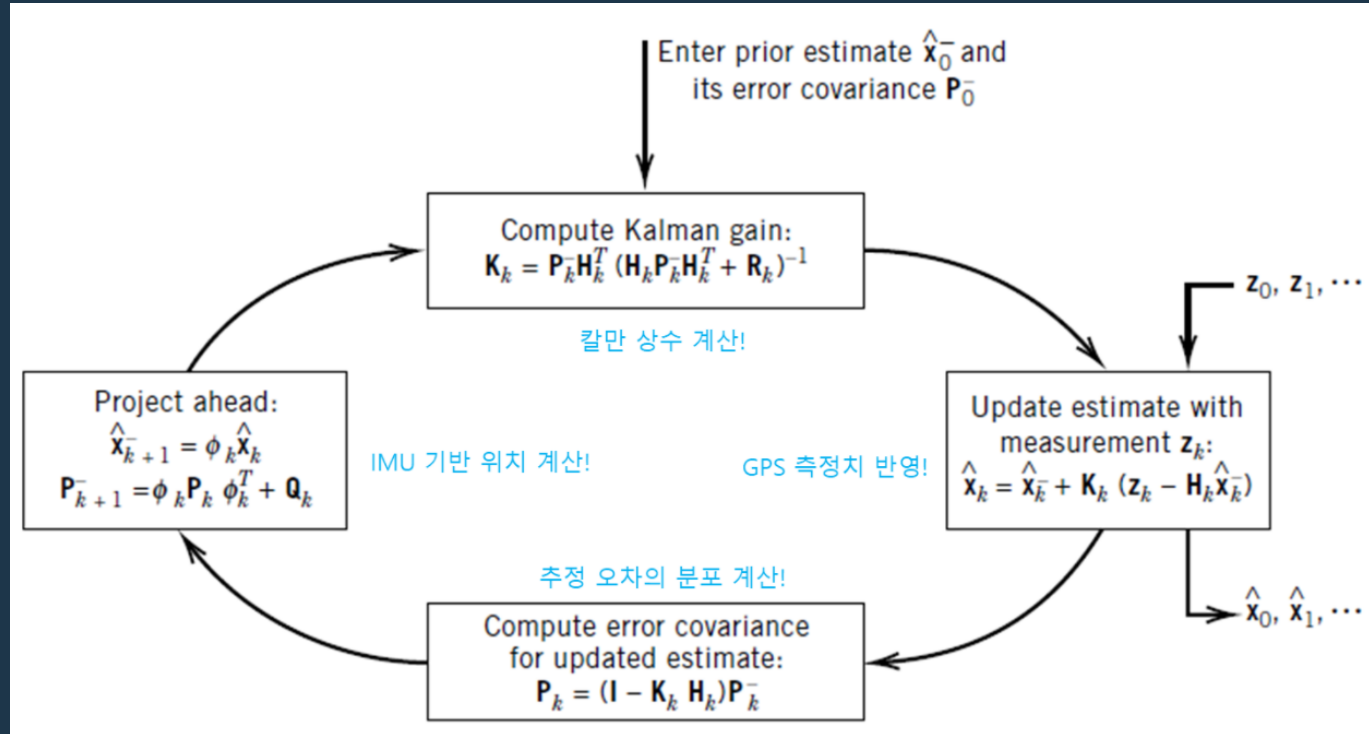


Figure 1. Three Example Diagrams of Types of Filters

3. Kalman Filter



1. 한 센서로 러프하게 추정값 계산
2. 추정 오차를 최소화 하는 칼만 상수 계산
3. 다른 센서의 값을 칼만 상수를 통해 반영하여 보정
4. 보정하고 나서 줄어든 추정 오차의 분포 계산

3. Kalman Filter

- N개의 데이터를 기반으로 예측

⇒ N이 너무 작으면 정확도 ↓, N이 너무 크면 실시간성 ↓

- Capping: Outlier를 제거(큰 Noise 방지)

⇒ Cut ratio를 0%~10% 정도로 사용

3. Kalman Filter - MatLab Library

1.AvgFilter	2021-06-30 오후 2:59	파일 폴더
2.MovAvgFilter	2021-06-30 오후 2:59	파일 폴더
3.LPF	2021-06-30 오후 2:59	파일 폴더
8.SimpleKalman	2021-09-14 오후 6:59	파일 폴더
9.DvKalman	2021-06-30 오후 2:59	파일 폴더
10.TrackKalman	2021-06-30 오후 2:59	파일 폴더
11.ARS	2021-06-30 오후 2:59	파일 폴더
12.EKF	2021-06-30 오후 2:59	파일 폴더
13.UKF	2021-06-30 오후 2:59	파일 폴더
14.PF	2021-06-30 오후 2:59	파일 폴더
15.HPF	2021-06-30 오후 2:59	파일 폴더
16.CompFilter	2021-06-30 오후 2:59	파일 폴더

```
TestSimpleKalman2.m  Kalman_RSS.m  +
1  function rss = Kalman_RSS(arr)
2  %
3  %
4  persistent A H Q R
5  persistent y P
6  persistent flagRun
7
8
9  if isempty(flagRun)
10     A = 1;    % 칼만 필터에서 필요한 parameter를 정의함.(A, H, Q, R)
11     H = 1;
12
13     Q = 1;
14     R = 1000;
15
16     length = size(arr,2) % arr 함수의 열의 갯수를 반환
17     cut_ratio = 0.10
18     num_of_cut = fix(length * cut_ratio) % 전체 열 갯수 중에서 cut_ratio 비율 만큼 잘라냄.
19
20     for a=1:num_of_cut % for문을 돌면서 최대, 최소값을 없앴
21         min_num = min(arr)
```

명령 창

```
ans =  
  
-66.8940
```

4. To Do List

```
1 function rss = Kalman_RSS(arr)
2 %
3 %
4 persistent A H Q R
5 persistent y P
6 persistent flagRun
7
8
9 if isempty(flagRun)
10     A = 1; % 칼만 필터에서 필요한
11     H = 1;
12
13     Q = 1;
14     R = 1000;
```

- Kalman Filter에서 사용하는 A,H,Q,R 값을 조정
- Cut Ratio 조정
- 결과 Data 정리 및 오차 측정