

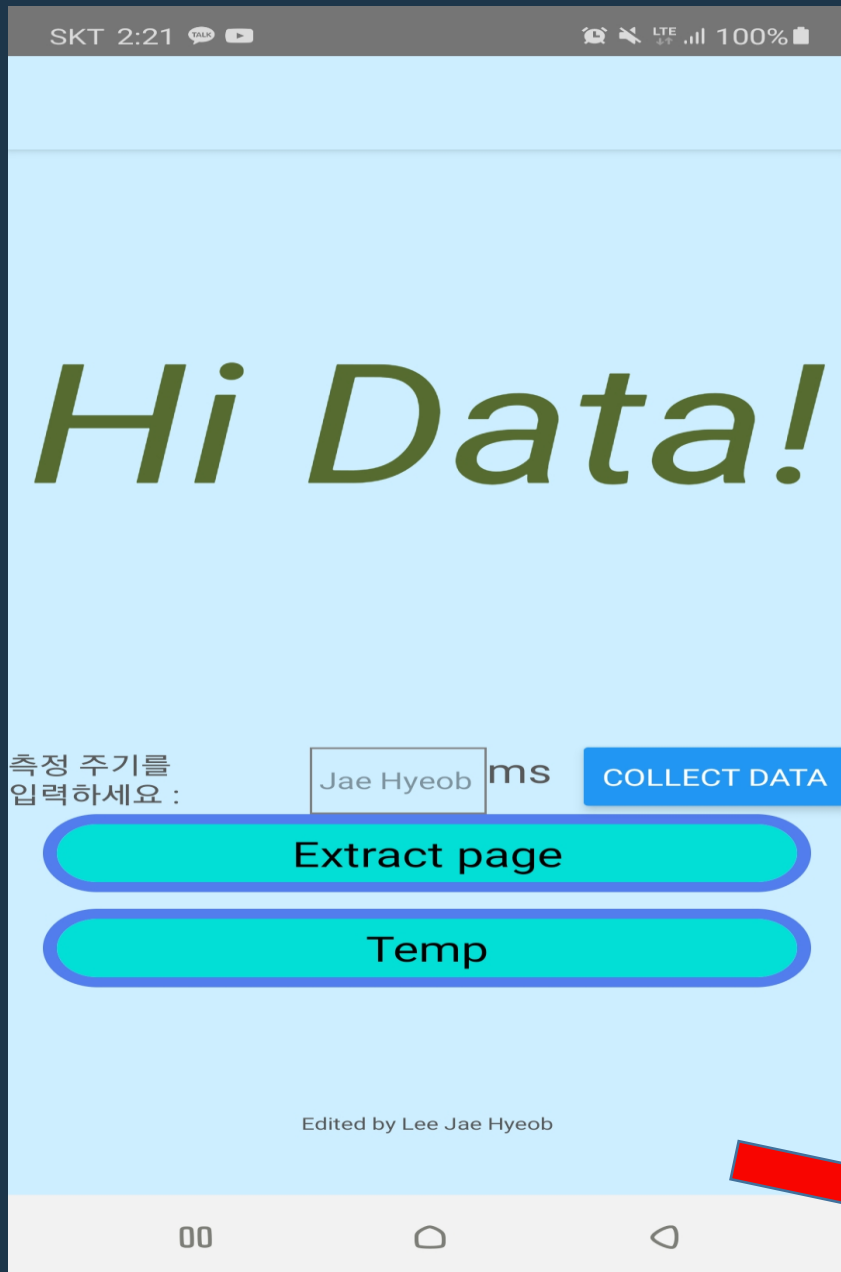
비콘 측위 수학적 알고리즘 관련 연구

Contents

- 1 블루투스를 이용한 실내 측위 방식
- 2 RSS를 관련 Formula
- 3 RSS에 영향을 주는 요인
- 4 RSS 세기 관련 연구
- 5 2 - Way 방식과 삼변측량 비교
- 6 정규분포를 이용한 측위 연구

1. 기존의 블루투스 실내 측위 방식 - FingerPrint

- 미리 해당 위치에서 측정되는 블루투스 신호들을 수집하여 유사도 계산
- 유사도 A와 유사도 B 사이의 우선 순위 부여 가능.(Tanimoto, cosine 유사도)
- 단점: 서비스 가능 지역의 모든 실내 지도와 신호 수집 필요

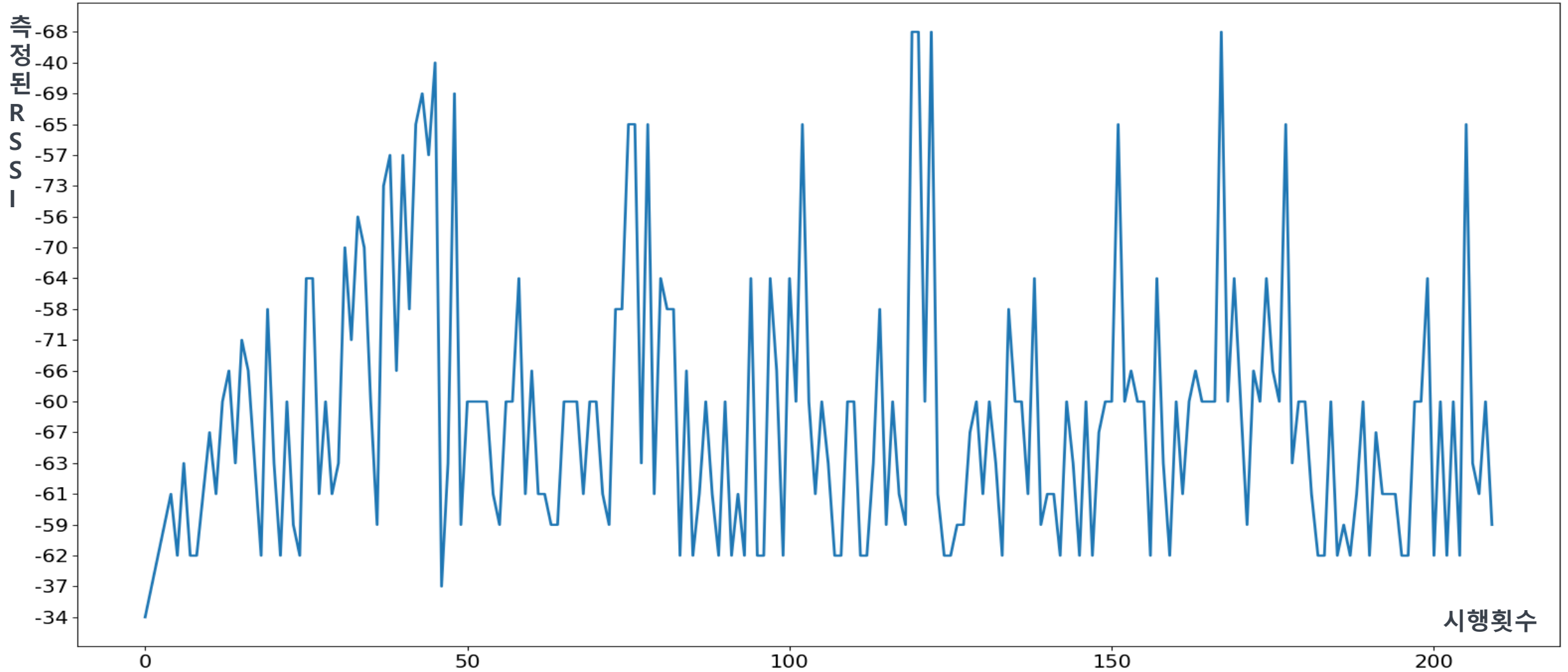


| H | I | J | K | L | M | N | O | P | Q | R | S | T |
|-------|-----|-------|---|-----------|------|-----|-----|-----|---------------------------------------|---|---|---|
| major | 100 | minor | 3 | proximity | near | rss | -70 | uid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 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] | | | |
| major | 100 | minor | 2 | proximity | near | rss | -62 | 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 | -68 | uid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 2 | proximity | near | rss | -61 | uid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 1 | proximity | near | rss | -68 | uid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |

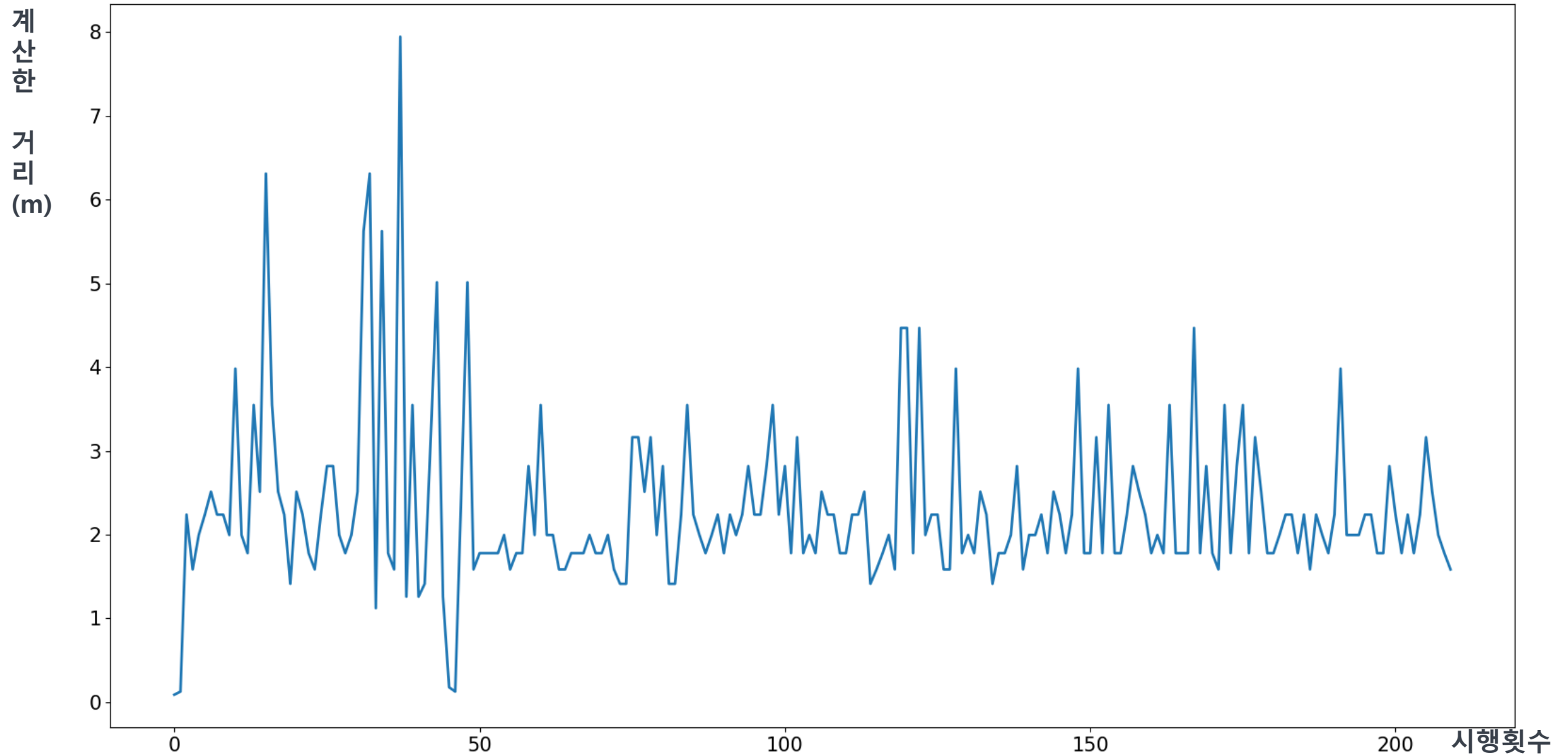
측정된 RSSI값

안드로이드 스마트폰이 받은 RSSI 데이터를 정리하기 위해 만들어진 어플리케이션임.

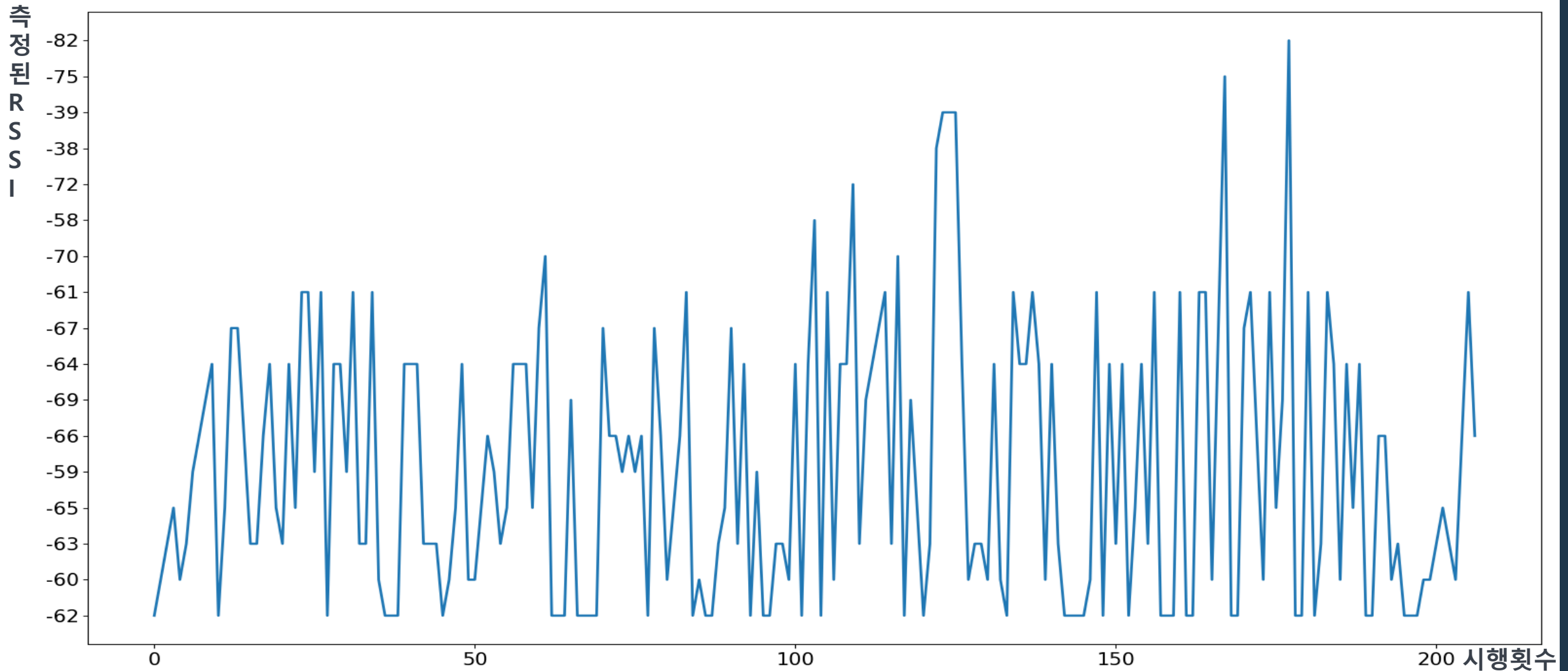
2. 실제로 측정한 RSSI (0.5m)



3. 측정된 RSSI로 계산한 거리(0.5m)

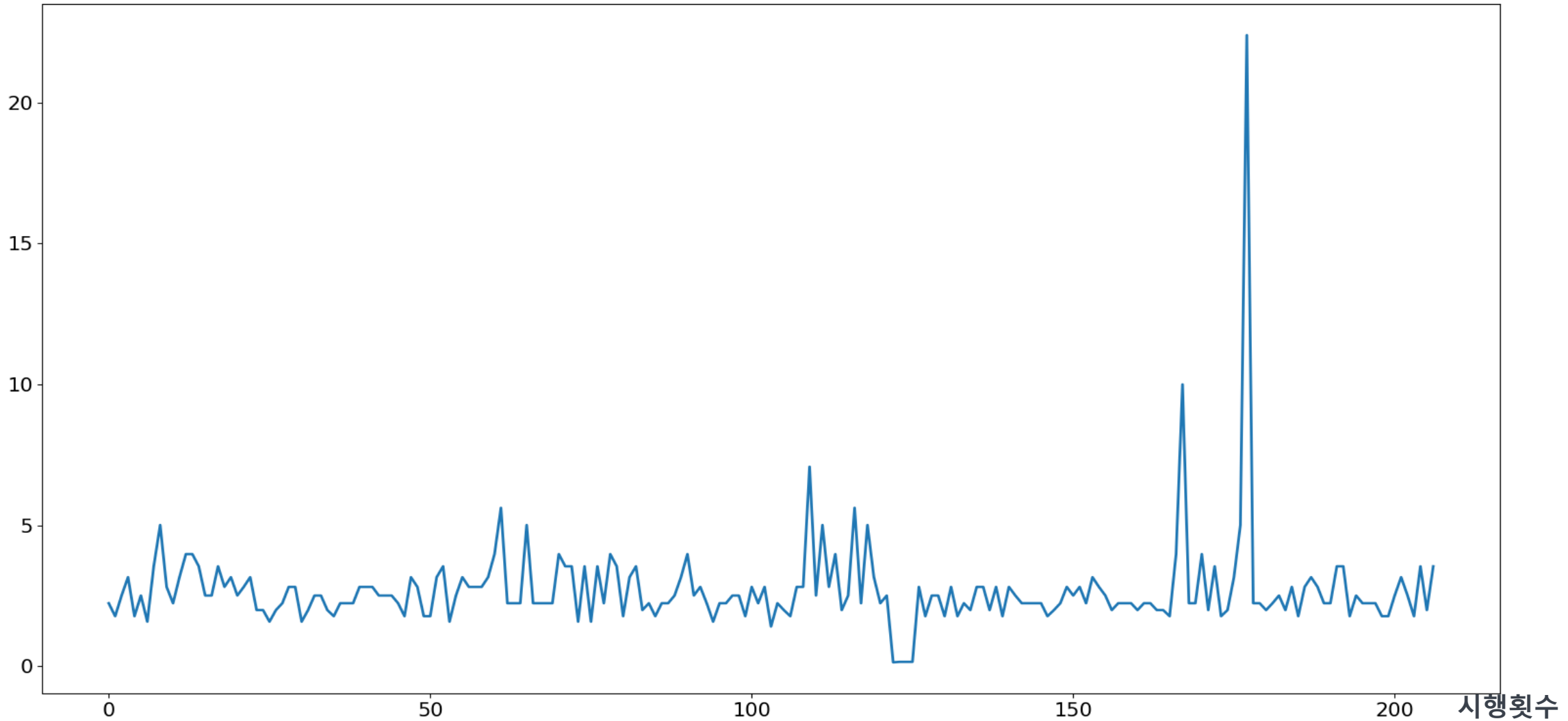


2. 실제로 측정한 RSSI (0.8m)



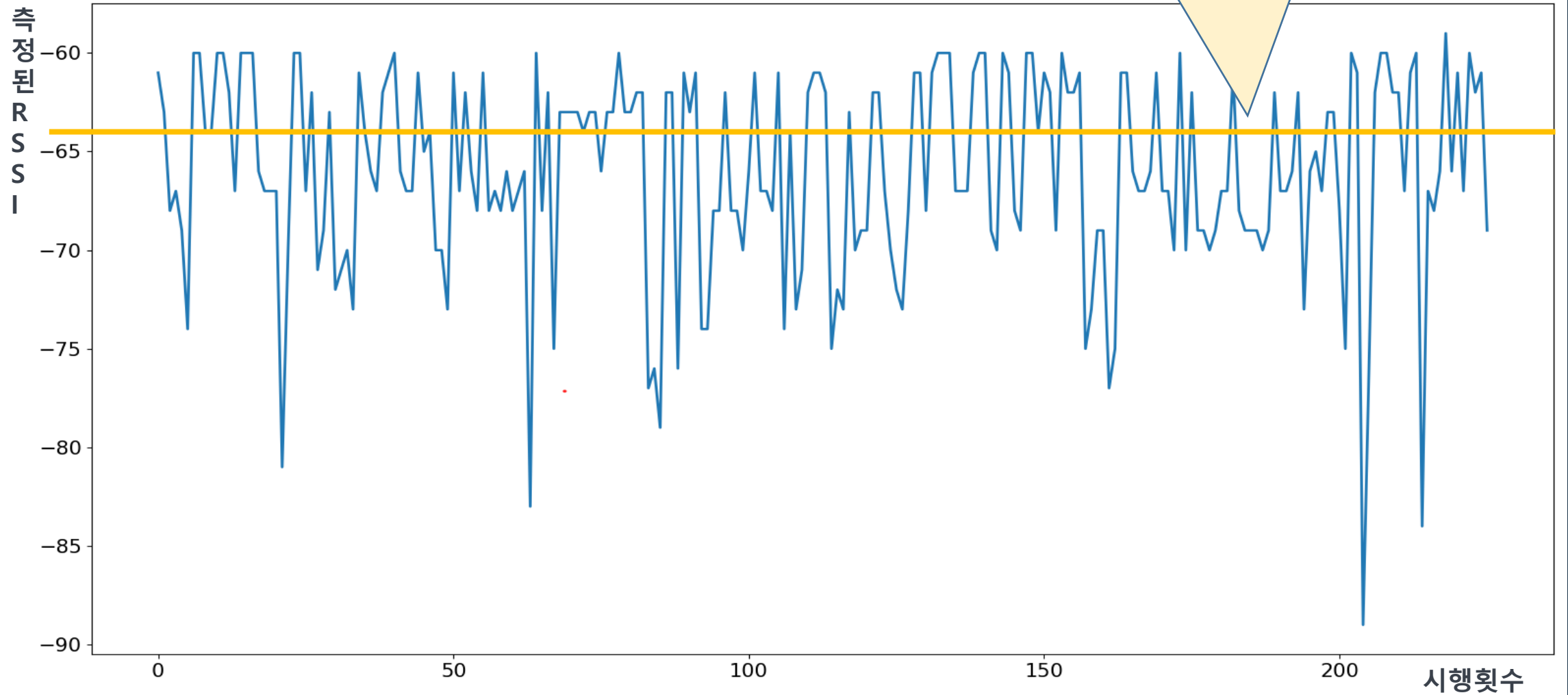
3. 측정된 RSSI로 계산한 거리(0.8m)

계산한
거리
(m)

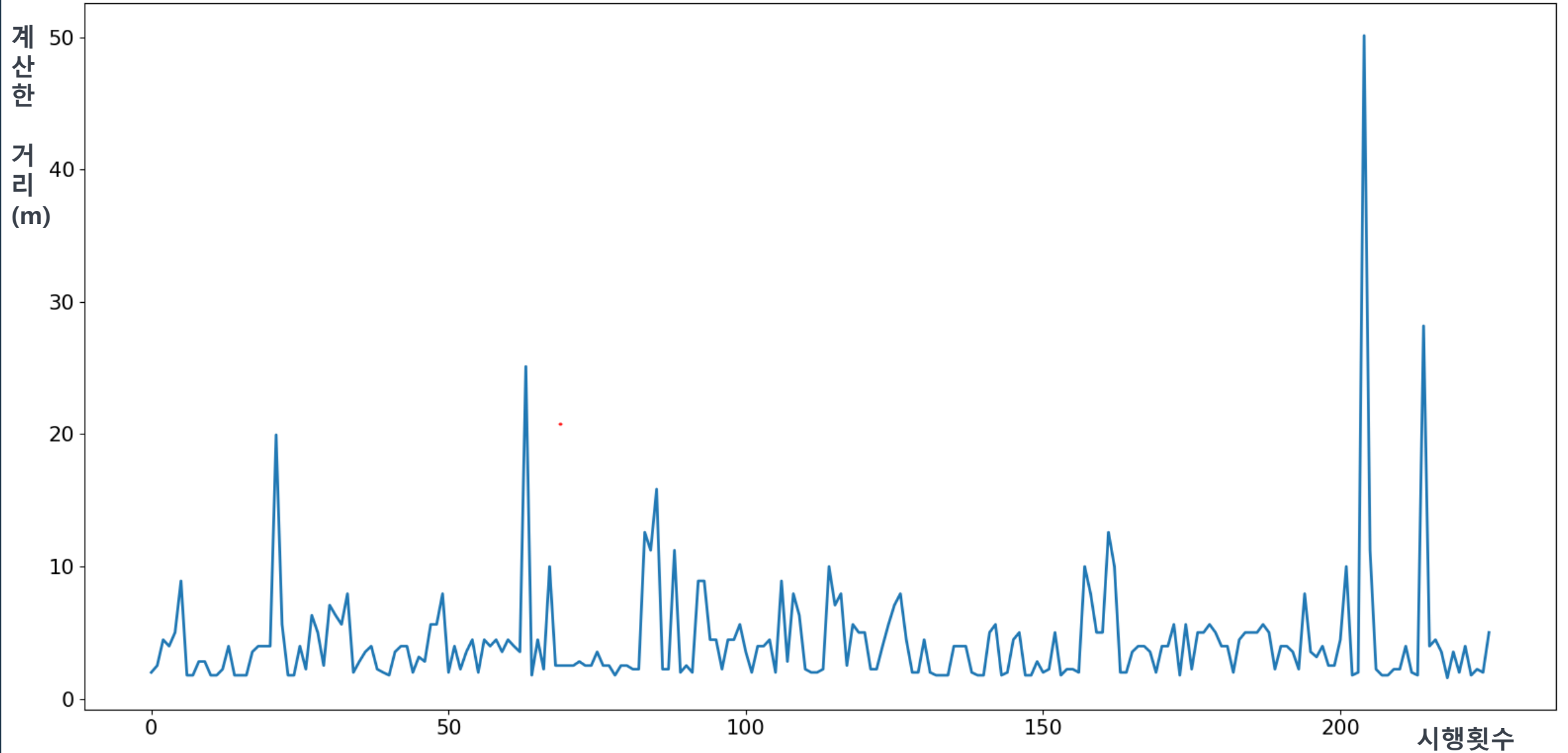


2. 실제로 측정한 RSSI (1.2m)

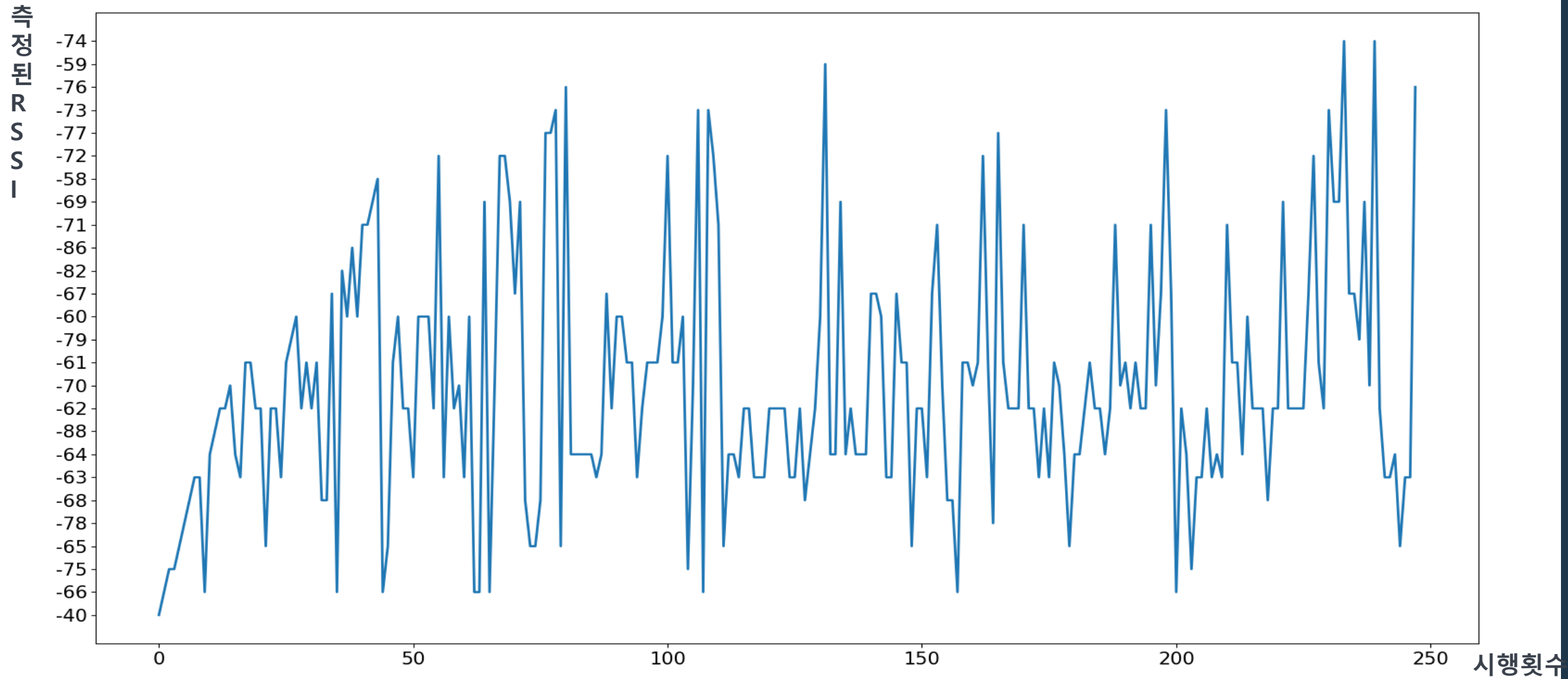
1.2M일 때, 이론 상 RSSI



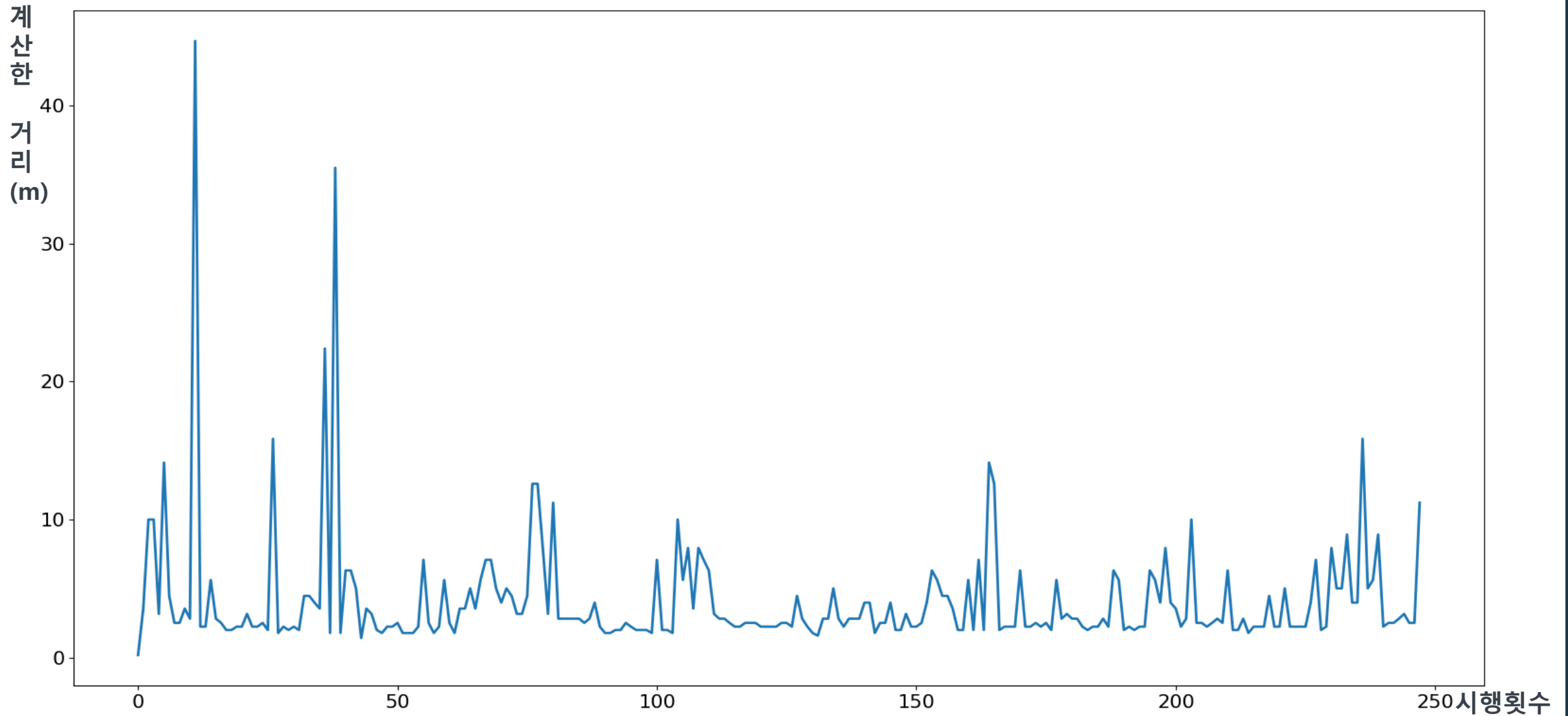
3. 측정된 RSSI로 계산한 거리(1.2m)



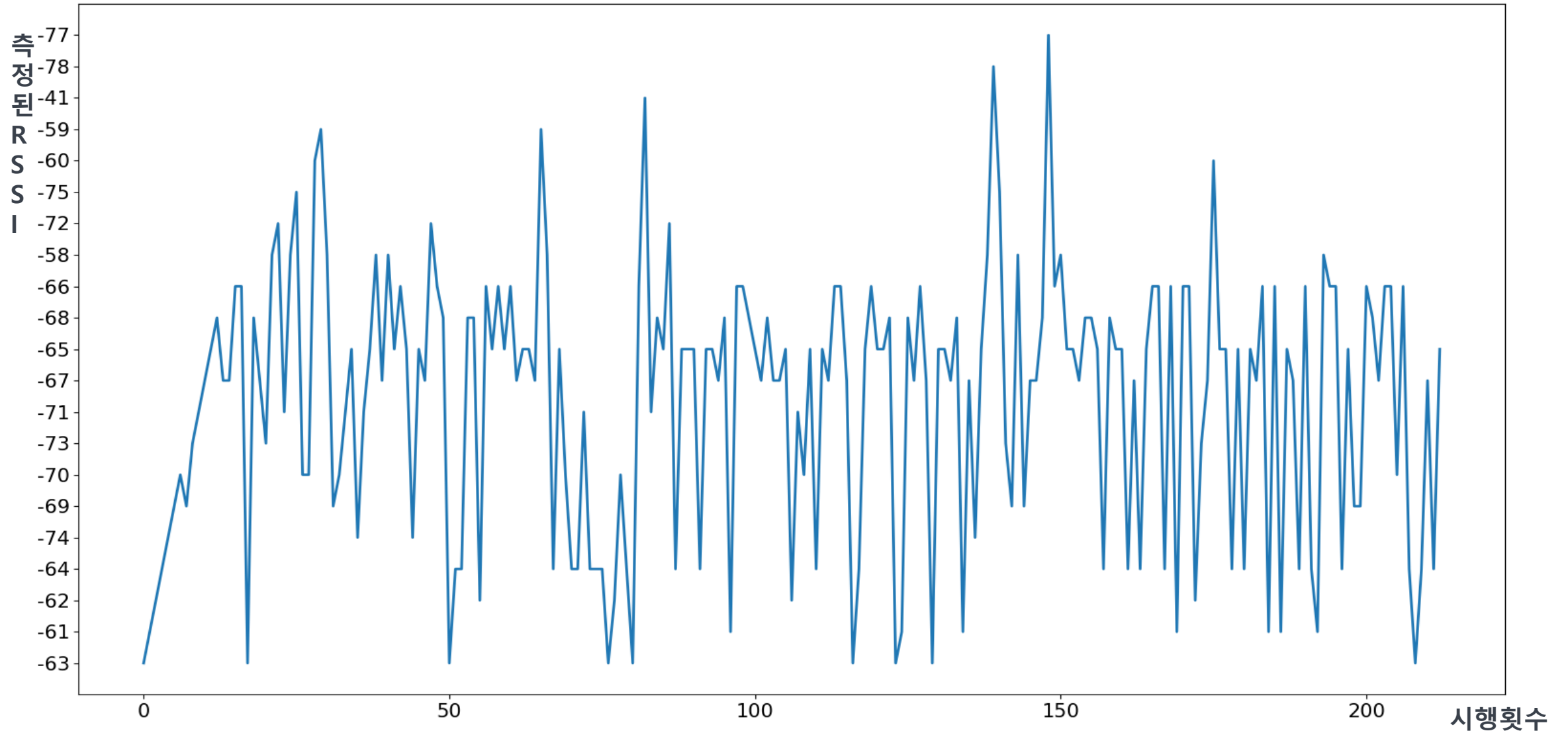
2. 실제로 측정한 RSSI (1.6m)



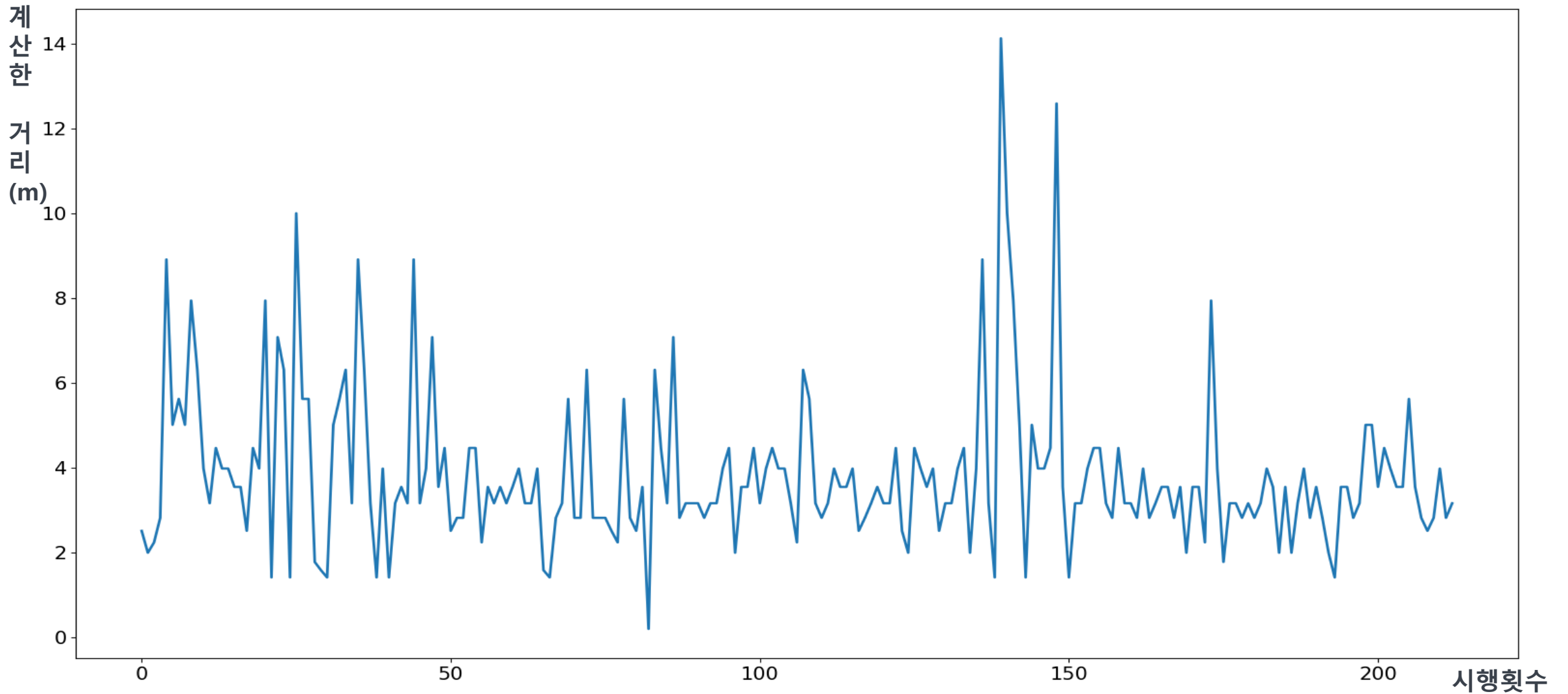
3. 측정된 RSSI로 계산한 거리(1.6m)



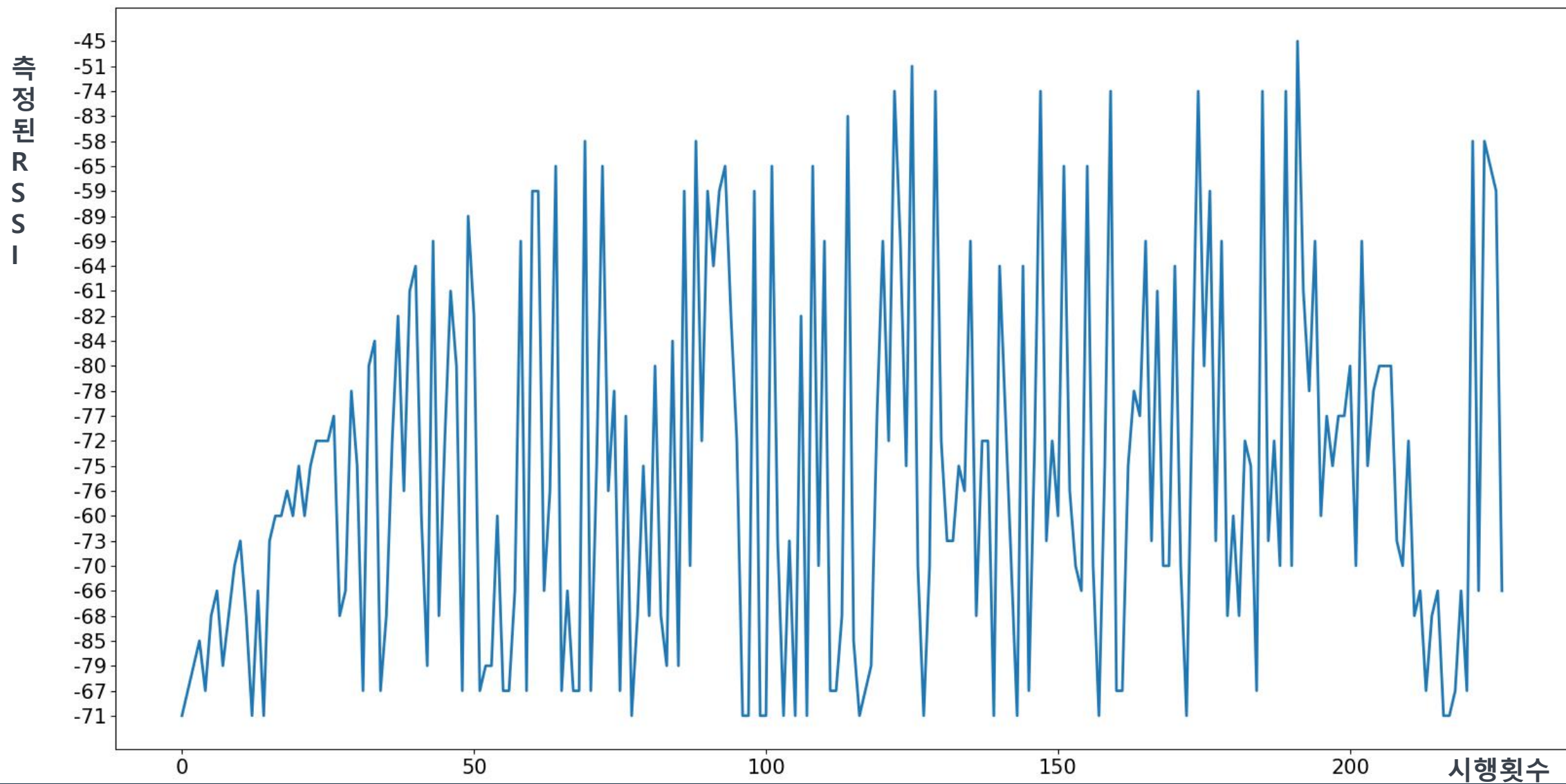
2. 실제로 측정한 RSSI (2.0m)



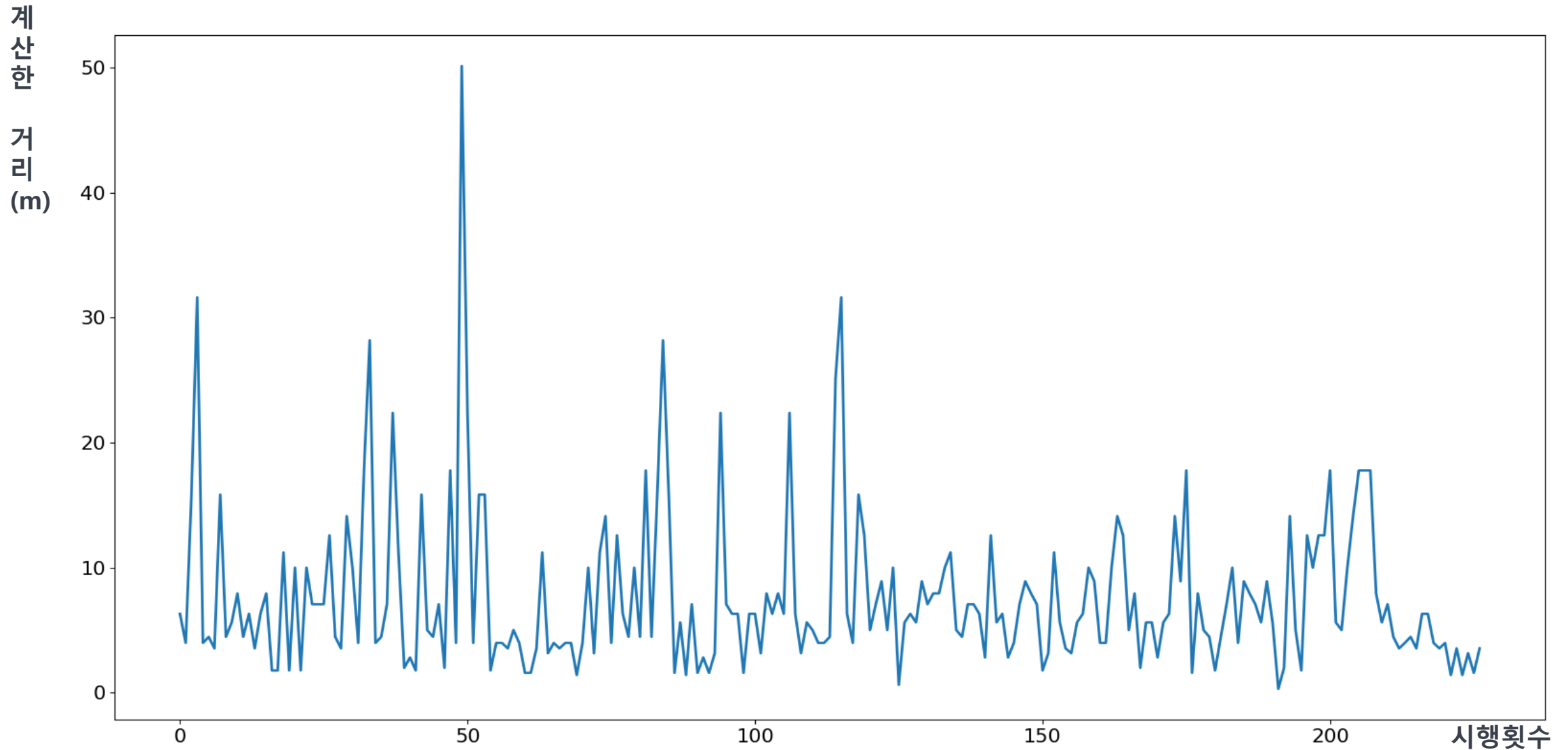
3. 측정된 RSSI로 계산한 거리(2.0m)



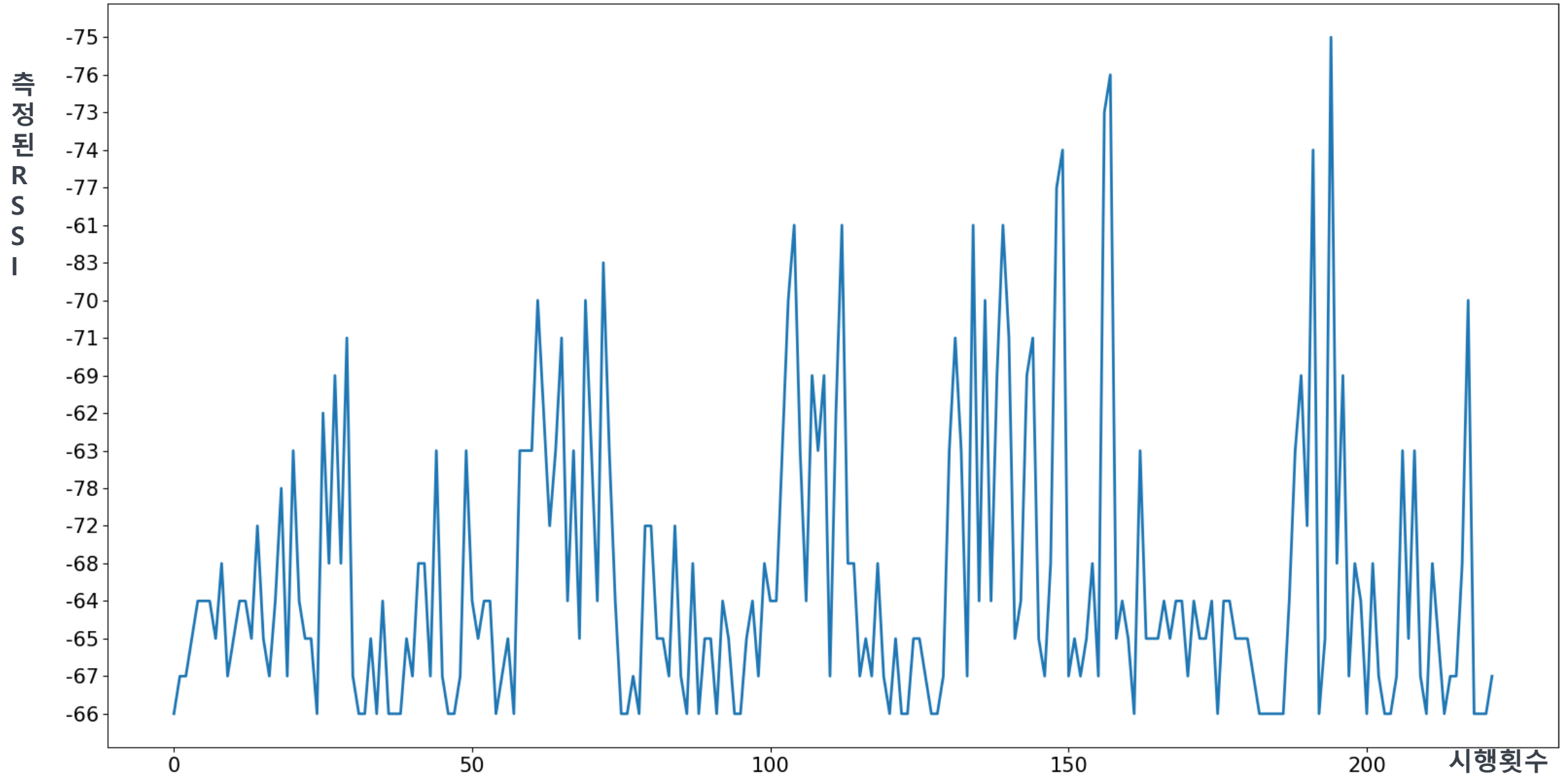
2. 실제로 측정한 RSSI (2.4m)



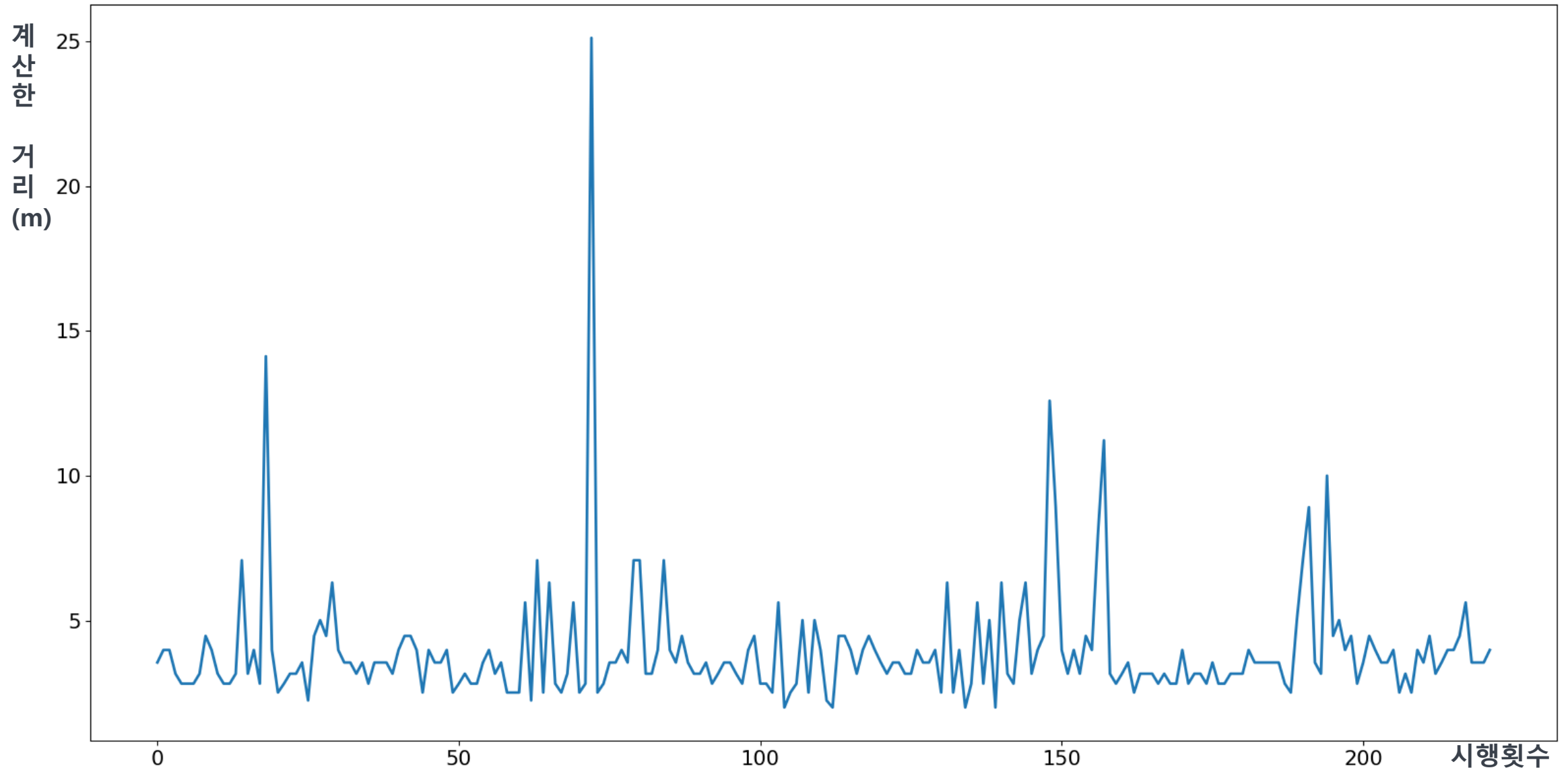
3. 측정된 RSSI로 계산한 거리(2.4m)



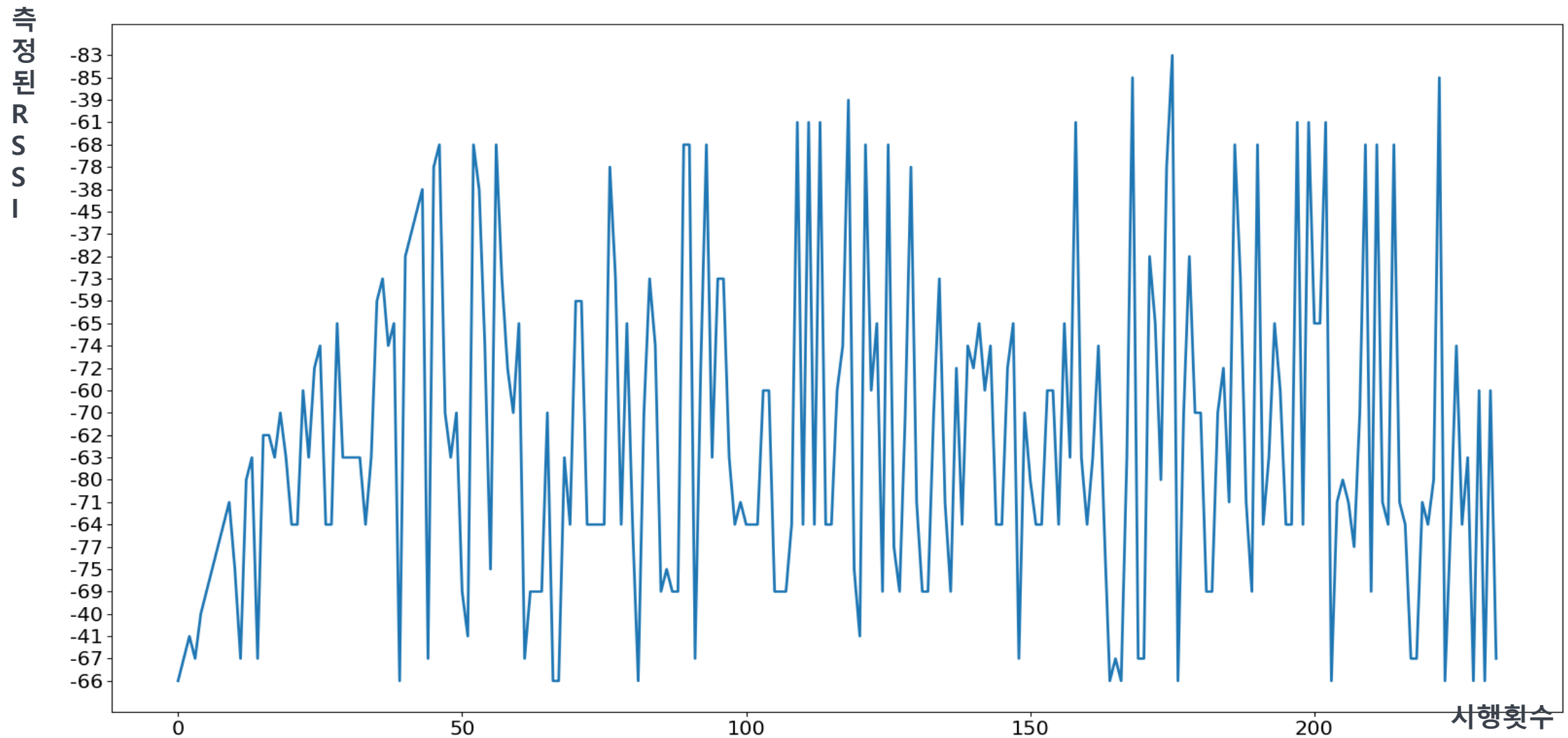
2. 실제로 측정한 RSSI (2.8m)



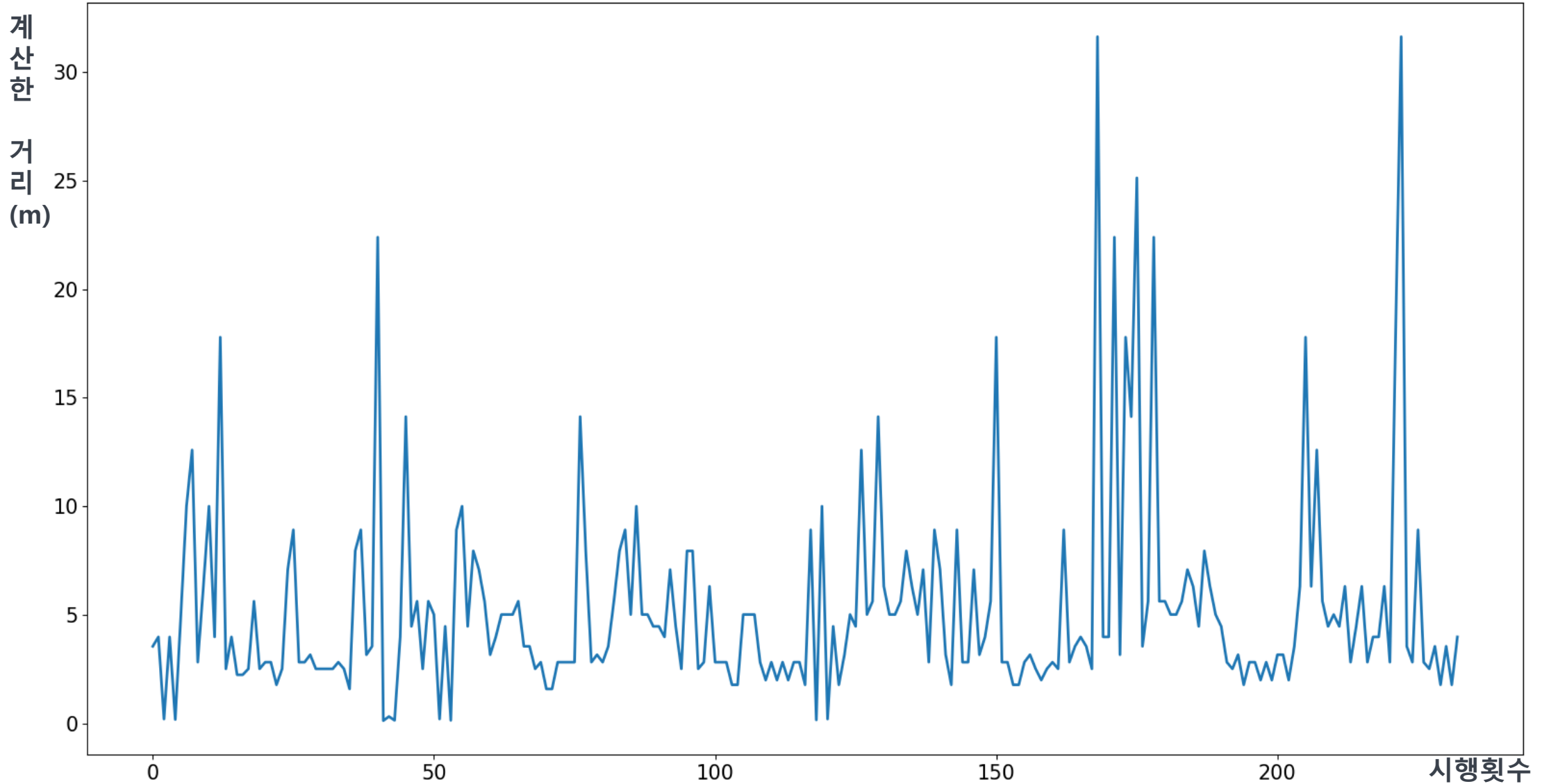
3. 측정된 RSSI로 계산한 거리(2.8m)



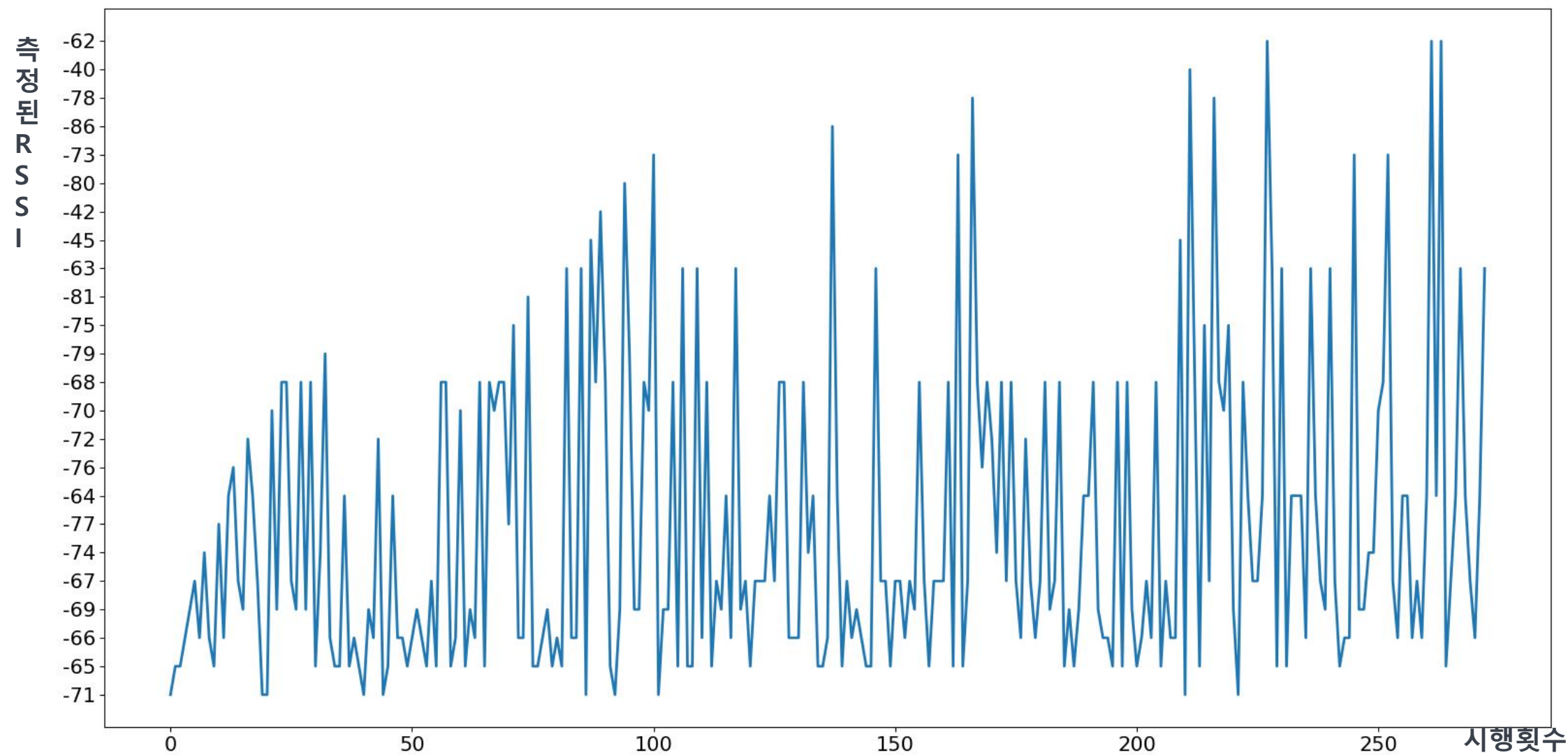
2. 실제로 측정한 RSSI (3.2m)



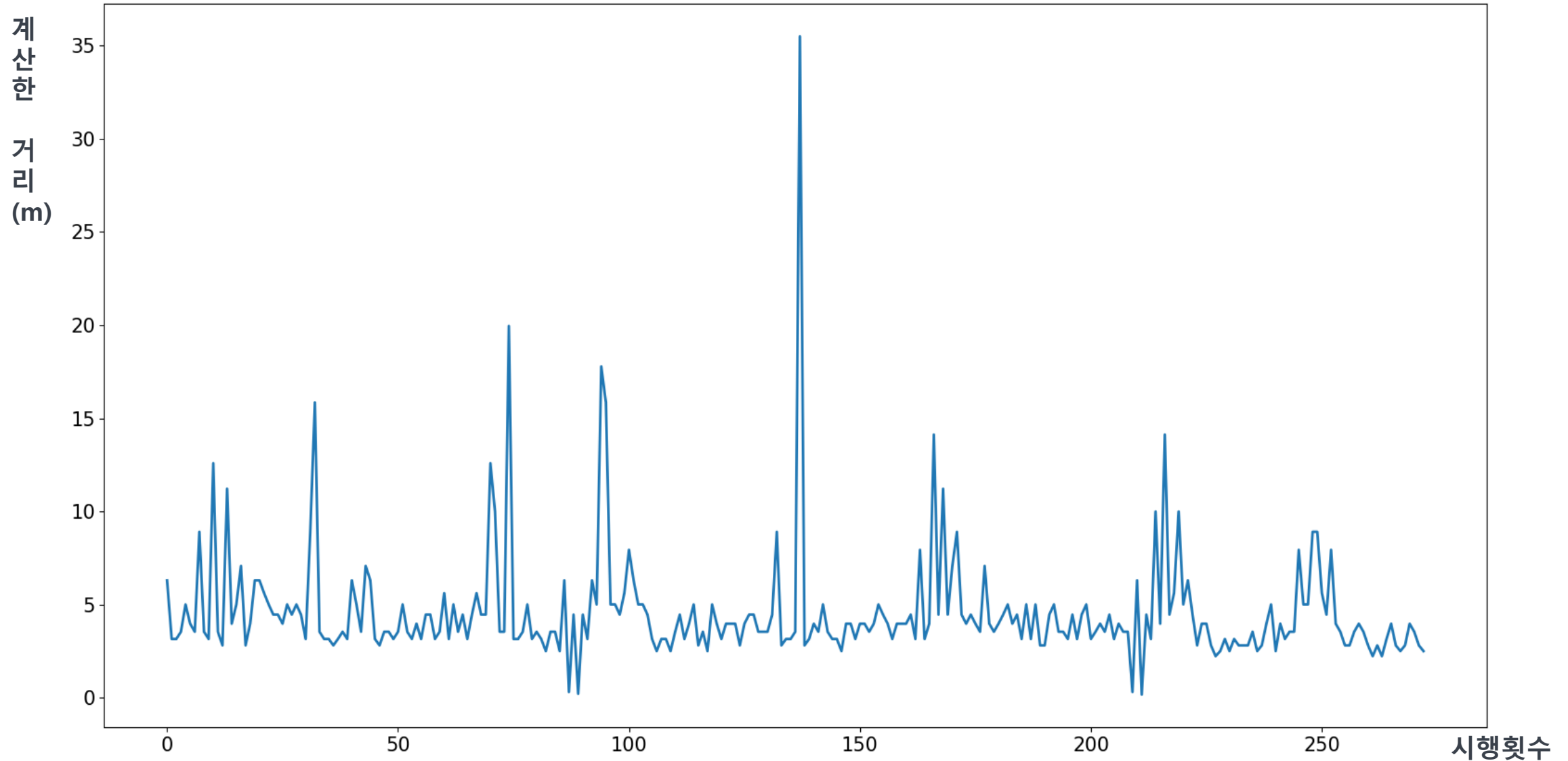
3. 측정된 RSSI로 계산한 거리(3.2m)



2. 실제로 측정한 RSSI (3.6m)



3. 측정된 RSSI로 계산한 거리(3.6m)



4. 측정한 RSSI와 거리를 측정할 때 사용한 코드

```
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)
```

```
def f(B1, x):
    return ((1 / np.sqrt(2 * np.pi * (B1.sigma_RSS ** 2))) * np.exp(-(x - (B1.mu_RSS)) ** 2 / (2 * B1.sigma_RSS ** 2)))

#x좌표, y좌표, N=2(아무런 제약이 없다고 가정), M_Power
B1 = Beacon(10, 0, 2, -55)

#np.linspace(a,b,c) a부터 시작해서 b까지 총 c등분 한다.
temp = "-61 -63 -68 -67 -69 -74 -60 -60 -64 -64 -60 -60 -62 -67 -60 -60 -66 -67 -67 -67 -81 -70 -60 -60 -67 -62 -71"
temp = temp.split("\t")
x = [(int(i)) for i in temp]
a = [i for i in range(len(x))]
for i in x:
    B1.add_RSS(i)

y = [B1.RSSI_to_distance(i) for i in B1.dataset]
B2 = Beacon(0, 0, 2, -55)
print(B2.Dist_to_RSSI(20))
print(B2.RSSI_to_distance(-81))

plt.style.use('default')
plt.rcParams['figure.figsize'] = (20, 20)
plt.rcParams['font.size'] = 15
plt.rcParams['lines.linewidth'] = 2
# plt.plot(a, temp)
plt.plot(a, y)
plt.show()
```

RSSI를 확인할 때는 1번,
측정된 거리를 확인할 때는 2번을
각각 사용함.