

UWB/BLE 기반 3D 실내측위 및 동기화



201524498 안 현 기

201624458 김 춘 수

201924553 이 현 진

지도교수 김 원 석

목 차

1. 서론.....	1
1.1. 연구 배경.....	1
1.2. 기존 문제점.....	1
1.3. 연구 목표.....	1
2. 연구 배경.....	2
2.1. BLE.....	2
2.2. RSSI.....	2
2.3. 사용 모듈.....	3
3. 연구 내용.....	3
3.1. 시스템 구성도.....	3
3.2. BLE 통신 프로토콜.....	4
3.3. 오차 제어 필터.....	5
3.4. 사변측량.....	7
4. 연구 결과 분석 및 평가.....	8
5. 결론 및 향후 연구 방향.....	14
6. 구성원별 역할 및 개발 일정.....	15
7. 참고 문헌.....	16

1. 서론

1.1. 연구 배경

우리는 4차 산업혁명의 시대에 살고 있다. 4차 산업혁명이란 인공지능, 블록체인, 빅데이터, 로봇공학, 양자암호, 무인 운송 수단, 3D 인쇄 등 정보통신 기술의 융합으로 대표되는 새로운 산업 혁명이다. 또한 4차 산업혁명에서 새롭게 주목을 받는 기술 영역이 있다. 그것은 바로 “디지털 트윈(Digital Twin)”이다.

디지털 트윈이란 현실 세계의 쌍둥이를 컴퓨터 소프트웨어로 표현하는 것을 말한다. 디지털 트윈은 생산과 건설 분야를 비롯해 교육, R&D, 설계, 마케팅 등 다양한 분야에서 사용되고 있다. 디지털 트윈을 이용하면 시뮬레이션을 통해 제품 출시 전 테스트와 검증을 할 수 있고, 다양한 시나리오에 대한 결과를 예측할 수도 있다.

실재하는 공간을 디지털 트윈으로 표현하기 위해서는 3D 실내측위 및 동기화 기술이 필요하다. 우리는 UWB/BLE 기반 3D 실내측위 및 동기화를 졸업 과제 주제로 선택하고 연구하기로 하였다.

1.2. 기존 문제점

기존 상용화된 대부분의 실내측위 시스템은 2D 공간상에서 동작한다. 실제 공간을 2D에서 구현하는 것은 3D 실내측위에 비해 현실성이 떨어진다. 측위에 오차가 발생하기 때문에 이를 보정하지 않고 바로 디지털 트윈상에 동기화 하는 것은 정확성이 떨어진다. 또한, 공간좌표를 계산하여 웹 서버로 업로드를 하는 데 지연 시간이 발생하여 실시간성이 보장되지 않는다.

1.3. 연구 목표

1. 실내측위 기법으로 대두되고 있는 UWB 기술과 BLE 기술 중 BLE 기술을 사용하여 저비용의 실시간 실내측위 시스템(Real-Time Location System)을 구현하고 공간좌표를 웹 서버에 업로드하여 3D 공간에 동기화한다.

-UWB 기술 대신에 BLE 기술을 선택한 이유는 다음과 같다. UWB 기술이 BLE 기술에 비해서 정확도도 높고, 초당 약 100회의 패킷 전송으로 지연 없이 위치 정보를 획득할 수 있는 장점이 있다. 하지만 장애물이 있는 실내에서 측위의 안정성이 떨어지며, 시간 동기화를 해야하는 단점이 있다. BLE 기술은 UWB 기술에 비해서는 정확도가 낮기는 하지만, 오차를 보완할 수 있는 방법이 많이 고안되고 있고, 장애물이 있는 실내에서도

오차가 많이 발생하지 않는다. 비콘이 저전력으로 작동하기 때문에 오랜 시간 측위를 진행할 수 있으며, 시간 동기화가 필요하지 않아서 실내측위 시스템을 구현하는데 더 적합하다.

2. 측위 오차를 줄이기 위한 보정 방법과 빠른 좌표 계산에 사용될 측량 알고리즘을 연구 및 개발한다.

2. 연구 배경

2.1. BLE

BLE는 Bluetooth Low Energy의 준말이며, BLE Beacon(비콘)의 브로드캐스트 신호(정보) 수신으로 위치를 확인하는 방식이다. 과거에는 정확도가 3 ~ 8m 수준으로 불안정하여 완벽한 정밀도가 요구되지 않는 실내 위치확인 서비스와 인원, 자산의 입출/반출 관리 및 Room 레벨 기반의 실시간 위치 확인(RTLS)에 사용되었다. Bluetooth 5.x 버전에서 정확도가 향상되었다. 거리 측정은 신호 세기 RSS(Received Signal Strength)를 이용한다. 출력에 따라 최대 40 ~ 400m 내외의 통신이 가능하고 주파수 대역은 2400 ~ 2483.5MHz를 사용한다. 비콘은 코인 셀 배터리로 평균적으로 1년 이상 사용 가능하다. BLE는 가속도, 온도 등의 센서 일체/연동형으로 모니터링 대상의 상태 모니터링 서비스로 확장되고 있다.

BLE는 서비스 구성 및 확장에 유연하며 비용 대비 효율적인 시스템을 구성할 수 있는 장점이 있다. 반면, 건물과 같은 장애물이 있을 때 신호가 감쇄되는 현상이 있으며, 레이아웃 변경과 무선 간섭에 따른 불안정성이 있다는 단점이 있다.

2.2. RSSI

RSSI는 Received Signal Strength Indicator의 약자로 수신기에서 수신되는 전파의 세기가 얼마인지를 수치(dB)로 나타내는 것이다. RSSI 값은 아래의 수식(1)으로 계산된다. 기본적으로 음수로 나타내며 0에 가까울수록 신호의 세기가 강하다.

$$RSSI = -10 \times \log D + TXPower \dots (1)$$

여기서 D 는 거리를, $TXPower$ 는 송신 신호의 세기를 뜻한다. 송신기에서 Broadcast한 신호를 수신기가 수신하여 거리를 측정하고, 측정한 각 송신기까지의 거리를 삼변측량법이나 기타 다른 방법을 이용해 좌표를 계산할 수 있다.

2.3. 사용 모듈

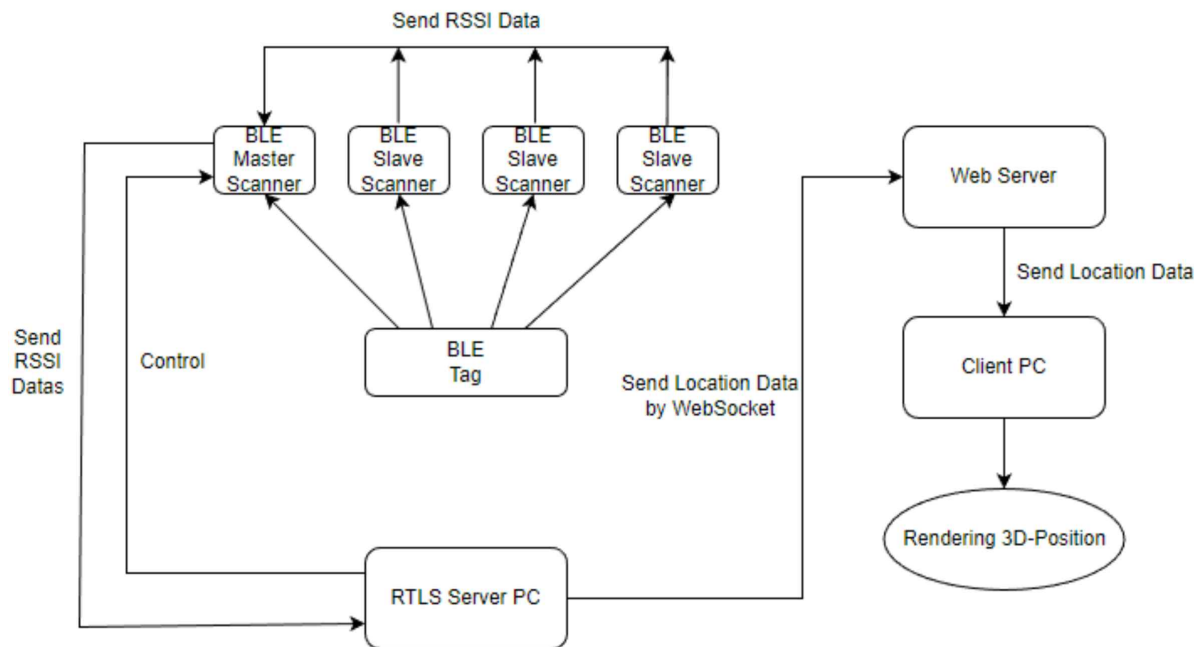
BLE 모듈로써 Adafruit ItsyBitsy nRF52840을 사용했다. 기존에는 BLE 통신칩이 내장된 Arduino Nano 33 BLE 모듈을 사용하기로 하였으나, 실제로 모듈을 구매한 후 통신 테스트를 해본 결과 3m 이상의 거리에서는 모듈 간 통신이 잘 되지 않았다. 오차나 지연시간 등과 관계없이 해당 모듈을 RTLS에 사용하기에는 무리가 있다고 판단하였다.

Adafruit ItsyBitsy nRF52840 모듈은 Arduino Nano 33 BLE과 마찬가지로 추가적인 센서 부착 없이 BLE 5.x버전 통신과 공식 라이브러리를 지원한다. 또한 Blink 신호의 송신과 수신이 동시에 가능하며 복합적인 시스템 구성이 가능하고, Arduino 라이브러리와 상당수 호환되므로 접근성이 높아 개발이 편하다.

3. 연구 내용

3.1. 시스템 구성도

BLE를 이용한 실시간 실내측위 시스템을 아래의 [그림 1]과 같이 구성했다. BLE Tag 모듈이 일정한 간격으로 Blink 신호를 방송하면 4대의 Scanner 모듈이 이를 수신하고 RSSI를 측정한다. 측정한 RSSI 값을 기반으로 3D 좌표를 계산하기 위해 RSSI 값을 Server PC로 전송해야 하는데 각각의 Scanner 모듈을 PC에 유선으로 연결하여 값을 전송하기에는 현실적으로 무리가 있다. 따라서 Scanner 모듈 중 하나를 Master 모듈로, 나머지를 Slaver 모듈로 임의로 정한 후 Master 모듈이 게이트웨이 역할을 수행하도록 구성했다. Master-Slave간 통신은 마찬가지로 BLE 통신을 이용한다.



[그림 1] 시스템 구성도

Master 모듈은 각 Scanner 모듈에서 측정된 RSSI 값을 Server PC와 유선으로 연결한 후 직렬통신을 통해 JSON 형식으로 전송한다. Server PC는 전송 받은 RSSI 값을 바탕으로 사변측량(Quadrilateration) 알고리즘으로 3D 좌표를 계산한다. 실시간으로 계산된 좌표를 Web Server에 업로드하면 Client PC가 원격으로 값을 참조하고, 3D 공간좌표를 렌더링하도록 구성했다.

기존에는 AWS서버에 3D 좌표를 업로드한 후 Client PC가 서버에 접속하여 렌더링을 할 계획이었다. 하지만 BLE Scanner가 수신한 정보를 처리하기 위해서는 가까운 거리에서 동작중인 Server PC가 필요하고, AWS 서버를 거쳐서 다시 Client PC로 데이터를 전송하는 것보다 Server PC가 바로 웹 서버를 호스팅하여 데이터를 업로드 하는 것이 불필요한 통신 및 지연시간을 줄이는 것이라 판단하여 설계를 수정했다.

3.2. BLE 통신 프로토콜

각 모듈(Tag, Slave scanner, Master scanner)은 모두 같은 BLE 통신을 사용하기 때문에 송신 모듈을 구별하기 위한 데이터를 패킷에 포함하여 전송해야 한다. 또한, Slave scanner에서 측정된 Tag 신호의 RSSI값을 Master scanner로 전송하기 때문에 패킷에 해당 RSSI 값도 포함해야 한다. 따라서 아래의 [그림 2]와 같이 11 bytes로 구성된 프로토콜을 설계하여 모든 패킷을 처리했다.

buf [0:1]	module name [2:5]	buf [6]	rsssi [7:10]
-----------	-------------------	---------	--------------

[그림 2] BLE 통신 프로토콜

본 과제에서 사용하는 모듈은 BLE 통신을 위해 Nordic社의 nrf52840 칩을 사용하는데, 해당 칩을 이용할 때 BLE 통신 패킷의 가장 앞 2 bytes는 일정한 값으로 설정되어 있고 버퍼의 역할을 겸한다. 그 이후의 Bytes는 Custom API를 사용하여 설정이 가능한데 가변 길이의 모듈 이름을 "TAGW0"과 같이 4 bytes 문자열로 고정하여 전송하고, 음수의 값을 가지는 RSSI 값 또한 "-67Wn"과 같이 4 bytes 문자열로 전송하도록 설계했다.

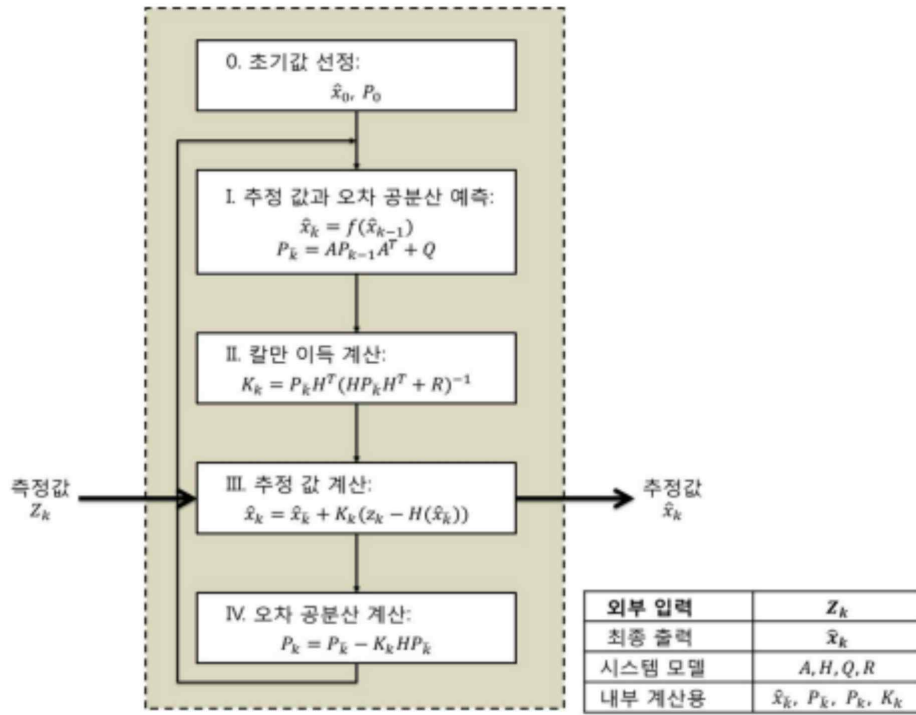
Master scanner가 Tag에서 보낸 패킷을 수신할 때는 신호 자체의 RSSI 값을 측정하고, Slave scanner에서 보낸 패킷을 수신할 때에는 패킷에 함께 담겨온 RSSI 값을 사용한다.

3.3. 오차 제어 필터

측정된 RSSI 값은 반사파, 장애물 및 다중경로 등 여러 이유로 오차가 크게 발생한다. 따라서 3D 좌표를 계산하기 전에 이를 보정할 필요가 있다. RSSI 값을 보정하기에는 확장 칼만 필터(Extended Kalman Filter)가 적절하다고 판단하여 해당 필터를 적용했다. 그 이유는 아래와 같다.

1. 칼만 필터는 앞서 입력된 값을 모두 기억하여 다음 값을 예측하는 것이 아닌, 직전의 추정값과 공분산 등 몇 개의 변수만 기억하여 연산하기 때문에 연산이 빠르다.
2. 입력값에 대한 통계적 추정을 통해 가중치가 유동적으로 조절되므로 불규칙적으로 변하는 물체의 위치를 보정하는 데에 적절하다.
3. 기존의 칼만 필터는 선형 시스템에서만 동작하므로 비선형 시스템에서도 동작하도록 개선된 확장 칼만 필터를 사용한다.

확장 칼만 필터는 측정값을 입력으로 받아 내부에서 처리한 다음 값을 예측하여 출력한다. [그림 3]에 상세한 과정이 묘사되어 있는데, 크게 예측 과정(I)과 추정 과정(II, III, IV)의 반복으로 구분 지어 볼 수 있다[1].



[그림 3] 확장 칼만 필터 알고리즘[1]

과제를 수행할 때는 초기값으로 -50, 공분산으로 3을 임의로 사용했다.

RSSI 값을 측정할 때에 그 이전 값에 비해 10 이상씩 변화하는 Outlier가 발생하는데 이를 확장 칼만 필터를 사용하기 전에 따로 제거하였다. 확장 칼만 필터 사용 전에 Outlier를 제거해야 필터 내의 변수들에 Outlier에 의한 영향이 미치지 않는다.

BLE Tag는 20ms에 한번씩 방송 신호를 보내는데, 이전 80번에 대한 정보를 저장하고 아래의 수식(2)으로 기댓값을 계산한 후 현재 받은 신호와 비교하여 Outlier인지 아닌지를 판단한다. 기대값과 측정된 RSSI 값이 3 이상 차이나면 Outlier로 판단하여 측위를 위한 계산에 포함시키지 않았다. RSSI 값이 3만큼 변화했을 때 이를 이동거리로 변환하면 약 1.412m 인데 현실의 물체가 20ms 동안 이동하기에는 너무나 긴 거리이다. 이론상으로는 Outlier의 판단 기준을 RSSI 값 차이가 2 이상 혹은 1 이상으로 좁혀도 되지만, BLE 센서의 RSSI 값 측정이 정확하지 않아 대부분 Outlier로 판단되어 3 으로 설정했다.

$$Expected_Value = (0.2) \times \sum_{i=0}^{i=80} RSSI[i] \times 0.8^i \dots (2)$$

3.4. 사변측량

사변측량 알고리즘은 삼변측량 알고리즘을 확장한 것으로, 한 평면에 있지 않은 3D 공간상의 점 4개로부터 측위하고자 하는 지점까지의 거리를 각각 알고 있으면 좌표를 구할 수 있음을 이용한 알고리즘이다. 하지만 실제 거리와 RSSI 값을 통해 계산된 거리는 차이가 있기 마련인데, 이 오차 때문에 가능한 좌표가 여러 개로 계산되거나 혹은 구할 수 없는 경우가 존재한다. 따라서 본 과제에서는 사변측량 알고리즘을 변형하여 적용했다.

3D 공간상의 한 점에서 측위하고자 하는 점까지의 거리를 알고 있다면, 그 점의 좌표가 될 가능성이 있는 위치를 구의 방정식으로 나타낼 수 있다. 또 다른 한 점에서 측위하고자 하는 점까지의 거리를 구하여 마찬가지로 구의 방정식으로 나타낸 후, 두 구의 방정식을 빼면 평면의 방정식이 나타난다. Scanner 모듈이 4개이므로 이 평면의 방정식은 최대 6개까지 구할 수 있는데, Scanner 모듈을 배치할 때 서로 다른 평면상에 있도록 배치하면 이 평면의 방정식들 중 3개를 임의로 골라 연립하면 단 하나의 공통근을 가짐이 증명되어 있으므로 이를 이용하여 Tag 모듈의 위치를 특정한다. 따라서 모든 평면 쌍에 대하여 해당 알고리즘을 적용한 후 결과로 나온 좌표들의 평균을 사용하여 특정 Scanner에서 가중치가 생기지 않도록 한다.

또한, RSSI 값과 마찬가지로 물체의 실제 위치는 짧은 시간내에 급격하게 변화할 수 없으므로 물체의 좌표 값에도 수식(2)를 적용하여 기대값을 계산한 후 좌표로 사용한다.

아래의 JavaScript 코드는 평면 쌍 하나에 대하여 좌표를 계산하는 과정을 구현한 것이다.

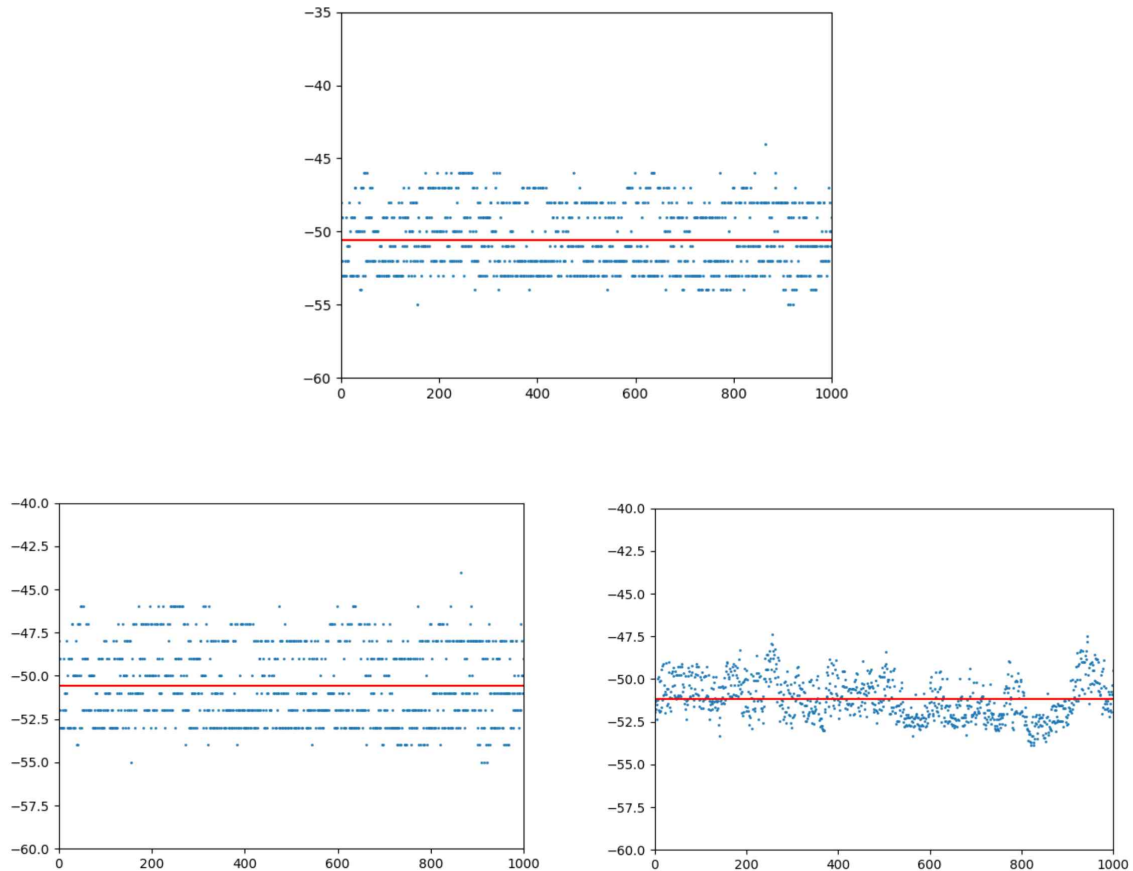
```
// @param: coord, distance
// coord => [[x1, y1, z1], [x2, y2, z2], [x3, y3, z3], [x4, y4, z4]]
// distances => [d1, d2, d3, d4]
function calculate(coord, distances) {
  1 let mA= new Array();
  1 let mb= new Array();
  f for(let i = 0; i < 3; i++) {
    mA mA[i] = new Array();
    1e let x0= coord[i][0];
    1e let x1= coord[i + 1][0];
    1e let y0= coord[i][1];
    1e let y1= coord[i + 1][1];
    1e let z0= coord[i][2];
    1e let z1= coord[i + 1][2];
    mA mA[i][0] = 2 * (x1 - x0);
    mA mA[i][1] = 2 * (y1 - y0);
    mA mA[i][2] = 2 * (z1 - z0);
```

```

    mb[i] = distances[i] * distances[i] - distances[i + 1] * distances[i + 1];
    mb[i] += (x1 * x1 + y1 * y1 + z1 * z1) - (x0 * x0 + y0 * y0 + z0 * z0);
  }
  let invertedA= math.inv(mA);
  return math.multiply(invertedA, mb);
}

```

4. 연구 결과 분석 및 평가



[그림 4] 1m 거리에서 측정한 RSSI 값(상), Outlier 제거(좌), 확장 칼만 필터 적용(우)

측정된 RSSI 값을 이용하여 두 모듈 간 거리(단위: m)를 역으로 계산하는 수식은 아래와 같다[2][3].

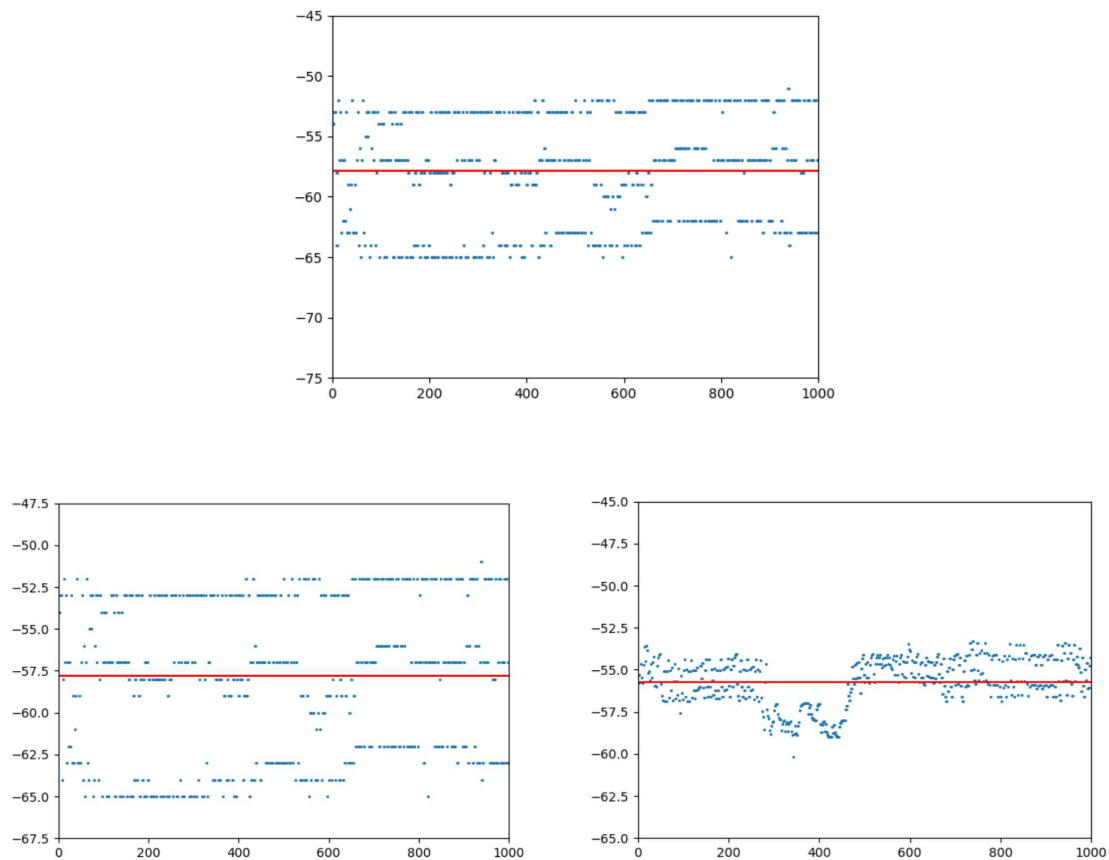
$$D = 10^{(A-RSSI)/10n} \dots (3)$$

위의 수식에서 A 는 기준거리(1 m)에서 태그로부터 수신된 신호 세기를, n 은 신호 감쇠와 관련된 상수를 나타낸다. n 은 자유 공간에서 약 2의 값을 가지며 장애물이 많을수록 커진다. [그림 4]는 1m 거리에서 측정한 RSSI 값(상), Outlier를 제거한 값(좌), 확장 칼만

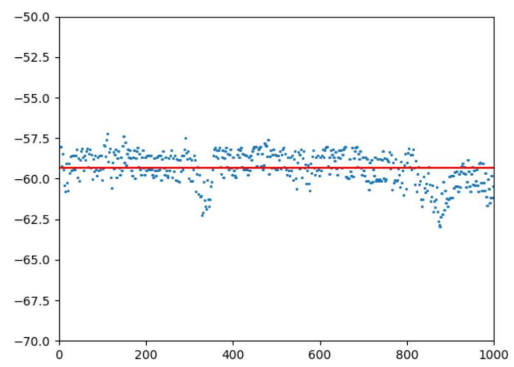
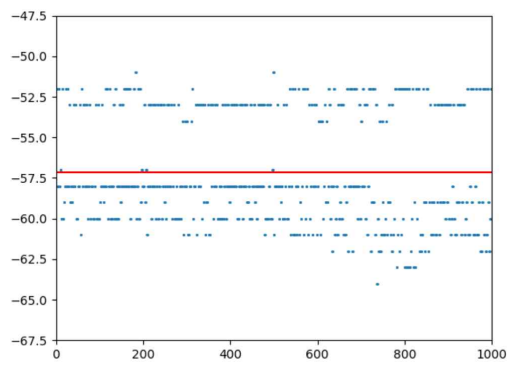
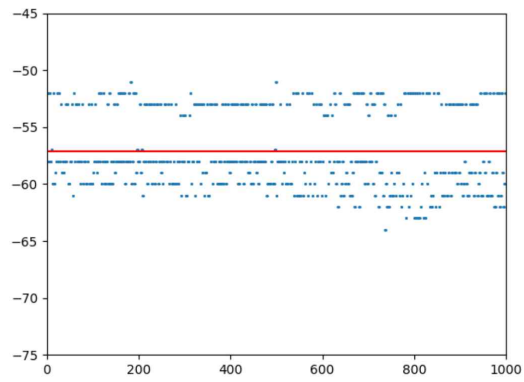
필터를 적용하여 오차를 제어한 값(우) 표본 1000개를 나타낸 그림이다. 푸른 점은 RSSI 값을, 붉은 실선은 평균값을 의미한다. 필터링된 RSSI 값들의 평균값으로 A 를 설정하여 과제를 수행했다.

$$A = -49.818248 \dots (4)$$

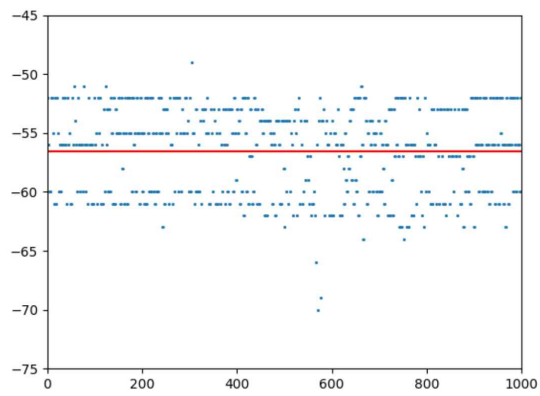
[그림 5], [그림 6], [그림 7]은 각각 2, 3, 4m 거리에서 측정한 RSSI 값(상), Outlier를 제거한 값(좌), 확장 칼만 필터를 적용하여 오차를 제어한 값(우) 표본 1000개를 나타낸 그림이다. 각 그림별로 푸른색 점은 1000개의 표본을 의미하고, 붉은 색 실선은 그 평균이다. 필터링된 값의 평균은 각각 -55.729427m, -59.353124m, -61.746778으로 나타났다.

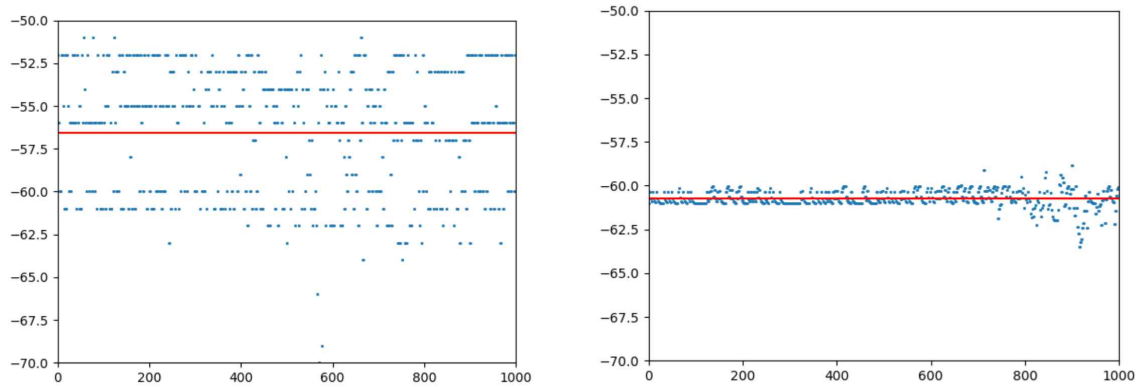


[그림 5] 2m 거리에서 측정한 RSSI 값(상), Outlier 제거(좌), 확장 칼만 필터 적용(우)



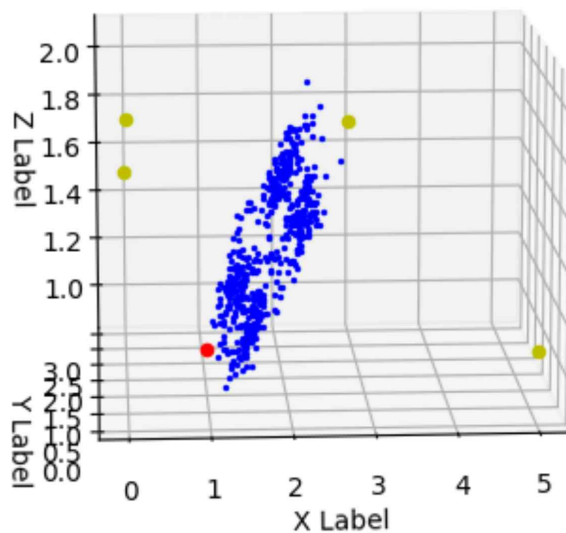
[그림 6] 3m 거리에서 측정한 RSSI 값(상), Outlier 제거(좌), 확장 칼만 필터 적용(우)



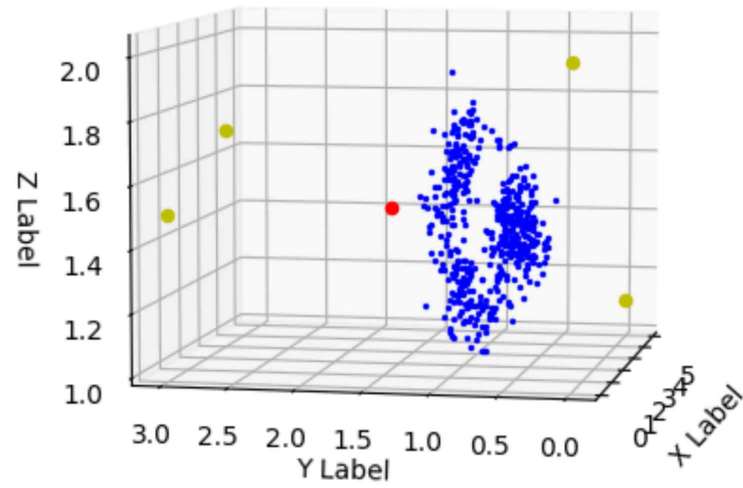


[그림 7] 4m 거리에서 측정한 RSSI 값(상), Outlier 제거(좌), 확장 칼만 필터 적용(우)

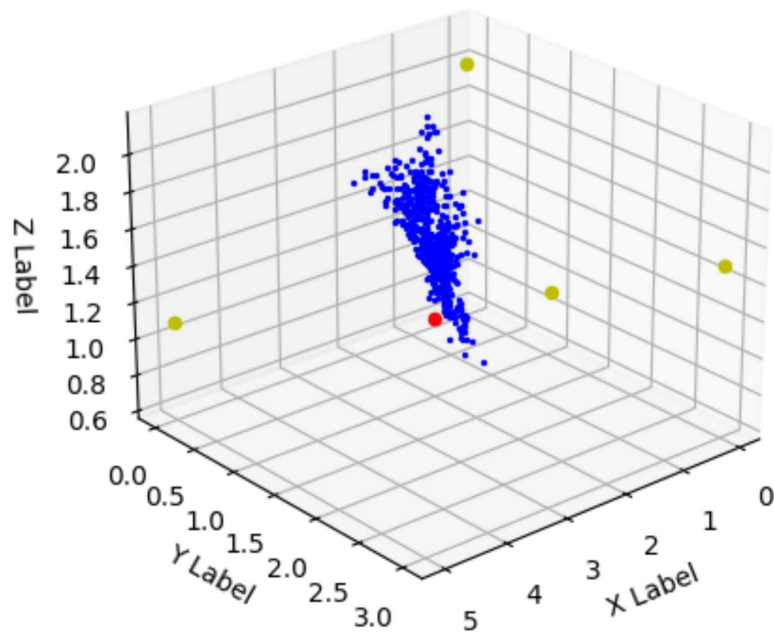
3D 공간상에서 Scanner 모듈 4개를 각각 (0, 0, 2), (3, 3, 1.7), (0, 3, 1.5), (5, 0, 1.1) 좌표 (단위: m)에 고정시키고, Tag 모듈의 위치를 바꿔가며 500회 측위한 결과는 아래의 [그림 8], [그림 9], [그림 10], [그림 11]과 같다. 각 그림에서 노란 점은 Scanner 모듈의 실제 위치, 붉은 점은 Tag 모듈의 실제 위치, 파란 점은 측정된 Tag 모듈의 위치를 의미한다.



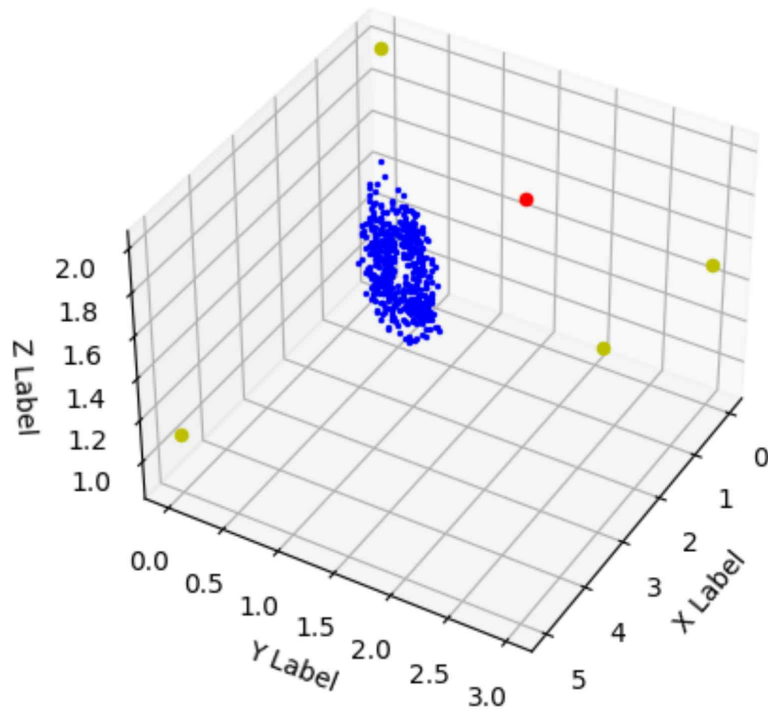
[그림 8] Tag 모듈의 실제 좌표가 (1, 1, 1)일 때 측위 결과



[그림 9] Tag 모듈의 실제 좌표가 (1.5, 1.5, 1.5)일 때 측위 결과

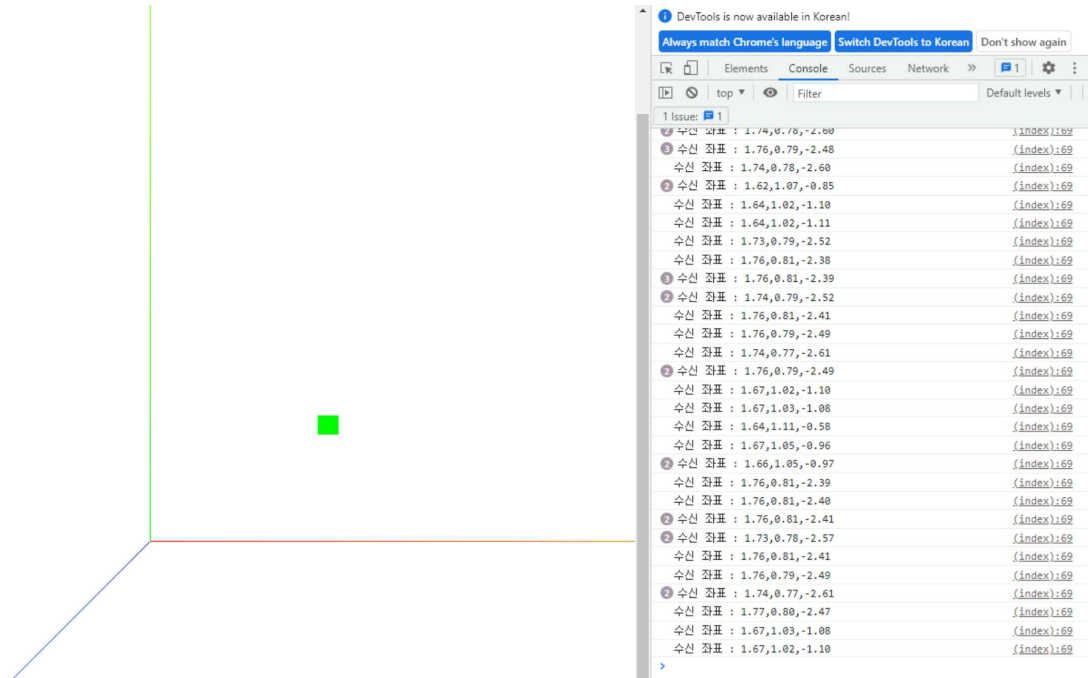


[그림 10] Tag 모듈의 실제 좌표가 (2, 1, 1)일 때 측위 결과



[그림 11] Tag 모듈의 실제 좌표가 (2, 2, 2)일 때 측위 결과

[그림 8], [그림 9], [그림 10], [그림 11] 에서의 파란 점의 분포를 보면, 각 축마다 최대 약 2m정도의 오차가 나타났고, 파란 점이 실제 위치와는 다른 곳에 위치해있다. RSSI 값은 오차범위 2.5dbm으로 보정이 잘 되었지만, 3D 좌표를 계산하는 과정에서 오차가 증폭되어 실제 위치와 최대 2~3m 떨어진 곳에 좌표가 나타났다.



[그림 12] 웹 서버로의 3D 좌표 전송과 렌더링

[그림 12]는 Web Socket을 통한 웹 서버로의 3D 좌표 전송 및 Client PC에서의 렌더링을 구현한 결과이다. 실시간 및 비동기적으로 데이터 송수신 및 렌더링이 되는 것을 확인할 수 있었다.

5. 결론 및 향후 연구 방향

우리는 BLE를 사용하여 3D 실시간 실내측위 및 동기화를 구현해보고자 하였다. Adafruit ItsyBitsy nRF52840 모델을 BLE Tag와 BLE Scanner 모듈로써 사용하고, RSSI 값을 전송하기 위해 필요한 프로토콜을 직접 설계했다. 확장 칼만 필터를 이용하여 측정된 RSSI 값의 오차를 제어하고, 변형된 사변측량 알고리즘을 통해 유일한 해를 계산하여 3D 공간상 좌표를 특정한 후 웹 서버에 업로드하는 과정 전체를 실험하였다.

기존의 BLE기반 RSSI를 이용한 측위는 오차가 약 3~8m정도로 나타난다고 알려져 있다. 본 과제에서는 RSSI 값의 오차를 $\pm 2.5\text{db}$ 정도로 줄였는데, RSSI값이 10db씩 변화할수록 약 3.3m의 오차가 발생하므로 기존의 시스템에 비해 더욱 정밀한 측위라고 할 수 있다. 다만 3D 좌표를 계산하는 과정에서 실제의 위치와는 다소 거리가 있는 곳에 좌표가 나타났는데, 사변측량 알고리즘으로 3D 좌표를 구하면서 더욱 증폭된 오차를 줄이고,

알고리즘 자체의 개선을 연구해볼 여지가 있다. 또한, BLE 모듈간 LoS(Line of Sight)가 보장되었는지를 RSSI 값만 가지고는 판별할 수 없으므로 수십cm 내외의 정밀한 측위를 달성하기 위해서는 단순히 오차 보정뿐만이 아닌 다중 경로, AoA(Angle of Arrival) 기술 등 다른 접근 방식으로도 연구할 필요가 있다.

6. 구성원별 역할 및 개발 일정

	6월					7월				8월					9월			
주	1	2	3	4	5	1	2	3	4	1	2	3	4	5	1	2	3	4
안현기		RSSI 측정 및 거리 계산 구현																
										오차 제어 필터 개발								
										WebGL 구현								
김춘수	Tag-Scanner간 통신 구현																	
						과제에 적합한 모듈 재조사				게이트웨이 모듈 구현								
										Web Socket 통신 구현								
이현진	Tag-Scanner간 통신 구현																	
						3D 좌표 계산 알고리즘 구현				WebGL 구현								
공통			기 말 고 사						중 간 보 고 서 작 성									최 종 보 고 서 작 성

7. 참고 문헌

- [1] I. Im, "Indoor Positioning System using BLE Beacon and Extended Kalman filter", KERIS, Aug. 2015. (in Korean)
- [2] Fengjun Shang, Wen Su, Qian Wang, "A Location Estimation Algorithm Based on RSSI Vector Similarity Degree", International Journal of Distributed Sensor Networks, Volume 2014, Article ID 371350, 22 pages, Aug. 2014.
- [3] Onkar Pathak, Pratik Palaskar, Rajesh Palkar, Mayur Tawari, "Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points -A Tri-lateration Approach", International Journal of Scientific & Engineering Research, Volume 5, Issue 4, Apr. 2014.