

2024 年度(令和 6 年度)

創造工学セミナー II Final Report

Wi-Fi/Bluetooth の電波強度を用いた追従対象の位置推定

知能機械研究室

s520063 山本浩隆

指導教員

金丸隆志 教授

内容

第 1 章 諸言	4
1.1 背景・目的	4
1.2 先行研究	5
1.2.1 Wi-Fi の電波強度を利用する先行研究	5
1.2.2 Bluetooth の電波強度を用いた先行研究	6
1.3 研究目的	7
第 2 章 自己位置推定システム	8
2.1 三角測量	8
2.2 自己位置推定システムで電波を受信する方法	8
2.2.1 Arduino IDE	8
2.2.2 M5Stack Core2	9
第 3 章 実験 1 : Wi-Fi の電波強度の測定実験	10
3.1 使用機器	10
3.1.1 Wi-Fi	10
3.1.2 電波強度測定機器	11
3.2 M5StickC PLUS を Wi-Fi のアクセスポイントとする方法	11
3.3 Wi-Fi の電波強度測定方法	12
3.4 実験方法	14
3.5 実験結果	14
3.6 考察	17
第 4 章 実験 2 : Bluetooth の電波強度の測定実験	18
4.1 使用機器	18
4.1.1 iBeacon	18
4.1.2 電波強度測定機器	18
4.2 M5StickC PLUS を iBeacon とする方法	18
4.3 Bluetooth の電波強度測定方法	22
4.4 実験方法	26
4.5 実験結果	27
4.6 考察	28
第 5 章 実験 3 : 複数の iBeacon を利用した電波強度の測定実験	29
5.1 使用機器	29

5.1.1	iBeacon.....	29
5.1.2	電波強度測定機器.....	29
5.2	実験方法.....	29
5.3	実験結果.....	29
5.4	電波強度を用いた位置推定	30
5.5	考察.....	32
第 6 章	結言.....	33
参考文献	34
謝辞	37

第1章 諸言

1.1 背景・目的

近年、ロボット技術の進展により、人間の日常生活を支援するロボットがより広く普及し始めている[1]。人間を支援するロボットの1つとして、対象の人物等を発見し、その動きを追従する追従ロボットが挙げられる[2]。案内や運搬用として人間に随行する支援ロボットだけでなく、スーツケースやベビーカーなどといった移動を伴うデバイスの多くが人物を追従するロボットに発展していくことが期待される。移動ロボットが人物を追従する手法は、ロボットに搭載されたRGB-Dカメラを用いて特定の色のジャケットを着用した追従対象を識別し、追従する手法[3]や、追従対象の足首に特定パターンで発光する赤外LEDタグを付け、LEDタグの発光パターンによって対象を識別、追従する手法[4]など多岐にわたるが、筆者はWi-Fi等の電波強度からスマートフォンなどの追従対象の周辺にあるものの位置を推定し、その対象を追従する手法[5]に着目した。なぜなら、屋外ではGPSを利用した精密な位置推定が可能であるものの、屋内ではGPSの電波が壁や屋根によってさえぎられ、位置推定を行えないからである。それに対し、Wi-Fi等の電波強度を用いる場合、屋内外問わず位置推定を行うことが可能である。

また、Wi-Fi等の電波強度を使用する手法は、特定の色のジャケットを着用する場合や赤外LEDタグを装着する場合などの他の追従対象識別手法に比べ、移動ロボットが人物を追従する際の環境の制約が少ないため、広域的に使用できると考えた。本システムは広く、あまり人が密集していないエリアを想定しており、ショッピングモールや美術館、空港などの屋内施設を利用対象としている。これらの施設ではキャリーケースや重たい荷物などを苦勞しながら持ち歩いている人[6]を見かける機会が多いため、これらの施設を利用対象とした。

本システムでは、これらの対象地域において、スーツケースや購入された商品等を運ぶ移動ロボット[7]や、施設を案内するガイドロボットなどとして利用することを想定している。

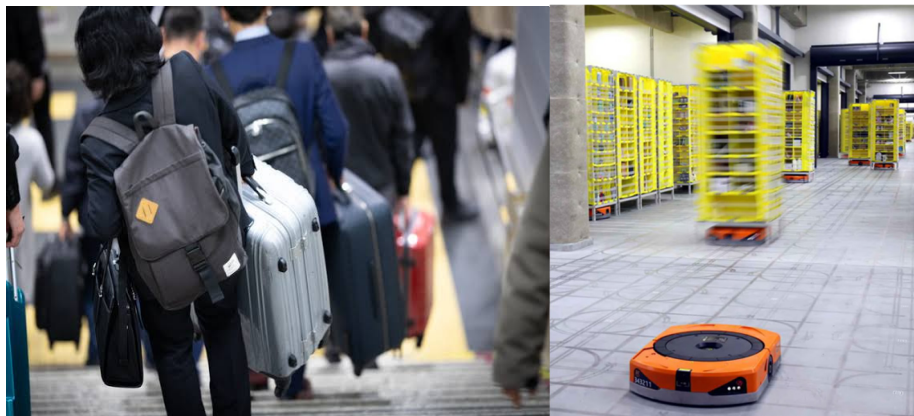


図1 重い荷物を運ぶ人[6](左)と商品運搬ロボット[7](右)

1.2 先行研究

本節では電波強度を用いた位置推定に関する先行研究について説明する。

1.2.1 Wi-Fi の電波強度を利用する先行研究

大竹らの研究では、環境に配置されている既存の Wi-Fi アクセスポイントの電波強度 (RSSI) を利用している [5]。まず、Wi-Fi アクセスポイントの電波強度を基に、追従対象のスマートフォンが最も電波強度が強いアクセスポイントを特定する。次に、移動ロボットが Wi-Fi 電波強度マップを用いて、そのアクセスポイントの電波強度が最大となるポイントをウェイポイントとして選択する。その後、移動ロボットはその情報をもとに追従対象に近づく。大竹らの手法では Wi-Fi を利用しているため、電波強度が不安定で、同じ地点で電波強度を計測した場合でも頻繁に電波強度が変化し、ロボットの追従精度が低下する可能性がある。そのため、ロボットにウェイポイント選択の制約を設けず、電波強度のみに基づいて追従目標位置を取得して走行させる実験では、ロボットが目標地点を頻繁に変更したり停止したりする状況が観測されている。また、Wi-Fi 電波強度マップの作成には、既存の Wi-Fi アクセスポイントが十分な密度で設置されている環境が前提となるため、本手法は適用可能な場所が制限される。図 2 は大竹らの Wi-Fi の電波強度を利用した追従手法を示したものである。

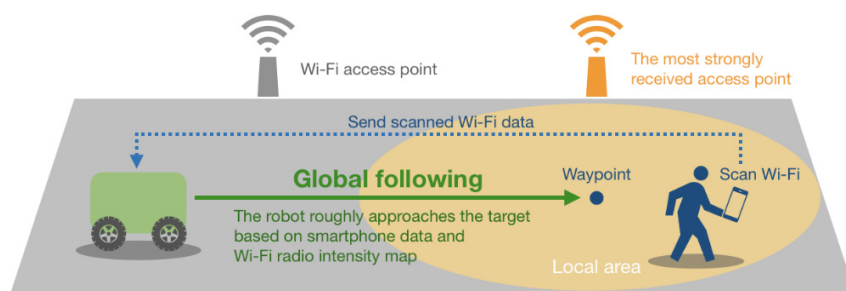


図 2 大竹らの Wi-Fi の電波強度を利用した人物追従手法 [5]

1.2.2 Bluetooth の電波強度を用いた先行研究

内田らの研究では、iBeacon 技術を活用した建設現場の見学者向け案内システムを開発している[8]。このシステムは、スマートフォンと iBeacon(BLE: Bluetooth Low Energy)を利用し、見学場所に応じて画像や音声で案内するためのものである。まず、見学者にスマートフォンでアプリをダウンロードしてもらう。そして、見学者が見学エリアに設置された iBeacon 内蔵ポールや iBeacon 内蔵マットのあるエリアに入ると、スマートフォンが自動で画像と音声で案内を開始し、見学者が iBeacon のあるエリアを離れると自動で案内を終了するというシステムである。

実験の結果、電波強度から正確な距離を測定することは困難であり、測距誤差が大きいと結論づけられている。図 3 は、実際に設置された iBeacon 内蔵ポールと iBeacon 内蔵マットを示している。



図 3 設置中の iBeacon 内蔵ポール(上)と iBeacon 内蔵マット(下)[8]

1.3 研究目的

1.1 節で述べたように、本研究では Wi-Fi や Bluetooth 等の電波強度から追従対象の位置を推定し、その対象を追従ロボット等適用することを目的としている。これにより、本システムの利用者、つまり追従対象は環境の制約なく人物追従ロボットを利用することが可能になる。

その結果、追従ロボット等の利用に対しての敷居を下げ、追従ロボットさらにはロボットに対するイメージの向上により、ロボット産業そのものの普及に寄与することを目標としている。

第 2 章 自己位置推定システム

本章では本研究で開発する自己位置推定システムに必要な技術について説明する。

2.1 三角測量

三角測量は、位置のわからないターゲットをあらかじめ位置を把握しているアクセスポイントからの距離で推定する手法である。

まず、アクセスポイントから発射した電波がターゲットに届く際にターゲットが受け取る電波の強度から 2 点間の距離を算出する。この情報からターゲットはアクセスポイントを中心とする、半径が測定距離の円上のどこかに存在しているということが分かり、この工程を同様に他のアクセスポイントについても行うことで、円の交点がターゲットの位置であるとおよそ推定することができる。

2.2 自己位置推定システムで電波を受信する方法

本節では自己位置推定システムについて説明する。本システムでは、複数の Wi-Fi、Bluetooth 等のアクセスポイントの電波を利用し、その強度に基づく三角測量の要領で対象の位置推定を行う。

以下の 2.2.1 節、2.2.2 節では、自己位置推定システムで電波を受信する際に使用する Arduino IDE と M5Stack Core2 について説明する。

2.2.1 Arduino IDE

Arduino IDE とは、Arduino と呼ばれるマイコンボード用プログラムを開発するためのオープンソースソフトウェアであり、arduino.cc によって開発された統合開発環境である。Arduino IDE は、さまざまなオペレーティングシステム (Windows、Linux、mac OS) 上で動作し、プログラミング言語 C/C++ を用いて、コードを書くためのテキストエディタ、メッセージエリア、テキストコンソール、一般的な機能のためのボタンを備えたツールバー、および一連のメニューが含まれている。Arduino IDE はハードウェアに接続してプログラムをアップロードし、それらと通信することも可能である [9]。

2.2.2 M5Stack Core2

M5Stack Core2 は、M5Stack 開発キットシリーズの第 2 世代のコアデバイスで、オリジナル世代のコアの機能をさらに強化したものである。

MCU は ESP32 モデル D0WDQ6-V3 であり、デュアルコアの Xtensa® 32-bit 240Mhz LX6 プロセッサを搭載し、個別に制御することが可能である。Wi-Fi を標準サポートし、オンボードで 16MB Flash と 8MB PSRAM、充電やプログラムのダウンロード、シリアル通信が可能な USB TYPE-C インターフェース、2.0 インチ内蔵静電容量式タッチパネル、内蔵振動モーターを搭載している[10]。

本研究では、この M5Stack Core2 用のプログラムを、前述の Arduino IDE を用いて開発する。プログラムは Wi-Fi や Bluetooth の電波強度を測定するためのものであり、それぞれ第 3 章および第 4 章で解説する。



図 4 M5Stack Core2[11]

第3章 実験1: Wi-Fiの電波強度の測定実験

本章では、自己位置推定に用いる Wi-Fi の電波強度の測定実験を行う。

3.1 使用機器

本節では本実験で使用した機器の説明をする。

3.1.1 Wi-Fi

本実験では3種類の Wi-Fi 機器を用いる。これらはそれぞれ電波強度が異なる。まず、図5は工学院大学八王子キャンパス4号館853室に設置されている NEC WG2600HS2 である。図6はマイコンの M5StickC PLUS であり、これを Wi-Fi のアクセスポイントとして用いる。図7は iPhone15 であり、そのテザリング機能を利用し、Wi-Fi 環境を実現する。



図5 NEC WG2600HS2



図6 M5StickC PLUS[12]



図 7 iPhone 15

3.1.2 電波強度測定機器

3.1.1 節で解説した 3 種類の Wi-Fi 環境に対し、2.2.1 節、2.2.2 節で示した Arduino IDE と M5Stack Core2 を使用し、電波強度を測定する。まず、M5StickC PLUS を Wi-Fi のアクセスポイントとする方法を解説する。

3.2 M5StickC PLUS を Wi-Fi のアクセスポイントとする方法

M5StickC PLUS を Wi-Fi のアクセスポイントとするためのコードは以下である (コード 1)[13]。

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WiFiAP.h>

const char *ssid = "yourAP";
const char *password = "yourpasswd";

void setup_wifi() {
  WiFi.softAP(ssid, password);
```

```

IPAddress myIP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(myIP);
}
void setup() {
  setup_wifi();
}

void loop(){
}

```

コード 1 M5StickC PLUS を Wi-Fi のアクセスポイントとするコード[13]

このコードの 5 行目が Wi-Fi アクセスポイントの名称となる SSID を決定するコードで、このコードでは SSID として `yourAP` が設定されている。また、6 行目のコードでは Wi-Fi アクセスポイントのパスワードを変更することが可能である。

3.3 Wi-Fi の電波強度測定方法

2.2.2 節で述べたように、Wi-Fi の電波強度を測定するために、M5Stack Core2 を利用する。その際、以下のコードを用いた（コード 2）[14][15]。

```

#include <M5Core2.h>
#include "WiFi.h"

void setup() {
  M5.begin();
  WiFi.mode(WIFI_STA);

  WiFi.disconnect();
  delay(100);
  M5.Lcd.print("WIFI SCAN");
}

void loop() {

```

```

M5.Lcd.setTextSize(2);
M5.Lcd.setCursor(0, 0);
M5.Lcd.println("Please press Btn.A to (re)scan");
M5.update();
if (M5.BtnA.isPressed()) {
    M5.Lcd.clear();
    M5.Lcd.println("scan start");
    int n = WiFi.scanNetworks();

    if (n == 0) {
        M5.Lcd.println("no networks found");
    } else {
        M5.Lcd.printf("networks found:%d¥n¥n", n);
        for (int i = 0; i < n;
            ++i) {

            String s = WiFi.SSID(i);
            if(s.indexOf("kanamaru")>=0 || s.indexOf("AP-TUT")>=0 ||
s.indexOf("yourAP")>=0 || s.indexOf("iPhone")>=0 ){
                M5.Lcd.printf("%d:", i + 1);
                M5.Lcd.print(WiFi.SSID(i));
                M5.Lcd.printf("(%d)", WiFi.RSSI(i));
                M5.Lcd.println(
                    (WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " " : "*");
                delay(10);
            }
        }
    }
    delay(1000);
}
}

```

コード 2 M5Stack Core2 で Wi-Fi の電波強度を測定するコード [14][15]

このコードでは、マイコンが検知できる全ての Wi-Fi とその電波強度がモニター画面に表示されることを防ぐため、表示される Wi-Fi に制限をかけた。それが 31 行目のコードであ

り、"kanamaru"、"AP-TUT"、"yourAP"、"iPhone"のどれかの文字列が SSID に含まれるアクセスポイントのみが M5Stack Core2 の画面に表示される。

3.4 実験方法

Wi-Fi の電波強度の測定実験は、以下の 3 つの手順で行う。

<実験手順>

1. 3 種類の電波強度の異なる Wi-Fi 環境のどれか 1 つを屋内に用意する
2. Wi-Fi のアクセスポイントから 0m から 5m の範囲内で電波強度を 0.5m 毎に 10 回、計 110 回測定する
3. それぞれの測定結果をグラフ化する

3.5 実験結果

本節では、Wi-Fi の電波強度の測定実験の結果について説明する。

図 8 が M5StackCore2 に表示された電波強度の測定結果である。networks found の数が 21 と表示されているため、21 個の Wi-Fi アクセスポイントが検出されているが、3.3 節で指定した文字列が含まれる Wi-Fi のみが表示されていることがわかる。ネットワーク名の後のかっこ内に記されているのが、そのアクセスポイントの電波強度が dBm 単位で表示されたものである。

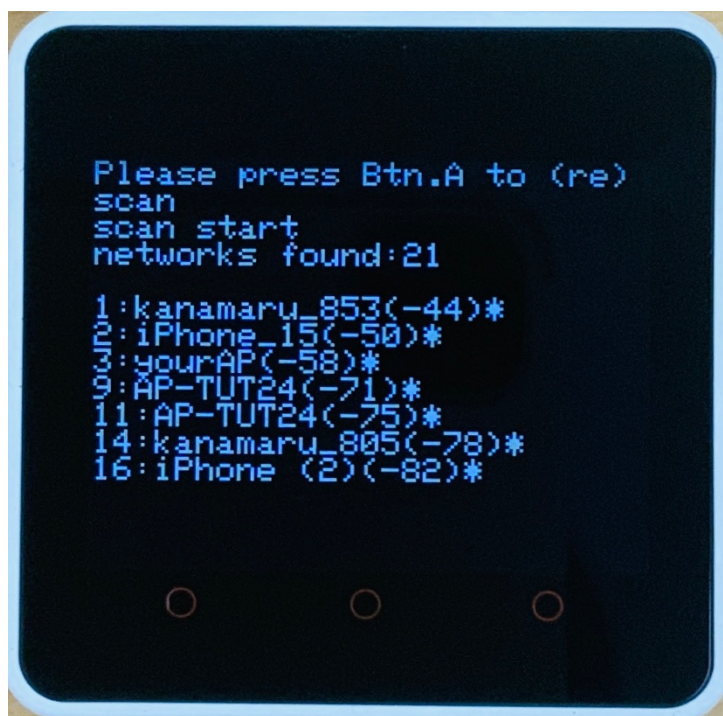


図 8 M5Stack Core2 に表示された電波強度

この電波強度の距離に対する変化を読み取っていく。図 9、図 10、図 11 がその結果である。図 9 が NEC WG2600HS2、図 10 が M5StickC PLUS を Wi-Fi のアクセスポイントとしたもの、図 11 が iPhone15 のテザリングによるものに対応している。図 12 はこれら 3 つのグラフを 1 つにまとめたものである。

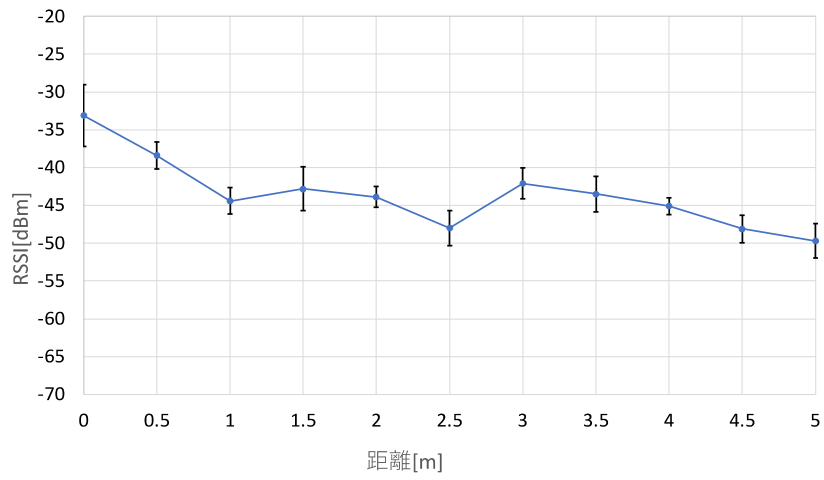


図 9 NEC WG2600HS2 の電波強度

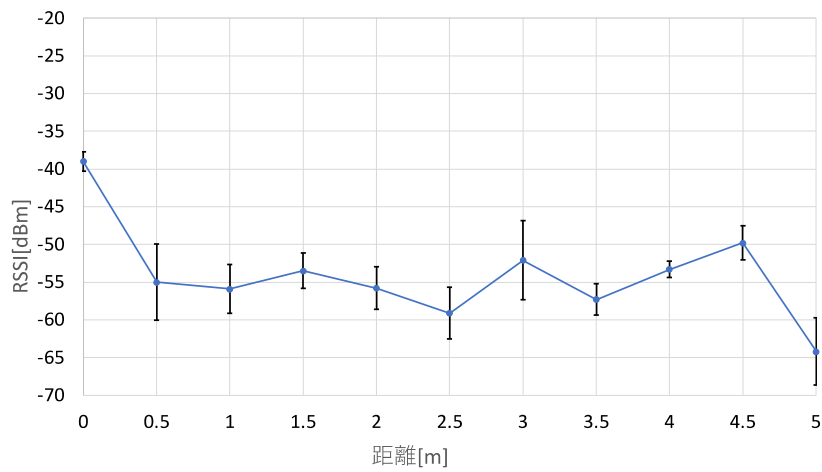


図 10 M5StickC PLUS の電波強度

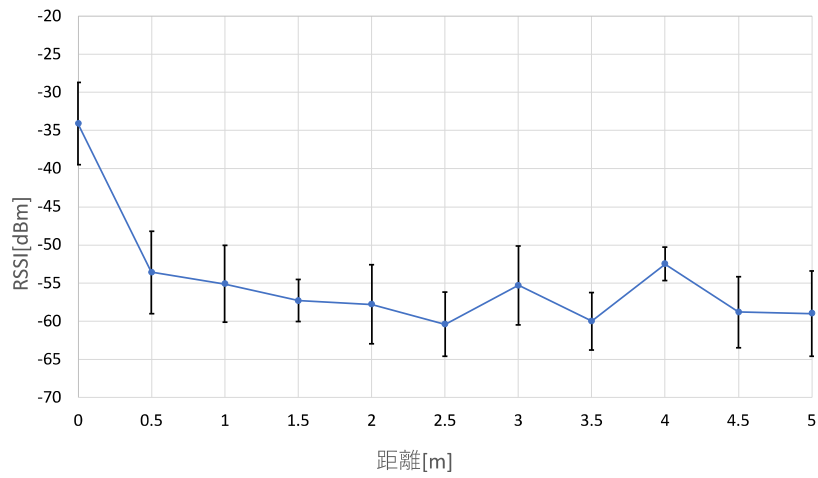


図 11 iPhone15 の電波強度

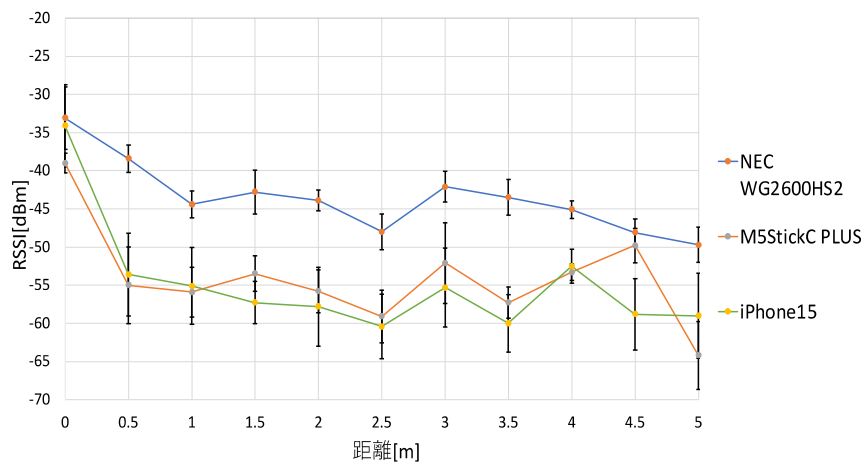


図 12 3つの Wi-Fi アクセスポイントの電波強度の比較

3.6 考察

図 12 を見てわかる通り、3つの Wi-Fi アクセスポイントの電波強度は、距離に対してあまり変化しておらず、さらに電波強度が安定していないとわかる。以上の理由から、Wi-Fi の電波強度を用いた位置推定は難しいだろうと考えられる。

第 4 章 実験 2 : Bluetooth の電波強度の測定実験

Wi-Fi と比較して通信距離の短い Bluetooth を利用して、第 3 章と同様の実験を行う。

4.1 使用機器

本節では本実験で使用した機器の説明をする。

4.1.1 iBeacon

本実験では Bluetooth のアクセスポイントが必要であるため、図 6 の M5StickC PLUS を iBeacon とし、Bluetooth のアクセスポイントとした。iBeacon とは、BLE(Bluetooth Low Energy)を利用した Apple 社が採用する近距離通信技術および通信プロトコルである。iOS7 以降の iOS 端末においては、iBeacon 発信機に近づいたときにアドバタイジングパケットを受信して自動的に何らかの動作をするアプリを作成することが可能である。iBeacon 発信機から発信する ID 情報は、Proximity UUID、Major、Minor の 3 種類の識別子から構成されている。1 つ以上の iBeacon 発信機を UUID で特定し、Major により同一 UUID の発信機をグループ分けすることができる。さらに、同一 Major の発信機から Minor によって個々の識別が可能である [8]。

4.1.2 電波強度測定機器

本実験では、電波強度を測定するために、3.1.2 節で示した M5Stack Core2 を用いる。

4.2 M5StickC PLUS を iBeacon とする方法

M5StickC PLUS を iBeacon とするためのコードは以下である (コード 3) [16]。

```
#include <M5StickCPlus.h>
#include "BLEDevice.h"
#include "BLEUtils.h"
#include "BLEBeacon.h"
```

```

#define BEACON_UUID          "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" // UUID 1
128-Bit (may use linux tool uuidgen or random numbers via
https://www.uuidgenerator.net/)

#define MAJOR    1000
#define MINOR    1000

#define TX_POWER -65

#ifdef __cplusplus
extern "C" {
#endif

#ifdef __cplusplus
}
#endif

BLEAdvertising *pAdvertising;

std::string setUUID(){
    std::string org = BEACON_UUID;
    std::string dst = "";
    if( org.length() != 36 ){
        return "error";
    }
    dst = org[34];
    dst += org[35];
    dst += org[32];
    dst += org[33];
    dst += org[30];
    dst += org[31];
    dst += org[28];
    dst += org[29];
    dst += org[8];
    dst += org[26];
    dst += org[27];
}

```

```
dst += org[24];
dst += org[25];
dst += org[23];
dst += org[21];
dst += org[22];
dst += org[19];
dst += org[20];
dst += org[18];
dst += org[16];
dst += org[17];
dst += org[14];
dst += org[15];
dst += org[13];
dst += org[11];
dst += org[12];
dst += org[9];
dst += org[10];
dst += org[6];
dst += org[7];
dst += org[4];
dst += org[5];
dst += org[2];
dst += org[3];
dst += org[0];
dst += org[1];
return dst;
}

void setBeacon() {

    BLEBeacon oBeacon = BLEBeacon();
    oBeacon.setManufacturerId(0x4C00);
    oBeacon.setProximityUUID(BLEUUID(setUUID()));
    oBeacon.setMajor(MAJOR);
    oBeacon.setMinor(MINOR);
    oBeacon.setSignalPower(TX_POWER);
```

```
BLEAdvertisementData oAdvertisementData = BLEAdvertisementData();
BLEAdvertisementData oScanResponseData = BLEAdvertisementData();

oAdvertisementData.setFlags(0x04);

std::string strServiceData = "";

strServiceData += (char)26;
strServiceData += (char)0xFF;
strServiceData += oBeacon.getData();
oAdvertisementData.addData(strServiceData);

pAdvertising->setAdvertisementData(oAdvertisementData);
pAdvertising->setScanResponseData(oScanResponseData);

}
#define LED 10

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(M5_BUTTON_HOME, INPUT_PULLUP);
  digitalWrite(LED, HIGH);
  M5.begin();
  M5.Lcd.setRotation(1);
  M5.Lcd.fillScreen(BLACK);

  M5.Lcd.setCursor(2, 0, 2);
  M5.Lcd.printf("UUID: %s", BEACON_UUID);
  M5.Lcd.setCursor(2, 40, 2);
  M5.Lcd.printf("Major %d Minor %d", MAJOR, MINOR);

  Serial.begin(115200);
  BLEDevice::init("");
  pAdvertising = BLEDevice::getAdvertising();

  setBeacon();
}
```

```

}

void loop() {
  pAdvertising->start();
  digitalWrite(LED, LOW);
  Serial.println("Advertizing started...");
  delay(100);
  Serial.println("Advertizing stop...");
  pAdvertising->stop();
  digitalWrite(LED, HIGH);
  delay(900);
}

```

コード 3 M5StickC PLUS を iBeacon にするコード[16]

4.3 Bluetooth の電波強度測定方法

Bluetooth の電波強度を測定するために、M5Stack Core2 を利用する。その際、以下のコードを用いた（コード 4）[17]。

```

#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>
#include <BLEAdvertisedDevice.h>
#include <M5Core2.h>

class IBeaconAdvertised: public BLEAdvertisedDeviceCallbacks {
public:

  void onResult(BLEAdvertisedDevice device) {
    if (!isIBeacon(device)) {
      return;
    }
    printIBeacon(device);
  }
}

```

private:

```
bool isIBeacon(BLEAdvertisedDevice device) {
    if (device.getManufacturerData().length() < 25) {
        return false;
    }
    if (getCompanyId(device) != 0x004C) {
        return false;
    }
    if (getIBeaconHeader(device) != 0x1502) {
        return false;
    }
    return true;
}

unsigned short getCompanyId(BLEAdvertisedDevice device) {
    const unsigned short* pCompanyId = (const unsigned short*)&device
        .getManufacturerData()
        .c_str()[0];

    return *pCompanyId;
}

unsigned short getIBeaconHeader(BLEAdvertisedDevice device) {
    const unsigned short* pHeader = (const unsigned short*)&device
        .getManufacturerData()
        .c_str()[2];

    return *pHeader;
}

String getUuid(BLEAdvertisedDevice device) {
    const char* pUuid = &device.getManufacturerData().c_str()[4];
    char uuid[64] = {0};
    sprintf(
        uuid,
```

```

"%02X%02X%02X%02X-%02X%02X-%02X%02X-%02X%02X-%02X%02X%02X%02X
%02X%02X",
    pUuid[0], pUuid[1], pUuid[2], pUuid[3], pUuid[4], pUuid[5], pUuid[6],
pUuid[7],
    pUuid[8], pUuid[9], pUuid[10], pUuid[11], pUuid[12], pUuid[13], pUuid[14],
pUuid[15]
    );
    return String(uuid);
}

signed char getTxPower(BLEAdvertisedDevice device) {
    const signed char* pTxPower = (const signed char*)&device
        .getManufacturerData()
        .c_str()[24];

    return *pTxPower;
}

void printIBeacon(BLEAdvertisedDevice device) {
    M5.Lcd.printf("rssi:%d uuid:%s¥n",
        device.getRSSI(),
        getUuid(device).c_str());
}
};

String getUuid(BLEAdvertisedDevice device) {
    const char* pUuid = &device.getManufacturerData().c_str()[4];
    char uuid[64] = {0};
    sprintf(
        uuid,

"%02X%02X%02X%02X-%02X%02X-%02X%02X-%02X%02X-%02X%02X%02X%02X
%02X%02X",
        pUuid[0], pUuid[1], pUuid[2], pUuid[3], pUuid[4], pUuid[5], pUuid[6], pUuid[7],
        pUuid[8], pUuid[9], pUuid[10], pUuid[11], pUuid[12], pUuid[13], pUuid[14],
pUuid[15]

```



```

);
return String(uuid);
}

void setup() {
  M5.begin();
  Serial.begin(115200);
  BLEDevice::init("");
  delay(100);
}

void loop() {
  M5.Lcd.setTextSize(2);
  M5.Lcd.setCursor(0, 0);
  M5.Lcd.println("Please press Btn.A to (re)scan");
  M5.update();
  if (M5.BtnA.isPressed()) {
    M5.Lcd.clear();
    M5.Lcd.println("scan start");

    BLEScan* scan = BLEDevice::getScan();
    scan->setAdvertisedDeviceCallbacks(new IBeaconAdvertised(), true);
    scan->setActiveScan(true);
    BLEScanResults results = scan->start(10);

    int n = results.getCount();

    if (n == 0) {
      M5.Lcd.println("no networks found");
    } else {
      M5.Lcd.printf("networks found:%d¥n¥n", n);
      for (int i = 0; i < n;
            ++i) {

        BLEAdvertisedDevice device = results.getDevice(i);

```

```

        int rssi = device.getRSSI();
        String uuid = getUuid(device).c_str();
        M5.Lcd.printf("RSSI: %d, UUID: %s¥n", rssi, uuid.c_str());
        delay(10);
    }
}
delay(1000);
}

//BLEscanResults foundDevices = scan->start(60);
//M5.lcd.println("Devices found: ");
//M5.lcd.println(foundDevices.getCount());
Serial.println("complete.");
}

```

コード 4 M5Stack Core2 で Bluetooth の電波強度を測定するコード [17]

4.4 実験方法

Bluetooth の電波強度の測定実験は、以下の 3 つの手順で行う。

<実験手順>

1. Bluetooth の iBeacon 環境を屋内に用意する
2. Bluetooth の iBeacon から 0m から 5m の範囲内で電波強度を 0.5m 毎に 10 回、計 110 回測定する
3. 実験 1 の Wi-Fi の電波強度の測定結果と本実験の Bluetooth の電波強度の測定結果をグラフ化する

4.5 実験結果

本節では、Bluetooth の電波強度の測定実験の結果について説明する。

実験の結果を図示したものが図 13 である。また、図 14 は図 12 の Wi-Fi のグラフと図 13 を同時に描いたものである。

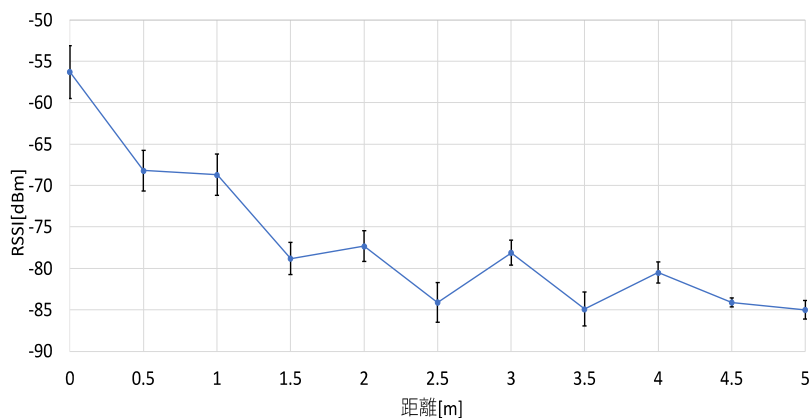


図 13 iBeacon の電波強度

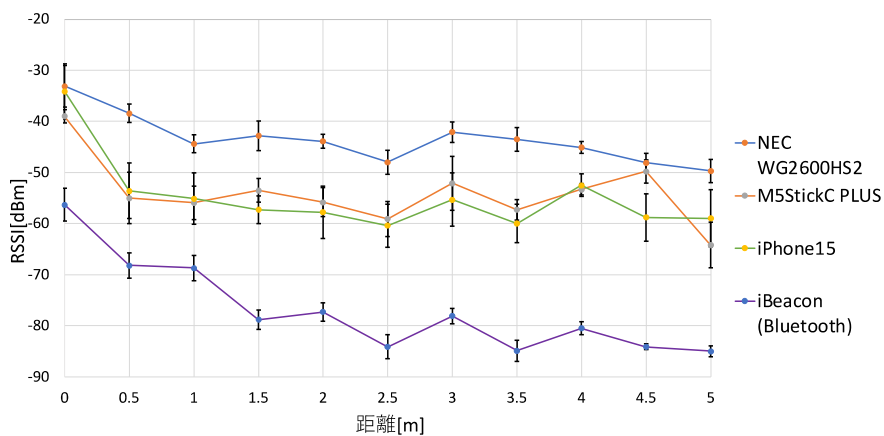


図 14 Wi-Fi と Bluetooth の電波強度の比較

4.6 考察

図 14 を見てわかる通り、Wi-Fi は 0-0.5m くらいまでしか電波強度が大きく変化しておらず、その後の電波強度のグラフの傾きは小さくなっているとわかる。それに対し、iBeacon、つまり Bluetooth の電波強度は距離の変化に伴って電波強度が変化しており、特に 0-1.5m 付近までは大きく変化している。よって、Wi-Fi と比較した際、Bluetooth が電波強度の変化を利用する本提案手法に適しているとわかる。

第5章 実験3：複数の iBeacon を利用した電波強度の測定実験

第3章と第4章の実験結果から、Wi-Fi よりも Bluetooth の方が電波強度の変動を利用する本提案手法に適していると判断した。そこで、実際に位置推定が実現可能なのかを調べるため、実験3として、複数の iBeacon を利用した電波強度の測定実験を行う。

5.1 使用機器

本節では本実験で使用した機器の説明をする。

5.1.1 iBeacon

本実験では、4.1.1 節で示した M5StickC PLUS を iBeacon として4つ用いる。プログラムは4.2 節のものをを用いる。

5.1.2 電波強度測定機器

本実験では、3.1.2 節で示した M5Stack Core2 を電波強度測定用に用いる。プログラムは4.3 節のものをを用いる。

5.2 実験方法

複数の iBeacon を利用した電波強度の測定実験は、以下の3つの手順で行う。

<実験手順>

1. 4つの iBeacon を用意し、屋内の3メートル四方の空間の隅にそれぞれ設置する
2. iBeacon から0m から3m の範囲内で0.5m 毎に10回、計490回測定し、各 iBeacon に関する電波強度のデータ、計1960個を取得する
3. それぞれの測定結果を電波強度マップとしてまとめる

5.3 実験結果

本節では、複数の iBeacon を利用した電波強度の測定実験の結果について説明する。

図 15 が作成された電波強度マップである。これは、左上に iBeacon が存在する場合のものである。実験では 4 つそれぞれの iBeacon に関する電波強度マップを得ることができたが、それぞれはほぼ同様のマップであった。

信号強度map								
		距離[m]						
iBeaconの位置		0	0.5	1	1.5	2	2.5	3
距離[m]	0	-53.3	-53.6	-63.5	-66.9	-75.9	-86.7	-79.6
	0.5	-60.8	-65.8	-70.3	-67	-76.8	-81.8	-79.2
	1	-65.3	-69.9	-73.2	-68.4	-66.3	-68.1	-69.4
	1.5	-65.5	-71.4	-65.3	-72.5	-72.2	-57.2	-69.7
	2	-71.4	-73.8	-69.3	-73.5	-84.6	-76.1	-75.2
	2.5	-71.5	-83.1	-82.4	-73.7	-78.3	-75.9	-70.8
	3	-81.6	-85.6	-78.9	-70	-73.1	-76.4	-74.7

図 15 iBeacon の電波強度 map

5.4 電波強度を用いた位置推定

最後に、電波強度を用いた位置推定を行う。

図 16 と図 17 は Wi-Fi と Bluetooth の位置推定における正しい距離 d と計測した電波強度から推定された距離 d' の関係を表したグラフである。

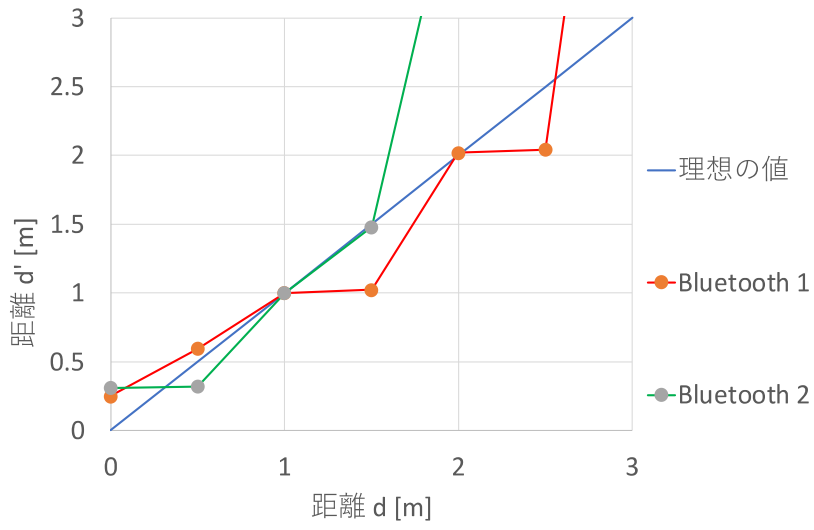


図 16 正しい距離 d と Bluetooth により推定された距離 d'

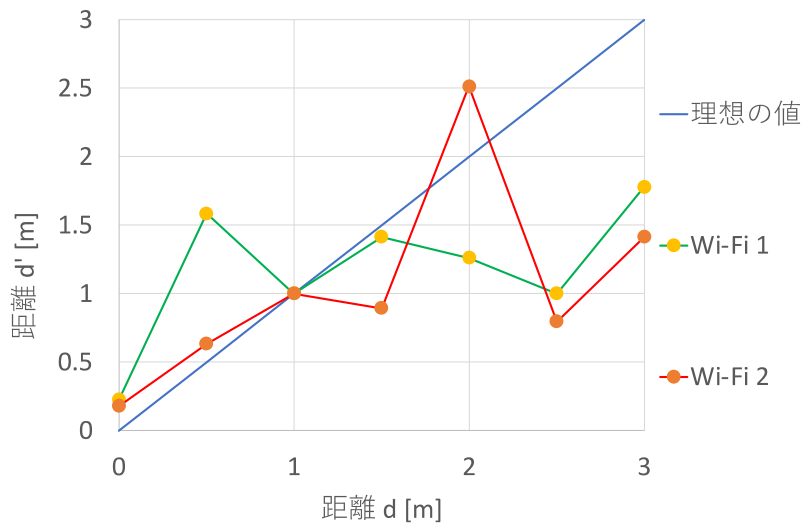


図 17 正しい距離 d と Wi-Fi により推定された距離 d'

図 16 と図 17 は、1.2.2 節で示した内田らの研究で示されていた、RSSI と距離の関係式である次式を図 14 にあてはめて用いた[8]。TxPower は、アクセスポイントからの距離 1m 地点での RSSI の値を図 14 のグラフから読み取り、用いた。

$$d = 10^a \times 100$$

ただし、

$$a = \frac{\text{TxPower} - \text{RSSI}}{20}$$

である。

5.5 考察

図 15 では赤い色が濃いほど電波強度が強くなっており、図をみてわかる通り、距離が離れるにつれて濃い赤色から薄い赤色に変化しているのが見て取れる。しかし、電波強度は iBeacon を中心に綺麗な円を描くように変化していると想定していたが、想定とは異なり、所々電波強度が弱い点などが点在しており、大まかに円を描いている程度であるとわかる。

また、図 16 と図 17 を見て分かる通り、Wi-Fi のビーコンから 0-0.5m くらいまでは理想の値と電波強度から求められた値が比較的近い。一方、Bluetooth は Wi-Fi に比べ、ビーコンからの距離が 0-1.5m くらいまで理想の値と電波強度から求められた値がかなり近く、一方で、距離がはなれるにつれて電波強度の変化が大きくなっており、近距離での位置推定に適しているとわかる。

第6章 結言

本研究では電波強度から対象の位置を推定し、荷物を運ぶ人物追従ロボット等へ適用することを目的として Wi-Fi と Bluetooth の電波強度を用いて追従対象の位置推定の精度がその目的に適しているのかを調べた。実験の結果から、Wi-Fi と Bluetooth の距離の変化にともなう電波強度の変化を比較した際、Bluetooth の電波強度が近距離において距離の変化に伴って電波強度が大きく変化しており、電波強度の変化を利用する本提案手法に適していると考察した。

しかし、本研究では実際にアクセスポイントを利用して三角測量による位置推定を行うことまではできなかった。位置推定を行い、実際のロボットに適用することが今後の課題である。

参考文献

[1]2023 年度 ロボット産業・技術振興に関する調査研究報告書 | 日本機械工業連合会
https://www.jmf.or.jp/jmf/wp-content/uploads/2024/04/23rbaw_h.pdf
(最終閲覧日：2025/02/09)

[2]人の後ろをついてくる「追従型ロボット」が、農場や店舗で活躍し始めている | WIRED
<https://wired.jp/2021/11/17/robots-follow-learn-where-go/>
(最終閲覧日：2025/02/09)

[3]萬 礼応 “人と協働する農作業支援ロボットの実践と評価” (2021)
https://www.jstage.jst.go.jp/article/jacc/64/0/64_269/_pdf/-char/ja
(最終閲覧日：2025/02/09)

[4] 小山 渚、但馬 竜介、田中 稔、廣瀬 徳晃、鋤柄 和俊、藤井 亮暢 “ヒトを活動的にする移動支援ロボット — ID を発信する赤外 LED タグと全方位カメラを用いた利用者の追跡 —” (2015)
https://www.jstage.jst.go.jp/article/sicetr/51/1/51_49/_pdf/-char/ja
(最終閲覧日：2025/02/09)

[5] 大竹 直道、森岡 一幸 “Wi-Fi に基づく大域的追従と歩行のマッチングに基づく局所的追従を融合したスマートフォンユーザーを追従する移動ロボットシステム” (2020/04/07)
https://www.jstage.jst.go.jp/article/transjsme/86/885/86_19-00402/_pdf/-char/ja
(最終閲覧日：2025/02/07)

[6] 重たいスーツケースを持って階段を下りる旅行者の無料の写真素材 | PAKUTASO
<https://www.pakutaso.com/20190739200post-20728.html>
(最終閲覧日：2025/02/07)

[7] アマゾン、大阪の物流倉庫にロボット導入 国内 2 カ所目 | 産経 WEST
<https://www.sankei.com/article/20190404-JKBCZ5S3XJO53DT3OQOTEPPRJA/>
(最終閲覧日：2025/02/07)

[8] 内田 理絵、川野邊 慧、岡本 修、藤原 泰明、高田 知典、宇野 昌利、宮瀬 文裕、米山 文雄、藤枝 達也 “iBeacon を利用した現場見学者のための工事概要案内システムの開発” (2016)

https://www.jstage.jst.go.jp/article/jscejcei/72/2/72_II_65/_pdf/-char/ja

(最終閲覧日：2025/02/07)

[9] Arduino Integrated Development Environment (IDE) v1 | Arduino Documentation

<https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/>

(最終閲覧日：2025/02/07)

[10] Core2 | M5STACK

<https://docs.m5stack.com/ja/core/core2>

(最終閲覧日：2025/02/07)

[11] タッチスクリーン備えた M5Stack Core2、ジェスチャーが実装可能に | 日経 BP

<https://xtech.nikkei.com/atcl/nxt/column/18/01109/080100033/>

(最終閲覧日：2025/02/07)

[12] #M5StickC PLUS ESP32-PICO Mini IoT Development Kit | M5STACK

https://shop.m5stack.com/products/m5stickc-plus-esp32-pico-mini-iot-development-kit?srsId=AfmBOor-bqfHR4lvJPFZ15oHRHJyAlYIW0xxOGqDY_b2dEsiUsI4P_7z

(最終閲覧日：2025/02/07)

[13] M5Stack で Wifi を AP と STA とで使い分ける | Qiita

<https://qiita.com/nnn112358/items/fbb4f24505c6ef8082cd>

(最終閲覧日：2025/02/09)

[14] m5stack | M5Core2 | GitHub

<https://github.com/m5stack/M5Core2/blob/master/examples/Advanced/WIFI/WiFiScan/WiFiScan.ino>

(最終閲覧日：2025/02/09)

[15]M5Stack Basic と M5Stack Core2 のデフォルトフォントのサイズステップが分かった
メモ | 1ft-seabass.jp.MEMO

<https://www.1ft-seabass.jp/memo/2021/02/12/m5stack-basic-and-core2-default-fontsize-maybe-7px-knowledge/>

(最終閲覧日 : 2025/02/09)

[16]M5StickC を iBeacon にする | Qiita

<https://qiita.com/SamAkada/items/8860f3ba77233852d24a>

(最終閲覧日 : 2025/02/09)

[17]M5StickC で iBeacon をスキャンする | Fusic TECH BLOG

<https://tech.fusic.co.jp/posts/2020-02-23-m5stickc-ibeacon/>

(最終閲覧日 : 2025/02/09)

謝辞

最後に、本研究を行うにあたり、2年間多大なご指導をいただきました金丸隆志教授、そして本研究に協力してくださった研究室の先輩方、同級生に心より感謝いたします。