

# BLE RSS dataset for fingerprinting radio map calibration

Marcin Kolakowski 

*Institute of Radioelectronics and Multimedia Technology*  
*Warsaw University of Technology*  
Warsaw, Poland  
m.kolakowski@ire.pw.edu.pl

## Abstract

This document describes a dataset of Bluetooth Low Energy signal strengths measured in a fully furnished flat. The dataset was originally used in the study concerning RSS-fingerprinting-based indoor positioning systems. The data were gathered using a hybrid BLE-UWB localization system, which was installed in the apartment and a mobile robotic platform equipped for a LiDAR. The dataset comprises power measurement results and LiDAR scans performed in 4104 points. The scans used for initial environment mapping and power levels registered for two test scenarios are also attached. The set contains both raw and preprocessed measurement data. The Python code for raw data loading is supplied.

## Index Terms

BLE, fingerprinting, positioning, RSS

## I. INTRODUCTION

This dataset contains Received Signal Strengths (RSS) of the Bluetooth Low Energy signals registered in a fully furnished apartment. The data were used to assess the effectiveness of radio map calibration methods in a yet unpublished paper.

The dataset consists of three parts:

- SLAM measurements - 79 stationary scans gathered in an apartment,
- RSS calibration dataset - 4104 fingerprints,
- test dataset - RSS measurements performed in two test scenarios - robot and walking person localization.

## II. FILES

The dataset consists of files containing raw measurement results, processed measurement results and Python scripts for data handling. The data repository structure is presented below.

```
/
├── calibration
│   ├── calibX_ble.txt ..... file containing RSS measurement results for a calibration path X X = 1..4
│   ├── calibX_df.csv ..... preprocessed calibration DataFrames (LiDAR locations + RSS) X = 1..4
│   ├── calibX_motion.txt ..... file containing motion history for a calibration path X X = 1..4
│   ├── calibX_scan.txt ..... file containing scans for a calibration path X X = 1..4
│   └── calibration_df.csv ..... calibration DataFrame (tag locations & RSS measurement results)
├── slam
│   ├── gridmap.pickle ..... gridmap obtained using the GraphSLAM algorithm
│   ├── mapping_motion.txt ..... file containing motion history for SLAM measurements
│   └── mapping_scan.txt ..... scans gather during the SLAM session
├── src ..... Python functions for data loading
└── tests
    ├── test_robot_motion.txt ..... file containing motion history for a robot test path X
    └── test_robot_scans.txt ..... file containing scans for a robot test path X
```

test_robot_ble.txt	.....	file containing RSS measurement results for a robot test path X
test_robot_df.csv	.....	preprocessed test DataFrame (tag locations & RSS measurement results)
test_walk_ble.txt	.....	file containing RSS measurement results for a walking person
test_walk_path.csv	.....	walking test path coordinates
test_walk_df.csv	.....	preprocessed test DataFrame (RSS measurement results)
anchors.csv	.....	locations of BLE anchor nodes
layout.csv	.....	test site layout (the lines include consecutive vertices of the walls)
load_ble_results.py	.....	python script loading BLE results
prepare_scans.py	.....	python script loading and preparing scans for SLAM

### III. MEASUREMENTS DESCRIPTION

The measurements were performed using the localization system described in [1]. The layout of the apartment and the loactions of the anchors are presented in Fig.1.



Fig. 1. Measurement environment with measurement locations and significant objects obstructing the path between the devices marked.

The system infrastructure used in the study comprised six anchors, distributed among the apartment. The layout of the walls and anchor locations are stored in `layout.csv` and `anchors.csv` files respectively.

The used anchors are equipped with two Laird BL652 modules with external antennas of perpendicular polarization. In the system, the tag transmits BLE packets five times per second in three advertisement channels. The anchors periodically switch the reception channels and measure the power of the received signals. The results from both modules are averaged and sent to a localization server, which stores them in a database for future processing. The measurement results were gathered using a robotic platform, which is presented in Fig.2.

The platform was based on the Dagu Wild Thumper 6WD chassis and was controlled with a specially developed Python-based controller running on Raspberry Pi 4. The code for the controller and the PC applications used to gather the LiDAR results can be found at [2]. The odometry measurements were performed by estimating the traveled distance and rotation angle through multiplying the elapsed time by the linear and rotation speed respectively.



Fig. 2. The mobile platform used in the study.

The platform was equipped with a Scansweep LiDAR, which is a discontinued 360 degree range sensor. During the study, the LiDAR was set to collect scans with a 2 Hz rate.

The system tag was attached at the middle the robot on a one meter long wooden pole, which placed it at the height of 1.3 m. Such height is close to a tag worn on a lanyard and ensures that low furniture pieces such as bed frames, couches or tables won't negatively impact the RSS measurement results.

The location of the calibration points are presented in Fig. 3.

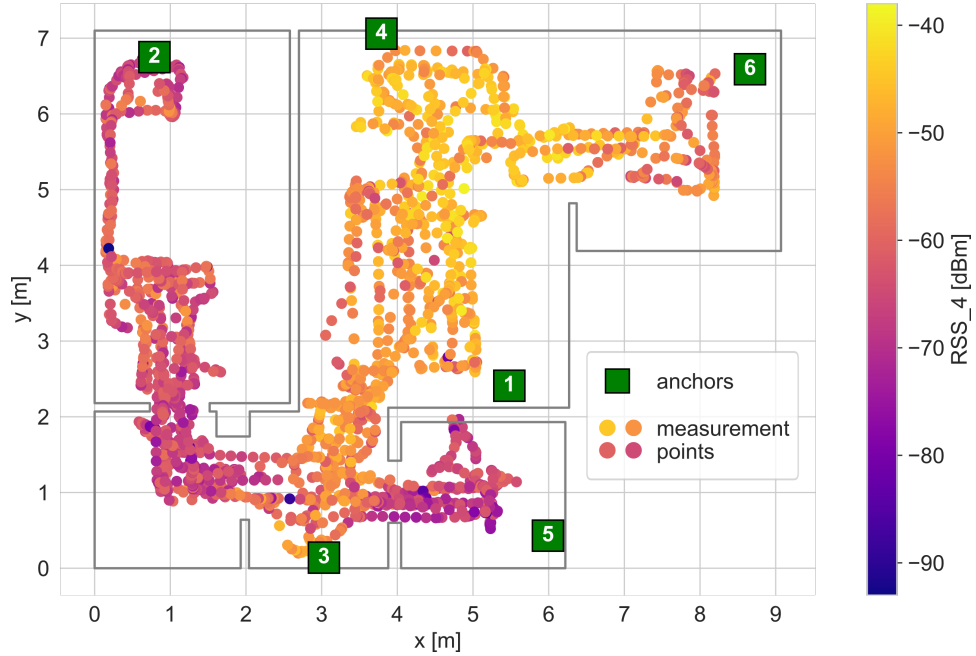


Fig. 3. Locations of calibration measurements. The colours of the points reflect the measured RSS value by anchor 4.

#### IV. DATA PROCESSING

The dataset contains preprocessed data. The SLAM measurements were used to create a radio map, which is a basis for positioning of the robotic platform during the calibration and test phases.

The processed calibration files contain DataFrames with robot locations and RSS values measured in them. The `calibration_df.csv` file contains a DataFrame in which the robot locations are recalculated to take into

account the displacement of the system tag from the LiDAR. The top-view schematic of the platform marking the respective sensors is presented in Fig. 4.

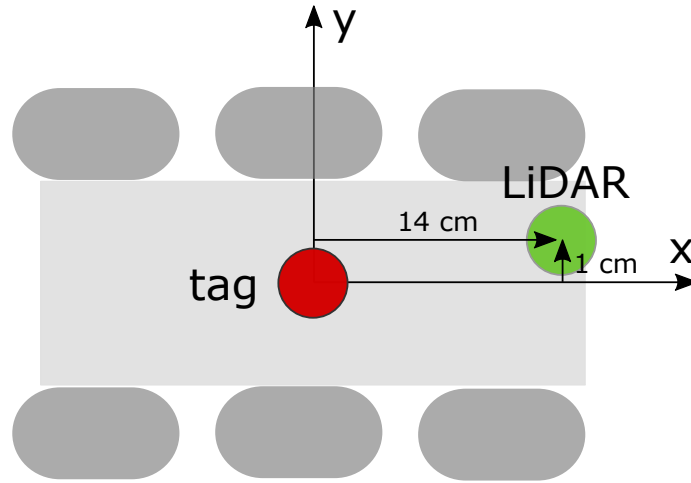


Fig. 4. The top-view schematic of the platform with Lidar and sensor displacements

The robot test files were processed in the same manner as calibration files. Please note that the identifiers of the anchors are changed after preprocessing from [3,4,5,8,10,11] to [1,2,3,4,5,6] respectively.

## V. TESTING DATA

The dataset requires two test measurements: one taken with a robot and one containing RSS measurements registered in a walking person localization. During the second experiment the person walked along the reference trajectory composed of straight lines. The coordinates of the trajectory vertices can be found in `tests/test_walk_df.csv` file. Each of the file's lines contains the coordinates of the line starting and ending points in the following order :  $x_0, y_0, x_1, y_1$ .

## VI. LICENCE AND ATTRIBUTION

The dataset is licensed under *Creative Commons Attribution 4.0 International* licence. The actual citation data can be found in the metadata on Zenodo or Readme file on Github.

## REFERENCES

- [1] J. Kolakowski, V. Djaja-Josko, M. Kolakowski, and K. Broczek, "UWB/BLE Tracking System for Elderly People Monitoring," *Sensors*, vol. 20, no. 6, p. 1574, Mar. 2020, doi: 10.3390/s20061574.
- [2] Marcin Kolakowski, Simple Raspberry Pi, Python based controlled for a robotic platform. 2021. Accessed: Aug. 12, 2021. [Online]. Available: <https://github.com/marckolak/wtController>