Non-invasive Localization using Software-Defined Radios

This “ReadMe.pdf” file was generated on 2022-04-19 by Mr. Muhammad Zakir Khan

# GENERAL INFORMATION

## **Title of Dataset**

Non-invasive Localization using Software-Defined Radios

## **Author Information**

Names: Mr. Muhammad Zakir Khan, Dr. Ahmad Taha, Mr. William Taylor, Professor Muhammad Ali Imran, and Dr. Qammer Abbasi

Institution: University of Glasgow

**Address:** James Watt School of Engineering, College of Science and Engineering, University of Glasgow, Glasgow, G12 8QQ

**Emails:** [m.khan.6@research.gla.ac.uk](mailto:m.khan.6@research.gla.ac.uk), [Ahmad.Taha@glasgow.ac.uk](mailto:Ahmad.Taha@glasgow.ac.uk), [w.taylor.2@research.gla.ac.uk](mailto:w.taylor.2@research.gla.ac.uk), [Muhammad.Imran@glasgow.ac.uk](mailto:Muhammad.Imran@glasgow.ac.uk), [Qammer.Abbasi@glasgow.ac.uk](mailto:Qammer.Abbasi@glasgow.ac.uk)

## **Date of data collection:**

The data was collected over the first week of August 2021.

## The geographic location of data collection:

Level 5 and Room 521, James Watt South Building, Glasgow, G12 8QQ, United Kingdom

## Information about funding sources that supported the collection of the data:

This work is supported in parts by Engineering and Physical Sciences Research Council (EPSRC) EP/T021020/1 and EP/T021063/1. Muhammad Zakir Khan’s Ph.D. is funded by Begum Akhter Rukhsana Memorial Trust (BARMT) Foreign Scholarship Pakistan.

# **SHARING/ACCESS INFORMATION**

## Licenses/restrictions placed on the data:

NA

## Links to other publicly accessible locations of the data:

NA

## Was data derived from another source?

No

Recommended citation for this dataset: Khan, M. Z., Taha, A., Taylor, W., Imran, M. A., & Abbasi, Q. H. (2022). Non-Invasive Localisation Using Software-Defined Radios. *IEEE Sensors Journal*.

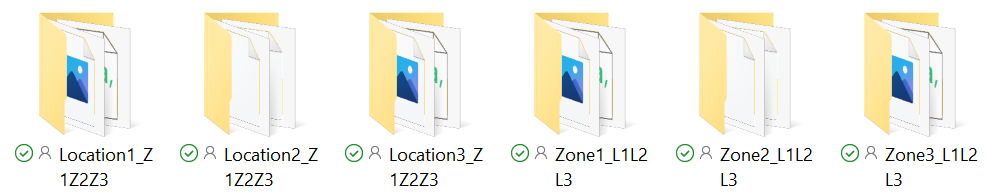
# **DATA & FILE OVERVIEW**

## **Details of Data Folders and Files**

The dataset was divided into 43 classes, that is, a total of 4300 data samples/files, each representing a particular number of subjects and activities where 43 classes are organized shown in Table 1. The main data folder is subdivided into 6 folders (see Figure 1 and Table 2).

Table 1: The data classes and their description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Class | Class Description | No of Classes | Count |
| 1 | Empty Activity | No Human subject in the activity area | 1 | 100 |
| 2 | No Activity | No activity performed by human | 9 | 900 |
| 3 | Sitting | The action of “Sitting” at the designated location within Zone | 9 | 900 |
| 4 | Standing | The action of “Standing” at the designated location with Zone | 9 | 900 |
| 5 | Leaning forward | Leaning forward with the upper body | 9 | 900 |
| 6 | Walking Rx-Tx and Tx-Rx | Walking from the USRP X310 Rx side to USRP X300 Tx side and vice versa | 3\*2 = 6 | 600 |



A screenshot of a computer

Description automatically generated with medium confidence

*Figure 1 Data folder Structure*

*Table 2 Details of the Data Set (Folders, Files, Description, and Number of Samples)*

|  |  |  |  |
| --- | --- | --- | --- |
| **Folder Name** | **Class/File Name** | **Description** | **Number of Samples per**  **Class** |
| **Location1\_Z1Z2Z3** | EmptyRoom | Empty activity area without any test subjects | 100 |
| LeaningL1Z1 | Subject performing the action of "Leaning" at location 1 in Zone 1 | 100 |
| LeaningL1Z2 | Subject performing the action of "Leaning" at location 1 in Zone 2 | 100 |
| LeaningL1Z3 | Subject performing the action of "Leaning" at location 1 in Zone 3 | 100 |
| NoActivityL1Z1 | Subject performing the action of "No activity" at location 1 in Zone 1 | 100 |
| NoActivityL1Z2 | Subject performing the action of "No activity" at location 1 in Zone 2 | 100 |
| NoActivityL1Z3 | Subject performing the action of "No activity" at location 1 in Zone 3 | 100 |
| SittingL1Z1 | Subject performing the action of "Sitting" at location 1 in Zone 1 | 100 |
| SittingL1Z2 | Subject performing the action of "Sitting" at location 1 in Zone 2 | 100 |
| SittingL1Z3 | Subject performing the action of "Sitting" at location 1 in Zone 3 | 100 |
| StandingL1Z1 | Subject performing the action of "Standing" at location 1 in Zone 1 | 100 |
| StandingL1Z2 | Subject performing the action of "Standing" at location 1 in Zone 2 | 100 |
| StandingL1Z3 | Subject performing the action of "Standing" at location 1 in Zone 3 | 100 |
| WalkingRxTxZ1 | Subject performing the action of "Walking" from Rx toward Tx in Zone 1 | 100 |
| WalkingTxRxZ1 | Subject performing the action of "Walking" from Tx toward Rx in Zone 1 | 100 |
| **Zone1\_L1L2L3** | EmptyRoom | Empty activity area without any test subjects | 100 |
| LeaningL1Z1 | Subject performing the action of "Leaning" at location 1 in Zone 1 | 100 |
| LeaningL2Z1 | Subject performing the action of "Leaning" at location 2 in Zone 1 | 100 |
| LeaningL3Z1 | Subject performing the action of "Leaning" at location 3 in Zone 1 | 100 |
| NoActivityL1Z1 | Subject performing the action of "No activity" at location 1 in Zone 1 | 100 |
| NoActivityL2Z1 | Subject performing the action of "No activity" at location 2 in Zone 1 | 100 |
| NoActivityL3Z1 | Subject performing the action of "No activity" at location 3 in Zone 1 | 100 |
| SittingL1Z1 | Subject performing the action of "Sitting" at location 1 in Zone 1 | 100 |
| SittingL2Z1 | Subject performing the action of "Sitting" at location 2 in Zone 1 | 100 |
| SittingL3Z1 | Subject performing the action of "Sitting" at location 3 in Zone 1 | 100 |
| StandingL1Z1 | Subject performing the action of "Standing" at location 1 in Zone 1 | 100 |
| StandingL2Z1 | Subject performing the action of "Standing" at location 2 in Zone 1 | 100 |
| StandingL3Z1 | Subject performing the action of "Standing" at location 3 in Zone 1 | 100 |
| WalkingRxTxZ1 | Subject performing the action of "Walking" from Rx toward Tx in Zone 1 | 100 |
| WalkingTxRxZ1 | Subject performing the action of "Walking" from Tx toward Rx in Zone 1 | 100 |

**METHODOLOGICAL INFORMATION**

## Description of methods used for collection/generation of data:

The dataset represents a combination of activities captured through wireless channel state information, using two USRP X300/X310 devices each equipped with the VERT2450 omnidirectional antenna and Horns. One USRP (X300) was used as the transmitter and the second USRP (X310) was used for the receiver. Each USRP was connected to an All-in-One PC that uses an Intel(R) Core (TM) i7-7700 3.60 GHz processor and 16 GB RAM. The system made use of virtual machines to provide the Ubuntu 16.04 operating system. On the Ubuntu virtual machines, Gnu Radio was used to communicate with the USRP devices, and it also hosted the artificial intelligence engine. Figure 2 show the experimental setup which was used to collect the data.

Graphical user interface

Description automatically generated

*Figure 2 Experimental Setup*

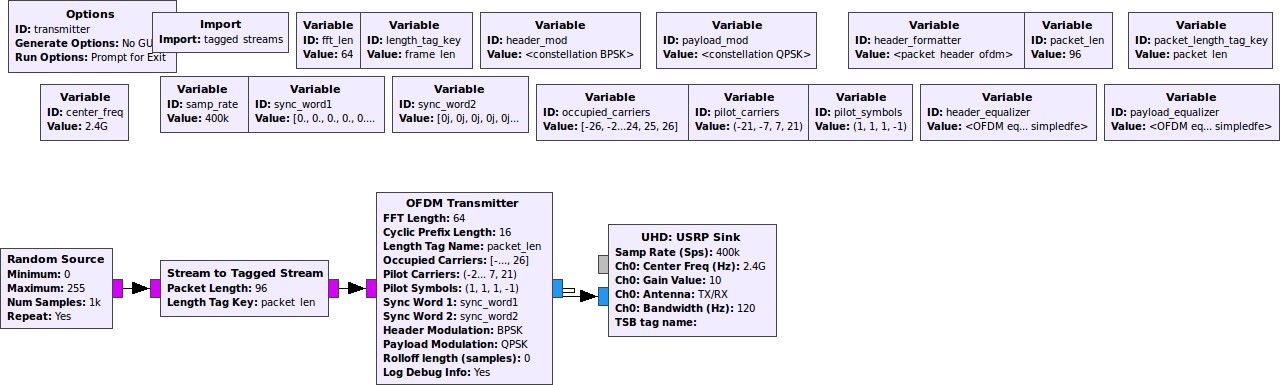
## **Methods for processing the data:**

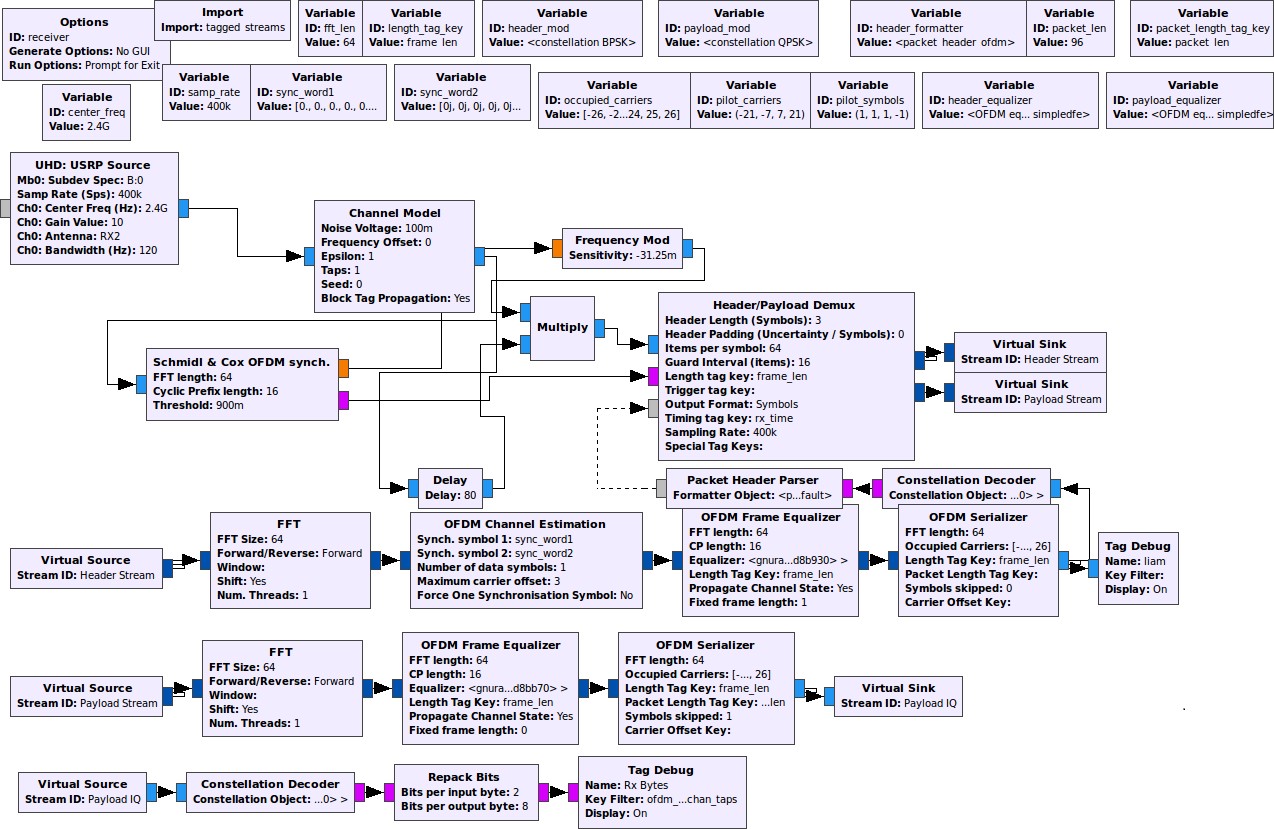
Firstly, the USRP transmitter and receiver devices were configured to communicate together using the GNU radio python package to set parameters such as central frequency, number of Orthogonal Frequency Division Multiplexing (OFDM) subcarriers, and power levels (see Table 2).

*Table 2 System Parameters*

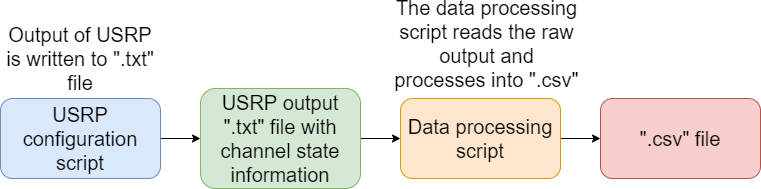
|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Central Frequency | 3.75 GHz |
| OFDM subcarriers | 64 |
| Transmitter Gain (dB) | 70 |
| Receiver Gain (dB) | 50 |

GNU Radio is a free and open-source software which is used in research for software-defined radios and signal processing. GNU Radio comes with examples of OFDM signal processing where the channel state information can be extracted. This example is modified to include the USRP as the transmitting and receiving devices. The GNU Radio software publishes the configuration in the format of a flow diagram which can be used to set up the blocks of the USRP and OFDM communication. Figure 3 shows the GNU Radio flow diagram that depicts the configurations of the USRP devices. The flow diagram can then be converted into a python script, which can be executed to begin OFDM communication. The raw form of the channel state information output is written to text files which are then converted to a processable format (see Figure 4).





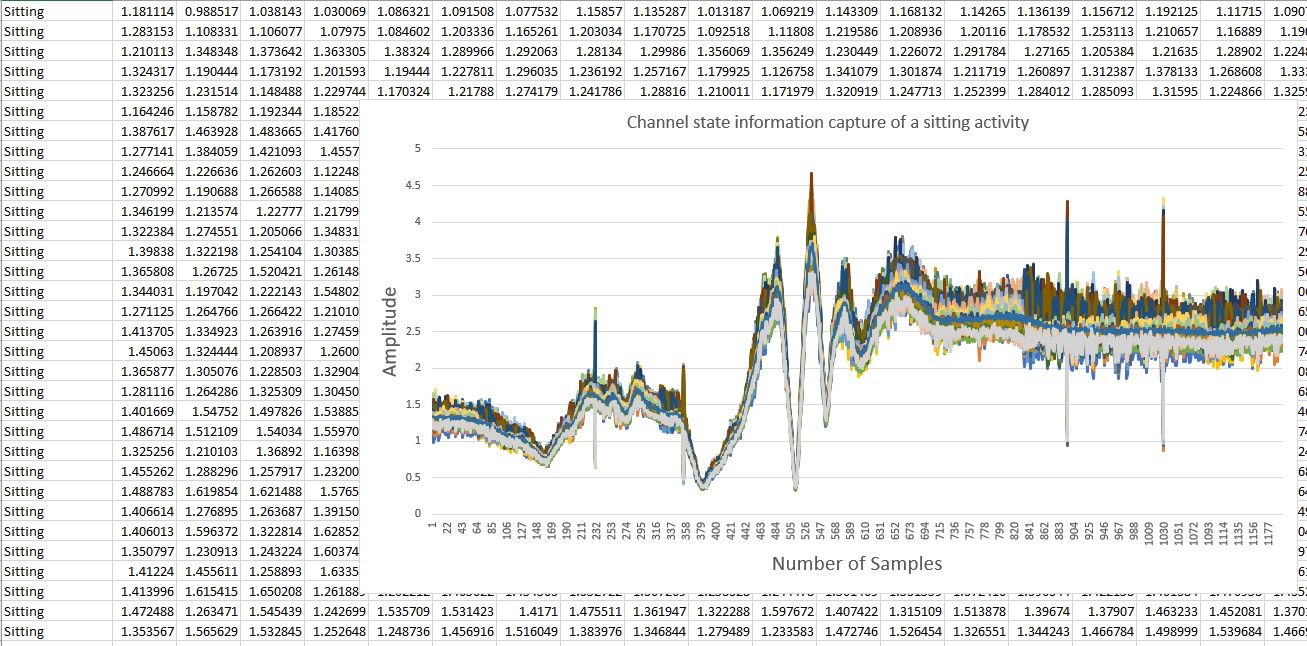
*Figure 3 GNU Radio Flow Diagram*



*Figure 4 Data flow in the data collection stage*

Secondly, the task was to collect the channel state information and create data sets from them in the form of “Comma Separated Value” (.csv) files. The CSV files would hold the data sets that will be used for training and testing of the Machine Learning (ML) algorithm. For this, another python script is used to process the terminal output and filter out the channel state information complex numbers. Python

carries out mathematical functions to calculate the amplitude of the RF signal from the channel state information complex numbers. The amplitude values are then saved to CSV format for ML and to visualise the signal propagation through line graphs, see an example in Figure 5. The above process was repeated for all the data files in this data set.



*Figure 5 Channel state information capture of “Sitting” activity in ".csv" format and the*

*corresponding plot*

## Instrument- or software-specific information needed to interpret the data:

Data files are all in “.csv” format which could be opened using Microsoft Excel and further processed using python.