Tools for modelling distance estimation based on RSSI

D Dobrilovic[1], Z Stojanov[1], J Stojanov[1], M Malic[2]

[1] University of Novi Sad, Technical Faculty “Mihajlo Pupin”, Zrenjanin, Serbia
[2] Panonit, Novi Sad, Serbia

e-mail: dalibor.dobrilovic@tfzr.rs; dalibor.dobrilovic@uns.ac.rs

Abstract. The systems for localization of resources in indoor environments based on Received Signal Strength Indicator (RSSI) are widely used today. Since satellite navigation systems, such as GPS or GLONASS, have certain difficulties in the indoor environments, the signals of deployed wireless devices, such as sensor nodes, access points etc, are used for localization instead. Those systems are known as Indoor Positioning System (IPS). Those systems are used for resource tracking and positioning in places such as airports, railway stations, shopping malls, warehouses, production facilities, construction sites, and healthcare institutions. The Bluetooth Low Energy is one of the wireless technologies that can be used with great efficiency for indoor localization. It offers easy and economic implementation on mobile devices such as smart phones and tablets. There are many techniques used for determination of position. In a number of methods, such as ROCRSSI or MinMax, the distance from the wireless nodes is used for calculating the location. In those systems the main challenge is to accurately estimate distance from the device based on signal strength. In this paper, usability of various software tools for modelling the distance estimation based on RSSI is discussed. Those software tools are Microsoft Access, R Studio, Octave, and Python.

1. Introduction

Localization in indoor environments requires a different approach comparing to the outdoor positioning systems. The reason is that satellite navigation systems, such as GPS or GLONASS, require line of sight and in covered areas can have weakened or no signal which results with inaccurate operation. In Indoor Positioning System (IPS) signals from various wireless devices such as wireless access points or wireless sensor nodes are used to determine the location. There are a number of techniques for localization, such as: Time of Arrival (ToA) or Time Difference of Arrival (TDoA), and Angle of Arrival (AoA) or Direction of Arrival (DoA) [1, 2], Time of Flight (ToF), Roundtrip Time of Flight (RToF), Phase of Arrival (PoA), and Phase Difference of Arrival (PDoA) [3, 4, 5]. Some techniques use Received Signal Strength Indicator (RSSI) for localization. The RSSI based localization techniques are ROCRSSI [6] and MinMax [7] among the others.

In [8] is presented an overview of techniques that can be used for localization in wireless sensor networks (WSN). Sensor networks vary significantly from traditional cellular networks to Wi-Fi. Generally speaking, the most common technologies that can be used in these systems are Wi-Fi and Bluetooth Low Energy (BLE). In IPS systems the main challenge is to accurately estimate distance from the device based on RSSI. The IPS systems has its appliance in airports, railway stations, shopping malls, production facilities, construction sites, warehouses, hospitals, and other healthcare institutions, etc. Those systems can be used to track human and material resources, e.g. employees, patients, healthcare workers or vehicles such as forklifts in large warehouses.

In this paper the authors discuss about usability of various software tools for modelling the distance estimation based on RSSI. Those software tools are: Microsoft Access, R Studio, Octave, and Python. Although, there is a number of other tools that can be used with the similar efficiency for the given tasks such as MathCAD, Statistica, etc., for this research four enlisted tools are selected. Microsoft Excel is chosen because of the large base of potential users and the simplicity of the fitting tools. GNU Octave is popular in academic community and Python is popular both in academic and commercial programming community. Both tools are part of the several courses in the curricula at our institution, and both stuff and students are familiar with their usage. R is chosen, despite its recent lost of popularity in the favour of Python, because it still has a large developing community and because of large quantity of available examples of various curve fitting models. For the model design and for its evaluation the real data from field measurement are used. The system used for data collection is based on open-source hardware and Bluetooth Low Energy (BLE) technology.

The paper is structured as follows. After the introduction and the brief explanation of IPS usage and related work, the experiment and data sets collection are described. Next, the software tools used for curve fitting, distance estimation model, its evaluation and analyses are presented. Finally, the findings and further research directions are discussed.

2. Related work

2.1. Appliance of indoor positioning systems

The appliance of indoor positioning systems can be very wide. In the research [9] is introduced a new prototype framework for automated tracking of construction materials on the job site. A hybrid sensor network based on ZigBee technology with mesh network topology can increase location accuracy by eliminating multipath property of radio signal propagation and can be suitable for implementation in complex construction sites in an energy-efficient way [9].

Nowadays, very popular implementation of localisation systems is in healthcare, especially in the time of ongoing pandemics. Bluetooth based approach for tracking healthcare providers in the emergency room is described in [10]. The goal of this research was to provide an approach for reliably collecting the door to physician (DPT) time automatically with accurate usage of sensor-based technology. Since emergency rooms are commonly crowded, performance measures are needed to gauge the efficiency, The usage of proposed solution can lead to improvements in clinical workflow, including staff reorganization, reduce ED crowding and improve patient safety in the long run.

Monitoring the locations and paths that individual’s traverse in an indoor environment has become an important element for analysing the workflow in a clinical environment and for modelling the spread of infectious diseases/hospital acquired-infections [11]. In this research is presented a novel method to detect the location of a moving user in indoor and dynamic environments. The received number of signals indicator (RNSI) is introduced as a new measurement and a practical method for removing signals that could lead to incorrect location prediction is presented.

In [12] is presented the use of wearable proximity sensors to measure the numbers and durations of contacts between individuals in an acute care geriatric unit at university hospital. The system is used to document the individuals with a high number of contacts, who could be considered as potential super-spreaders of infections. In that way systems can be used in prevention and control of hospital-acquired infections.

2.2. Bluetooth Low Energy used for indoor localization

Since this paper is focused on the usage of Bluetooth Low Energy (BLE) technology, the various implementation of indoor positioning systems based on BLE will be presented in this subsection. The research presented in [13] describes indoor localization based on BLE devices deployed under the WLAN (Wireless LAN) environment. In [14] the extensive set of experiments to quantify the impact of various indoor obstacles on the BLE signal variance is made. It is shown that different iBeacon devices from different vendors along with the paired mobile device platform (e.g., iOS or Android) can give significant impact on the RSSI measurements. This complicates the process of designing a generally acceptable distance/location estimation model. Furthermore, practical factors such as the deployment height or environmental factors also introduce a significant amount of impact on the RSSI values to introduce an additional level of complexity in designing highly accurate location estimation systems using iBeacons. The other approach in achieving localization using smartphone sensors is based on Pedestrian Dead Reckoning (PDR) [15]. The basic idea presented in this paper is to derive the current location based on the previous location, the current walking length and walking direction. This paper presents a new framework of combining PDR and iBeacon. The iBeacon technique is built upon Bluetooth Low Energy (BLE) version 4.0, thus, it is very energy efficient and can be utilized for localization based on Received Signal Strength (RSSI) of BLE devices such as smartphone.

The focus of the paper [16] is a iBeacon technology which provides a higher level of location awareness in indoor positioning. It is a built-in, cross-platform technology for Android and iOS devices, which utilizes Bluetooth Low Energy (BLE) as technology with significant advantages compared to other types of indoor positioning technologies. These advantages are less expensive hardware, less energy consumption, needless to internet connection, and capability of receiving notifications in background. Another research with similar focus is presented in [17] where the localisation system based on iBeacon technology is presented.

2.3. Distance estimation based on RSSI

In order to estimate distances based on RSSI values, RF propagation models such as model shown in (1) can be used [14, 18, 19]. The model is used to convert RSSI readings into distance [14].

$$RSSI = R_0 - 10 \cdot \gamma \cdot log\left(\frac{d}{d_0}\right) \Rightarrow \qquad (1)$$
$$d = 10^{(R^0 - RSSI)/10 \cdot \gamma}$$

Since this model is general, it should be fitted to data sets in order to achieve the highest accuracy in accordance to data sets used in experiment. The formula for distance calculation given in (1) should be used with predefined values for R_0 and γ. R_0 stands for RSSI value calculated on reference distance d_o, usually 1m. Alternatively, for the calculation of R_0 the formula for open space propagation loss can be used. Open space propagation formula is used in this paper for calculating the referential model parameters used in the evaluation presented the next sections. The value of parameter γ (path loss distance exponent) can be used as one of proposed values for different environments and frequencies obtained from empirical researches [18, 19]. In this paper the value of γ is calculated by the best fit with the data used as data sets. The values of R_0 and γ will be presented in evaluation section.

Besides, propagation loss model presented in equation (1) other indoor propagation loss models such as ITU indoor propagation model [19], can be used as well. For this research only the fitted model (1) is used because its high accuracy and the authors experiences in previous researches.

3. The experiment and data sets

The experiment is made in institutional building on the 2nd floor. The layout of the 2nd floor, BLE transmitting and receiving devices position and experiment details are presented in greater detail in [20, 21]. The platform for the experiment described in this paper is based on open source hardware. The central unit of the platform is microcontroller board Arduino/Genuino UNO Rev3. Arduino/Genuino is open-source hardware platform suitable for experiments and implementation of indoor positioning systems, not only for BLE, but for the other technologies as well [22]. This microcontroller is connected to Bluetooth Low Energy communication module AT-09 based on Texas Instruments CC2541 chip with the receiver sensitivity of -90 dBm for 2 Mbps GFSK and -94 dBm for 1 Mbps GFSK respectively.

The data set used in this research is collected during experiments presented in [20] and are presented in Table 1.

Table 1. Data sets used in this research.

Data Set No.	Distance (m)	RSSI (dBm)	Data Set No.	Distance (m)	RSSI (dBm)
1	4	-81.1572	12	11.8	-88.3324
2	4.3	-79.1939	13	12.01	-89.9201
3	4.65	-71.128	14	13.43	-91.3241
4	6.43	-78.5308	15	13.64	-89.1103
5	6.63	-79.109	16	13.71	-88.1114
6	7.2	-88.8033	17	13.97	-86.3636
7	8.17	-81.7438	18	14.02	-79.5302
8	8.23	-77.6223	19	15.65	-89.0563
9	8.27	-85.0958	20	15.79	-92.0534
10	8.8	-85.0405	21	16.42	-91.0504
11	8.88	-87.372	22	19.71	-86.7351

4. The software tools

In this section software tools used for distance estimation are described. Those software tools are Microsoft Excel, R programming language with RStudio, GNU Octave, and Python programming language with its numerical libraries.

4.1. Microsoft Excel

Microsoft Excel [23] is widely used spreadsheet software developed by Microsoft. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. Excel forms part of the Microsoft Office suite of software. The software has pre programmed features that allow user to find the best fitting equation for a data set for a select number of functions:

- Linear model,
- Exponential model,
- Polynomial model,
- Logarithmic model and
- Power model.

The process of finding the best fitting equation is simple. After the selection of data sets to fit (in this case data set contains distance and RSSI values) the option for Scatter graph (*Insert* > *Scatter*) from the Menu should be selected. After that, right click on the data points presented on the graph will show menu with the option *Add Trendline*. After selecting the option, the fitting function should be selected between following options: Polynomial (order 2), Exponential or Linear, Also, the option *Display Equation on chart* should be checked in order to display the fitting function on the graph. After applying the Trendline settings the fitting equation is displayed on the graph as it is shown in Figure 1.

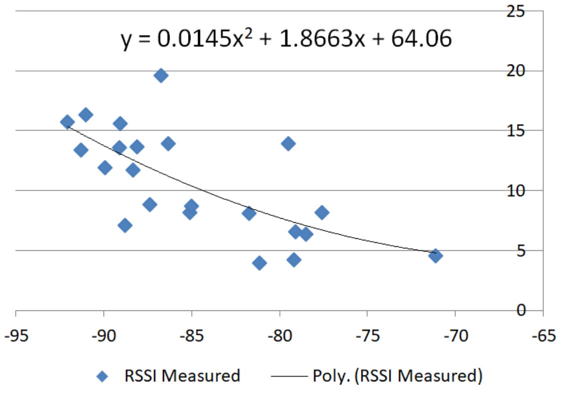

Figure 1. Excel scatter graph with polynomial trend line.

The *Add Trendline* option gives following fitting equations for the data set used in this example:

Polynomial order 2:

$$d_{excelPol2} = 0.0145 \bullet x^2 + 1.8663 \bullet x + 64.06 \qquad (2)$$

Exponential:

$$d_{excelExp} = 0.0627 \bullet e^{-0.06x} \qquad (3)$$

And linear:

$$dist_{excelLin} = -0.543 \bullet x - 35.355 \qquad (4)$$

4.2. R programming language

R [24, 25] is both a language and environment for statistical computing and graphics. It is a GNU project similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies). R may be considered as a different implementation of S with some differences. R is designed to provide a variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques, and is highly extensible.

R can be used with RStudio, an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging, and workspace management. In the following listing (Listing 1) are presented values for the polynomial fittings order 1 (linear) and 2 respectively, and the graph of fitting curves is presented in Figure 2:

```
> fit
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
    -35.355       -0.543

> fit2
Call:
lm(formula = y ~ poly(x, 2, raw = TRUE))

Coefficients:
            (Intercept)  poly(x, 2, raw = TRUE)1  poly(x, 2, raw = TRUE)2
               64.06035                  1.86630                  0.01453
```

Listing 1. Calculated Polynomial fitting equations

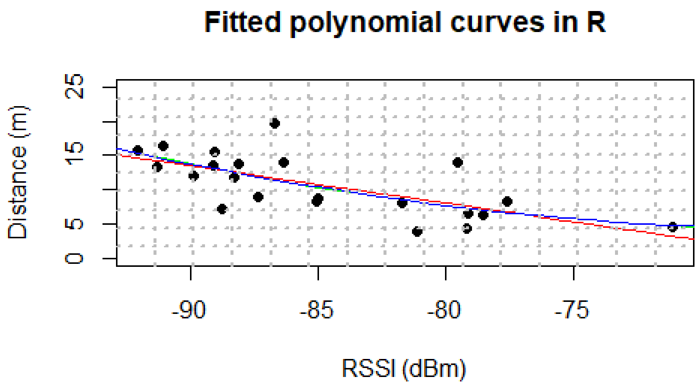

Figure 2. Fitted polynomial curves in R.

According to the Listing 1 R gives following fitting equations for the data set used in this example:

Polynomial order 1 (linear):

$$d_{RPol1} = -0.543 \cdot x - 35.355 \qquad (5)$$

Polynomial order 2:

$$d_{RPol2} = 0.01453 \cdot x^2 + 1.86630 \cdot x + 64.06035 \qquad (6)$$

As one can see, fitting equations are similar to the equations given by Microsoft Excel.

4.3. GNU Octave

GNU Octave [26] is a high-level language, primarily designed for numerical computations. It has command line interface and it can be used for solving linear and nonlinear problems numerically, and for performing other numerical experiments. The Octave syntax is largely compatible with Matlab and therefore the Octave scripts can be used in Matlab with little or no modifications. GNU Octave has extensive tools for solving common numerical linear algebra problems and can be used for finding the roots of nonlinear equations, integrating ordinary functions, manipulating polynomials, and integrating ordinary differential and differential-algebraic equations. Besides its own programming language, GNU Octave supports usage of dynamically loaded modules written in C++, C, Fortran, or other languages.

The *polyfit()* function in Octave is used for polynomial fitting. The usage of this function for polynomial fitting order 1 (linear) and order 2 is presented in Listing 2.

```
p = polyfit(xdata,ydata,1)

p =

   -0.54304  -35.35478

p = polyfit(xdata,ydata,2)

p =

     0.014528     1.866295    64.060348
```

Listing 2. Calculated Polynomial fitting equation parameters

The fitting of polynomial function order 2 is presented in Figure 3 together with data set.

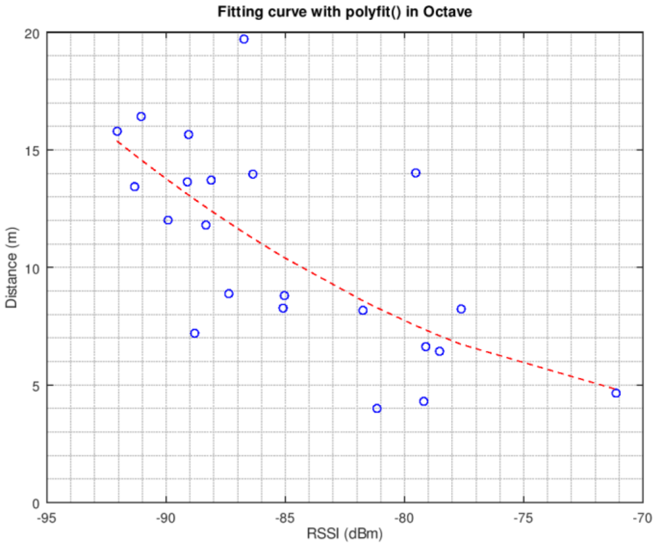

Figure 3. Polynomial fitting in Octave.

4.4. Python programming language

The Python [27, 28] is programming language with extensive usage in scientific programming [29]. Core numeric libraries for Python are:

- Numpy [30, 31], designed for numerical computing with powerful numerical arrays objects, and routines to manipulate them (http://www.numpy.org/)
- Scipy [27] with high-level numerical routines. Designed for optimization, regression, interpolation, etc (http://www.scipy.org/)
- Matplotlib [32], designed for 2-D visualization and plotting (http://matplotlib.org/)

The example of using Python for logarithmic fitting is given in the Figure 4, The Figure 4 show data sets used for fitting, as well as fitting curve and the equation parameters. The equation calculated with the Python is given in (7).

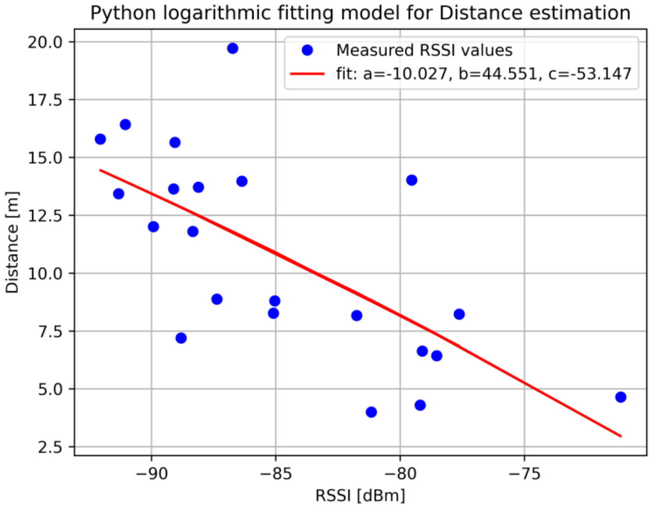

Figure 4. Logarithmic fitting in Python.

$$d_{logPython} = -10.027 + \ 44.551 \cdot log_2(\frac{-RSSI}{53.147}) \qquad (7)$$

5. Result evaluation and analyses

In order to evaluate the fitting equations the results calculated with the usage of described software tools are compared with measured results. The Root Mean Square Error (RMSE) is used to compare the accuracy of fitted models. For the evaluation equations (1), (2), (3), (4), and (7) are used. It is important to note that equation (1) is fitted to the experimental data sets. The fitting is made in Octave for values of R_0 and γ. The fitted value for γ is 2.5 and for the R_0 reference value is -62.37 dBm (value is calculated with free space propagation formula). The process of fitting value of γ is done with calculating RMSE for each value of γ between 1 and 4 (step 0.1). The value of 2.5 is chosen because of lowest RMSE value, comparing to others.

Results of evaluation are presented in Figure 5 and in Table 2.

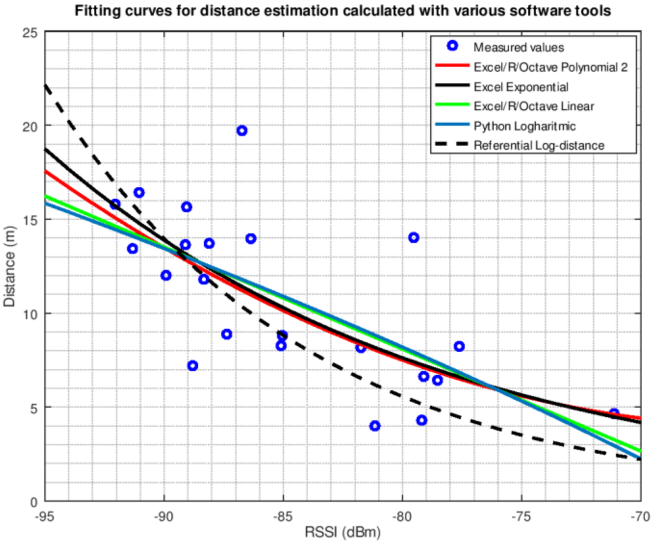

Figure 5. Comparison of fitting curves and measured RSSI data.

Table 2 shows that the most accurate are polynomial order 2 and exponential models. The referential model presented in [14] is the least accurate of all compared models. The polynomial order 2 and exponential models comparing to the reference model are more accurate for about 0.89 percent.

Table 2. RMSE values for the evaluated models.

Model	RMSE [m]
Reference equation	3.4893
Polynomial Order 2	3.1373
Exponential	3.1305
Linear	3.1606
Logharitmic	3.1783

Generally, there is no significant difference in the results, since the mean square error in distance estimation ranges from 3.1 to 3.5 meters. The error in distance is too high for efficient application in indoor localisation systems, but the assumption is that the difference is resulted not from fitting model inaccuracy, but from specific environment where RSSI measurements took place [20, 21]. Some ongoing experimental measurements in the same environment acquire data sets with higher level of fitting to the model, so the model will be evaluated in the future with new data sets.

6. Conclusion and further work

The presented results showed that popular software tools, presented in this paper, can be used efficiently for model fitting in order to estimate distance for the purpose of indoor localization systems. The mean square error of the models ranges from 3.1 to 3.5 meters with the highest accuracy achieved with the polynomial and exponential model. The difference in estimated and real distances is not affected with the model accuracy, but with the data sets collected in the experiment.

The new experimental settings as well as RSSI measuring results should be used in near future for the further exploration of the usability of fitting models calculated with the software tools. In this research primarily the embedded fitting functions are used. Because of that, in the next phase of the research, and in order to obtain more accurate models, various existing fitting models used in different fields can be used for distance estimation.

Finally, the research aimed at determining which model is the easiest and the most suitable for implementation on the devices used within localization system should be one of future research directions.

Acknowledgments

Ministry of Education, Science and Technological Development of the Republic of Serbia, supports this research under the project ”The development of software tools for business process analysis and improvement”, project number TR32044.

References

[1] Brchan J L, Zhao L, Wu J, Williams R E and Pérez L C 2012 A real-time RFID localization experiment using propagation models, *Proc. of 2012 IEEE International Conference on RFID (RFID)*, Orlando, FL, p 141, doi: 10.1109/RFID.2012.6193042.

[2] Deak G, Curran K and Condell J 2012 A survey of active and passive indoor localisation systems. *Computer Communications*, 35(16), p 1939. doi: 10.1016/j.comcom.2012.06.004.

[3] de San Bernabé A, Martinez-de Dios J and Ollero A 2017 Efficient integration of RSSI for tracking using Wireless Camera Networks. *Information Fusion*, 36, p 296. doi: 10.1016/j.inffus.2016.11.001.

[4] Liu C, Scott T, Wu K and Hoffman D 2007 Range-free sensor localisation with ring overlapping based on comparison of received signal strength indicator, *International Journal of Sensor Networks*, 2(5-6), p 399.

[5] Nikitin P V, Martinez R, Ramamurthy S, Leland H, Spiess G and Rao K V S 2010 Phase based spatial identification of UHF RFID tags, *Proc.of 2010 IEEE International Conference on RFID (IEEE RFID 2010)*, Orlando, FL, 02-109, doi: 10.1109/RFID.2010.5467253.

[6] Wu K, Liu C, Pan J and Huang D 2007 Robust range-free localization in wireless sensor networks. *Mobile Networks and Applications*, 12(5), p 392.

[7] Langendoen K and Reijers N 2003 Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4), p 499.

[8] Mao G, Fidan B and Anderson B D 2007 Wireless sensor network localization techniques. *Computer Networks*, 51(10), p 2529. doi: 10.1016/j.comnet.2006.11.018.

[9] Skibniewski M J and Jang W-S 2007 Localization technique for automated tracking of construction materials utilizing combined RF and ultrasound sensor interfaces, *Proc.of ASCE International Workshop on Computing in Civil Engineering*, Pittsburgh, PA. USA.

[10] Frisby J, Smith V, Traub S and Patel V L 2017 Contextual Computing: a Bluetooth based approach for tracking healthcare providers in the emergency room, *J. Biomed. Inform*. 65 p 97, https://doi.org/10.1016/j.jbi.2016.11.008.

[11] Suriana D, Kima V, Menonb R, Dunna A G, Sintchenkob V and Coieraa E 2019 Tracking a moving user in indoor environments using Bluetooth low energy beacons, *Journal of Biomedical Informatics* 98, p 1.

[12] Vanhems P, Barrat A, Cattuto C, Pinton J-F, Khanafer N, Régis C, et al. 2013 Estimating potential infection transmission routes in hospital wards using wearable proximity sensors, *PLoS ONE 8*, e73970, , https://doi.org/10.1371/journal.pone.0073970

[13] Huh J H and Seo K 2017 An Indoor Location-Based Control System Using Bluetooth Beacons for IoT Systems, *Sensors (Basel)*, 17(12), p 2917. Published 2017 Dec 19. doi:10.3390/s17122917

[14] Paek J, Ko J and Shin H 2016 A Measurement Study of BLE iBeacon and Geometric Adjustment Scheme for Indoor Location-Based Mobile Applications, *Self-Organization in Mobile Networking Systems*, Vol. 2016, https://doi.org/10.1155/2016/8367638

[15] Chen Z, Zhu Q, Jiang H and Soh Y C 2015 Indoor localization using smartphone sensors and iBeacons, *Proc. of the IEEE 10th Conference on Industrial Electronics and Applications (ICIEA '15)*, p 1723, IEEE, Auckland, New Zealand, June 2015.

[16] Fard H K, Chen Y and Son K K 2015 Indoor positioning of mobile devices with agile iBeacon deployment, *Proc. of IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE '15)*, p 275, Halifax, Canada, May 2015.

[17] Kouhne M and Sieck J 2014 Location-based services with ibeacon technology, *Proc. of 2nd IEEE International Conference on Artificial Intelligence, Modelling, and Simulation (AIMS '14)*, p 315, IEEE, Madrid, Spain, November 2014.

[18] Rappaport T S 2002 *Wireless communications principles and practices*, Prentice-Hall

[19] Seybold J S 2005 *Introduction to RF propagation*, Wiley and Sons.

[20] Dobrilović D Malić M Malić D and Šinik V 2018, Methodology for Bluetooth Low Energy Performance Analyses Based on Open-source Hardware, *Proc. Of International Conference on Applied Internet and Information Technologies 2018 (Bitola, Northern Macedonia)*, p 116.

[21] Dobrilović D Petrović D and Malić M 2018 Usability of open-source hardware based platform for indoor positioning systems, *Journal of engineering management and competitiveness (JEMC)*, Vol. 8, No. 2, p 113.

[22] Barai S, Biswas D and Sau B 2017 Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique, *Proc. of 2017 IEEE Conference on Antenna Measurements & Applications (CAMA)*, p 170, doi:10.1109/cama.2017.8273392

[23] Microsoft Corporation 2018 Microsoft Excel, Retrieved from https://office.microsoft.com/excel

[24] R Core Team 2019 R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[25] Ritz C, Baty F, Streibig J C and Gerhard D 2015 Dose-Response Analysis Using R PLOS ONE, 10 (12), e0146021

[26] Eaton J W, Bateman D, Hauberg S and Wehbring R 2018 GNU Octave version 4.4.1 manual: a high-level interactive language for numerical computations, URL https://www.gnu.org/software/octave/doc/v4.4.1/

[27] Jones E, Oliphant T and Peterson P et al. 2001 SciPy: Open source scientific tools for Python. Retrieved from "http://www.scipy.org/

[28] Oliphant T E 2007 Python for Scientific Computing, Computing in Science & Engineering, 9, 10-20 (2007), DOI:10.1109/MCSE.2007.58

[29] Millman K J and Aivazis M 2006 Python for Scientists and Engineers, Computing in Science & Engineering, 13, 9-12 (2011), DOI:10.1109/MCSE.2011.36

[30] Travis E and Oliphant A guide to NumPy, USA: Trelgol Publishing.

[31] van der Walt S, Chris Colbert S C and Varoquaux G 2011 The NumPy Array: A Structure for Efficient Numerical Computation, *Computing in Science & Engineering*, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37

[32] Hunter J D 2007 Matplotlib: A 2D Graphics Environment, *Computing in Science & Engineering*, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55