Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique

Conference Paper · December 2017

DOI: 10.1109/CAMA.2017.8273392

3 authors, including:

Suvankar Barai
Jadavpur University

Buddhadeb Sau
Jadavpur University

Estimate Distance Measurement using NodeMCU ESP8266 based on RSSI Technique

Suvankar Barai[1], Debajyoti Biswas[2] and Buddhadeb Sau[3]
Department of Mathematics, Jadavpur University, Kolkata- 700032, India
E-mail: [1]suvankar.barai@rediffmail.com, [2]debajyotibiswas0@gmail.com, [3]buddhadeb.sau@gmail.com

***Abstract*—The use of WiFi is now a part of each human life. Airports, railways, bus-stand, home, markets everywhere now people uses WiFi because its reliability and low-cost. WiFi is also applicable in future tech of IoT (Internet of Things). In this work we used two NodeMCU (ESP8266 WiFi module) which is a easily programmable. NodeMCU which acts itself as a sensor node can be used as Access Point (AP) or as a STAtion (STA). We used one node as a AP and another as a STA. To locate a device distance measurement one of most important issue. There are lots of technique to find out the distance between two nodes (e.g. Time of Arrival (TOA), Time Difference of Arrival (TDOA) or Received Signal Strength (RSS) algorithms etc). In this work we use RSSI technique to determine the distance. First, we take around 300 sample data (RSSI values) and find the the standard deviation to calculate how much the RSSI values are spread out and use curve fitting technique to find suitable equation for estimate distance. Then, we compared the estimated distance with actual distance to find the error level in percentage. We are success to reduce the average error level up to 8.32%.**

***Keywords*—Arduino IDE, Curve fitting, ESP8266, NodeMCU, RSSI values, Error level.**

I. INTRODUCTION

This paper is describing how to measure the distance between two nodes (we used NodeMCU as sensor node). The NodeMCU is a Arduino type board which runs on ESP8266 module. We find the distance using RSSI value and use Curve Fitting Technique (CFT). We also compare this technique with Estimated Signal Strength (ESS) formula and Friis Transmission Equation (FTE).

In Wireless Sensor Network (WSN), RSSI measurements [1], [2], multipath [3] is a relevant issue for the distance measurement performance. There are several methods proposed in [4], [5], [6], [7], [8] to the RSSI measurements. Changing the values of RSSI due to the transmitters and receivers non idealities [9] and the influence the variability and accuracy of RSSI measurements. In the paper [10], T. I. Chowdhury et al. have used a multi-step RSSI-based distance calculation model using Samsung Galaxy S4 smartphone. They achived the error level upto 13.4%.

In this work, we used two NodeMCU +25dBm, one as a Access Point and another as station. First, we program one NodeMCU using Arduino IDE software as a Access Point by connecting it via USB to micro-USB cable. Once the program is loaded, we connect it using a power bank to provide continous power supply. Then the other node is programed as a station by connecting it USB to micro-USB cable. We programmed the station node in such a way that it will only take the RSSI value of the first node if any other access point fall in to its range it will neglate them. Once the program is loaded, we start to save the data using Processing. NodeMCU (ESP8266): Developer ESP8266 Opensource Community, Type Single-board microcontroller, Operating system XTOS, CPU ESP8266 (LX106), Memory 128kBytes, Storage 4MBytes, Power USB [11].

II. SAMPLE AND POPULATION STANDARD DEVIATION CALCULATION FOR SPREAD OUT RSSI VALUES FOR A PARTICULAR DISTANCE

We calculate Standard Deviation to measure how much the RSSI values are spread out from the average (i.e. mean), or expected value. It is Normal distribution and we calculate sample standard deviation taking around 300 sample RSSI values also,we calculate population standard deviation by the formula (1) and (2) in TABLE I. We measure from 0.3 meter to 10 meter (we take 17 values of distance and each distance we take around 300 RSSI values) we find there some lowest standard deviation close to One (i.e. the RSSI values are close to the mean for a particular distance) and fewer high standard deviation greater than Three (i.e. the RSSI values are spread

TABLE I
CALCULATING STANDARD DEVIATION FOR SPREAD OUT RSSI VALUES

Distance	Sample Standard Deviation (s)	Population Standard Deviation (σ)	Mean ($\bar{x}$)
0.3m	1.2040	1.2020	-55.0598
0.4m	1.5584	1.5559	-59.3521
0.5m	1.9998	1.9965	-59.9368
0.6m	3.1203	3.1151	-61.7176
0.7m	1.1691	1.1671	-66.1993
0.8m	2.3658	2.3619	-68.1262
0.9m	1.1807	1.1787	-69.4418
1m	1.7201	1.7172	-71.2724
2m	2.6989	2.6944	-73.5049
3m	2.1855	2.1818	-79.0598
4m	2.5681	2.5639	-81.1328
5m	2.5540	2.5498	-83.1528
6m	2.2814	2.2777	-83.5747
7m	1.9820	1.9787	-84.7209
8m	1.8456	1.8425	-86.1295
9m	1.8790	1.8759	-86.5813
10m	1.5578	1.5552	-88.0564

high range from the mean for a particular distance). One can find some variety of standard deviation (S.D.) between this high and low S.D., this variety indicate that the RSSI values changing their character for a particular distance with respect to mean. This difference of standard deviation may be vary for multi path signal, various radio frequencies around us, weather, high winds, moisture and various obstacles etc. Although, it may be not, that our Calculating standard deviation is much higher in all radio frequencies measurements. We get maximum difference 19 of around 300 RSSI values at distance 0.6 m and minimum Value -93 of around 300 RSSI at distance 10 m. Although one can observe that the S.D. is varying at different distance and low S.D. gives high signal and high S.D. gives low signal. In the equations (1) and (2), we use N is number of sample, μ is Population mean (for large sample size, $\mu = \bar{x}$), x_i is belongs to around 300 RSSI values for any particular distance $(i = 1 : N)$ [12].

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{1}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{2}$$

III. Output (Receive) Power measurement to find distance by Friis transmission equation for free space propagation

In wireless sensor network, a cell data device is radio device and there signal strength and signal quality both are measured in dBm(i.e. decibels relative to one milliwatt), is connected to the cellular tower. RSSI is a negative dBm value, values closer to 0 dBm are strong signals. We observe that NodeMCU (ESP8266) generally dose not recognized the signal strength less than -95 dBm. We calculate the output (receive) power for a particular dBm value in TABLE II. When we save data (around 300 RSSI values) using processing, we observe, different distance, takes different time. Here we use a transmitter and receiver NodeMCU (ESP8266) devices, receiver connected to one laptop and the transmitter to a power bank. Here we can get an idea for which RSSI values what power need to the receiver. We do our experiment on a roof of two storied building and place them on Roof top. Here the equation (3) and (4) represent the dBm to milliwatt conversion and milliwatt to dBm conversion respectively. We Calculate the Receive Power and by this power, we can find the distance by Friis transmission equation (8) for free space propagation. In this section we calculate the maximum, minimum and average of around 300 RSSI value for each distance and calculate how much milliwatt power receive by this devices for one milliwatt input power. World moves to WiFi and High speed wireless technology. Here we get an idea how and where the NodeMCU (ESP8266) may use in modern wireless technology.

$$P_{out} = 10^{\frac{dBm}{10}} \tag{3}$$

$$dBm = 10 \times log_{10}[\frac{P_{out}}{1mW}] \tag{4}$$

TABLE II
Find Output Power (Receive power) of RSSI Values to find distance by Friis transmission equation for free space propagation

Distance	Largest RSSI Value	Smallest RSSI Value	Average (Round off) RSSI Value	Average Output Power(mW)
0.3m	-51	-58	-55	0.0000031622776602
0.4m	-54	-63	-59	0.0000012589254118
0.5m	-55	-65	-60	0.0000010000000000
0.6m	-57	-76	-62	0.0000006309573445
0.7m	-61	-69	-66	0.0000002511886432
0.8m	-63	-73	-68	0.0000001584893192
0.9m	-67	-72	-69	0.0000001258925412
1.0m	-67	-75	-71	0.0000000794328235
2.0m	-70	-80	-74	0.0000000398107171
3.0m	-73	-87	-79	0.0000000125892541
4.0m	-75	-88	-81	0.0000000079432823
5.0m	-77	-90	-83	0.0000000050118723
6.0m	-78	-89	-84	0.0000000039810717
7.0m	-78	-89	-85	0.0000000031622777
8.0m	-79	-92	-86	0.0000000025118864
9.0m	-81	-92	-87	0.0000000019952623
10.0m	-82	-93	-88	0.0000000015848932

IV. Fit Curve (function), find the distances and compare the error Percents with different formulas

Here we take 0.3 m to 10 m actual value and try to fitting a curve by different equations, among all curves this function (5) are fit suitable then other curves. This graph takes the positive value between -5 dBm to -95 dBm RSSI values and the other values are negative(i.e. the integer values less than -95 and greater than -5 are takes the graph negative co-ordinate). By using this graph we reduce the percentage of error as much as minimum, taking the actual and the estimated value (which given by the function and two different formulas). In 300 RSSI values we take the round off average value of them. Fig. 1 says that the (RSSI, Distance) co-ordinates are close to our estimated graph, which gives us a minimum error to our measurement. So by the graph it is easy to say that at which RSSI value, we get what distance and conversely we can find out the RSSI values by distance with minimum error. We only take a portion of graph, the graph is not universally true for all RSSI values. We compare the graph with the Estimated Signal Strength formula and Friis transmission equation for free space propagation in TABLE III and Fig. 2. For Estimated Signal Strength formula (6) and (7), dBm_e is estimated received power in mobile device, dBm_0 any known signal power at any distance d_o, γ is Path loss exponent (average value of 4 for mobile devices). Here we take any known signal power dBm_0=-71 for distance d_0=1 [13] and for Friis transmission equation for free space propagation (8) Where, $P_t = VI$ ($V = 3.3V$ and $I = 1A$), P_r is receive signal Power, P_t is transmitted signal Power (taken

TABLE III
COMPARING ACTUAL AND ESTIMATED DISTANCE FOR FUNCTION, ESTIMATED SIGNAL STRENGTH FORMULA AND FRIIS TRANSMISSION EQUATION

Actual Distance	RSSI Value in dBm	Estimated Distance by CFT function (d_A)	Estimated Distance by ESS formula (d_B)	Estimated Distance by FTE formula (d_C)	Error in CFT function (d_A)	Error in ESS formula (d_B)	Error in FTE formula (d_C)	error percent in CFT function (d_A)	error persent in ESS formula (d_B)	error persent in FTE formula (d_C)
0.3m	-55	0.2900m	0.3981m	0.3213m	0.01	0.0981	0.0213	3.333%	32.7%	7.1%
0.4m	-59	0.3742m	0.5011m	0.5092m	0.0258	0.1011	0.1092	6.45%	25.275%	27.3%
0.5m	-60	0.4029m	0.5308m	0.5714m	0.0971	0.0308	0.0714	19.42%	6.16%	14.28%
0.6m	-62	0.4723m	0.5956m	0.7193m	0.1277	0.0044	0.1193	21.283%	0.733%	19.88%
0.7m	-66	0.6761m	0.7498m	1.14m	0.0239	0.0498	0.44	3.4142%	7.114%	62.85%
0.8m	-68	0.8236m	0.8413m	1.43m	0.0236	0.0413	0.63	2.95%	5.1625%	78.75%
0.9m	-69	0.9127m	0.8912m	1.61m	0.0127	0.0088	0.71	1.41%	0.9777%	78.88%
1m	-71	1.1293m	1.0000m	2.02m	0.1293	0	1.02	12.93%	0%	102%
2m	-74	1.5829m	1.1885m	2.86m	0.4171	0.8115	0.86	20.855%	40.575%	43%
3m	-79	2.9307m	1.5848m	5.09m	0.0693	1.4152	2.09	2.31%	47.173%	69.67%
4m	-81	3.8387m	1.7782m	6.41m	0.1613	2.2218	2.41	4.0325%	54.545%	60.25%
5m	-83	5.1232m	1.9952m	8.07m	0.1232	3.0048	3.07	2.464%	60.096%	61.4%
6m	-84	5.9720m	2.1134m	9.05m	0.028	3.8866	3.05	0.4666%	64.7766%	50.83%
7m	-85	7.0137m	2.2387m	10.16m	0.0137	4.7613	3.16	4.4542%	68.0185%	45.14%
8m	-86	8.3118m	2.3713m	11.40m	70.3118	5.6287	3.4	3.8975%	70.3587%	42.5%
9m	-87	9.9605m	2.5118m	12.79m	70.9605	6.4882	3.79	10.6722%	72.09%	42.1%
10m	-88	12.1052m	2.6607m	14.35m	2.1052	7.3393	4.35	21.052%	73.393 %	43.5%

0.0033 mW), G_t is the transmitter antenna gain (taken 1), G_r is the receiver antenna gain (taken 1), λ is wave length in meter(take, 0.125m) [14]. Although this equation gives Huge error for finding distance. The fitting graph is better accurate than this two formula. Average Error(%) by the function d_A: $(141.3892/17) = 8.317294$, Average Error(%) by the formula d_B: $(629.146/17) = 37.0085$, Average Error(%) by the formula d_C: $(849.43/17) = 49.9664$

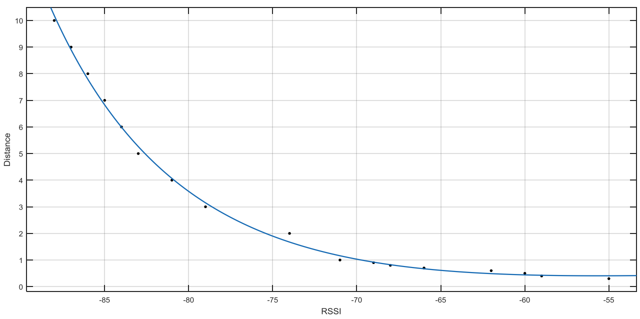

Fig. 1. Distance vs RSSI Curve Fit by Plotting Various Points

$$d_A = \frac{(-0.043x^5 - 4.92x^4 - 171.5x^3 - 600.8x^2 + 41.41x - 0.84)}{(x^4 + 250.4x^3 + 14780x^2 - 455.9x + 12.24)} \tag{5}$$

$$dBm_e = dBm_0 - 10\gamma \times log_{10}[\frac{d}{d_0}] \tag{6}$$

$$d_B = d_0 \times 10^{\frac{(dBm_0 - dBm_e)}{10\gamma}} \tag{7}$$

$$d_C = \sqrt{\frac{P_t G_t G_r \lambda^2}{(4\pi)^2 P_r}} \tag{8}$$

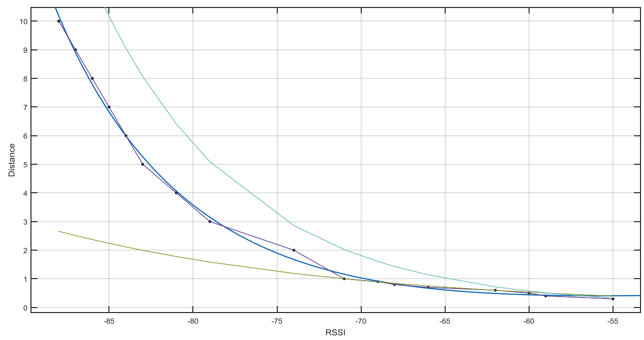

Fig. 2. Compare between Fitting Graph(deep blue), Actual Graph (violet), Estimated Signal Strength Formula(green) and Friis Transmission Equation(blue)

$$Error(\%) = \frac{(d_{actual} - d_{estimated})}{d_{actual}} \times 100 \tag{9}$$

V. PROGRAMMING CODE AND ENVIRONMENT

Arduino IDE is used to program the nodes which allows to program the Micro-controller. In Arduino IDE programming there are two part one is setup() and another is loop(). setup() function runs only once but loop() function runs continuously or repeatedly. All the codes and functions are called from this two funtion. At the time of upload the program to Arduino board, it should connected with computer USB port. Arduino IDE is C++ type language shown in *Program 1* and *Program 2*. In *Program 1* defines Arduino code to make the NodeMCU into an Access Point which act as transmitter and *Program 2* defines Arduino code to make the NodeMCU into a Station which act as receiver.

Program 1: To make NodeMCU as Access Point

```
#include<WiFiClient.h>
#include<ESP8266WiFi.h>
#include<ESP8266WebServer.h>
const char *ssid="NodeMCU01";
const char *password="12345678";
ESP8266WebServer server(80);
void handleRoot()
{
  server.send(200,"text/html","<h1>Connected</h1>");
}
void  setup()
{
  delay(1000);
  Serial.begin(115200);
  Serial.println();
  Serial.print("Configuring access point...");
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);
  server.on("/",handleRoot);
  server.begin();
  Serial.println("HTTP server started");
}
void  loop()
{
  server.handleClient();
}
```

Program 2: To make NodeMCU as station

```
#include<ESP8266WiFi.h>
void  setup()
{
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  Serial.println("Setup done");
}
void  loop()
{
  int n=WiFi.scanNetworks();
  if(n==0)
    Serial.println("No networks found");
  else
  {
    for(int i=0;i<n;++i)
    {
      if(WiFi.SSID)(i)!="NodeMCU01")
        continue;
      Serial.print(WiFi.RSSI(i));
      Serial.print(",");
    }
  }
}
```

We use processing to save the recived data from station node into text file via Universal Serial Bus (USB) port. Processing software is an open-source programming language which allow to use the serial port (COM1, COM2, COM3,...) access. The USB port can be accessible as virtual serial port which can be used as a normal serial port. In processing there are two part one is setup() and another is draw(). setup() function runs only once but draw() function runs continuously or repeatedly. The programming language is java type shown in *Program 3*.

Program 3: To save data into text file using Processing

```
import processing.serial.*;
Serial mySerial;
PrintWriter output;
int cnt=0;
void  setup()
{
  mySerial=new Serial(this, Serial.list()[0],115200);
  output=createWriter("data.txt");
}
void  draw()
{
  if(mySerial.available()>0)
  {
    String value=mySerial.readString();
    if(value!=null)
    {
      output.print(value);
      print(value);
      cnt++;
    }
  }
  if(cnt>300)
  {
    output.flush();
    output.close();
    exit();
  }
}
```

VI. Conclusion and Future Works

In this work we try to find the distance between two wireless nodes using RSSI technique. Here, we first take 300 RSSI sample values for each fix distance then we take the average for plotting graph to fit proper graph. Using curve fitting we got an equation to find estimate distance of target node and find the error percentage in measurement. The drawback of this work that we can't tell the direction of target node as we use only two nodes.

In future, we will find out the distance between two nodes from a distributed system like wireless sensor network where we find the direction as there will be multiple sensor node. We also plan to develop the robot navigation technique [15] using this distance finding method.

Acknowledgment

The first and second authors would like to thank UGC and CSIR respectively to provide them formal financial support to complete the project work and carry their research works.

References

[1] L. Xu, F. Yangl, Y. Jiangl, L. Zhang, C. Fengl and N. Baol, "Variation of Received Signal Strength in Wireless Sensor Network" Proc. of 2011 3rd International Conference on Advanced Computer Control (ICACC 2011), pp. 151-154.

[2] D. Capriglione, L. Ferrigno, E. D'Orazio, V. Paciello and A. Pietrosanto, "Reliability analysis of RSSI for localization in small scale WSNs", proceedings of IEEE International Instrumentation and Measurement Technology Conference (I2MTC) 2012, pp. 2149-2154.

[3] D. Yang, Y. Xu and M. Gidlund, "Coexistence of IEEE802.15.4 Based Networks: A Survey", IECON 201036th Annual Conference on IEEE Industrial Electronics Society, pp. 2107-2113.

[4] A. Moschitta, D. Macii, F. Trenti, S. Dalpez and A. Bozzoli, "Characterization of a Geometrical Wireless Signal Propagation Model for Indoor Ranging Techniques", 2012 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 2598-2603.

[5] C. Ladha, B. Sharif and C. Tsimenidis, "Mitigating propagation errors for indoor positioning in wireless sensor networks", IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS 2007), 2007, pp. 1-6.

[6] A. Lazaro, D. Girbau, P. Moravek and R. Villarino, "A Study on Localization in Wireless Sensor Networks using Frequency Diversity for Mitigating Multipath Effects", Elektronika ir Elektrotechnika, vol. 19, no. 3, pp. 82-87, 2013.

[7] C. Zhang, W. Qi, P. Liu and L. Wei, "Multipath cancellation by frequency diversity: a training-free and analytical approach to accurate RSS ranging in ground-deployed wireless sensor networks", Electronics Letters, vol. 50, no. 6, pp. 471-473, 2014.

[8] A. Ramachandran and J. Sarangapani, "Use of frequency diversity in signal strength based WLAN location determination systems", in 32nd IEEE Conference on Local Computer Networks, 2007 (LCN 2007), pp. 117-124.

[9] D. Capriglione, E. D'Orazio, L. Ferrigno, V. Paciello and A. Pietrosanto, "Reliabildity Analysis of RSSI for Localization in Small Scale WSNs", in Proceedings of IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2012 , pp. 935-940.

[10] T. I. Chowdhury et al., "A multi-step approach for RSSi-based distance estimation using smartphones", 2015 International Conference on Networking Systems and Security (NSysS), Dhaka, 2015, pp. 1-5. doi: 10.1109/NSysS.2015.7042942

[11] http://www.nodemcu.com

[12] https://www.scribd.com/doc/211901395/S-C-Gupta-V-K-Kapoor-Fundamentals-of-Mathematical-Statistics-a-Modern-Approach-10th-Edition-2000.

[13] Weixing Xue, Weining Qiu, Xianghong Hua, and Kegen Yu,Senior Member, IEEE, "Improved Wi-Fi RSSI Measurement for Indoor Localization", IEEE Sensors Journal, Vol. 17, Issue. 7, pp. 2224-2230.

[14] Free Space Radio Wave Propagation (Website: http://www.iitg.ernet.in)

[15] S. Barai, A. Dey, B. Sau, "Path following of autonomous mobile robot using passive rfid tags", In Proceedings of the IEEE International Conference on Microelectronics, Computing and Communications (MicroCom), Durgapur, India, 2325 January 2016, pp. 16.