



The Egyptian International Journal of Engineering Sciences and Technology

Vol. 24 (January 2018) 48-55

<http://www.eijest.zu.edu.eg>



Formal Verification of Fuzzy Logic Wireless Sensor Network Localization System

Hisham Abdullah*, Hazem Shehata, Ibrahim Ziedan

Computer and Systems Engineering Department, Zagazig Univeristy, Zagazig – Postcode: 44519, Egypt.

ARTICLE INFO

Article history:

Received 19 Oct. 2017
Received in revised form
01 December 2017
Accepted 05 December 2017
Available online 15 June 2018

Keywords:

Formal Verification
Wireless Sensor Network (WSN)
Fuzzy Based Trilateration (FBT)
Modified Fuzzy Based Trilateration (MFBT)
Fuzzy logic
Received Signal Strength Indicator (RSSI)
Link Quality Indicator (LQI)
Power Level (PL)
PRISM model checker
Discrete-Time Markov Chains (DTMC)

ABSTRACT

Formal verification is mainly used to prove the correctness of safety-critical hardware and software systems. Localization problem in Wireless Sensor Network (WSN) is a hot research topic with many critical applications. Fuzzy Based Trilateration (FBT) algorithm is a simple and efficient localization technique that has been recently proposed. In this paper, we use PRISM model checker to formally verify the correctness of FBT algorithm. Our verification results show that the fuzzy rules used in the original FBT algorithm can be modified to reduce the average localization error without increasing the complexity of the algorithm. Our results also show that the localization error in the modified FBT (MFBT) algorithm is more likely to be acceptable than the original FBT algorithm.

© 2018 EIEST. All rights reserved

1. INTRODUCTION

A Wireless Sensor Network consists of thousands of sensor nodes which are limited in terms of energy sources, storage capacity and processing capabilities [1]. Wireless Sensor Networks have many critical applications such as: detection and controlling of forest fires, monitoring and assisting disabled patients, Studying the life style of endangered animals by tracking their movements, environmental monitoring, military surveillance, and, many others [2]. Sensor node location is very important and

critical in most applications where the collected data is worthless without knowing sensor's location [3, 4]. Formal verification is one of the most important methods of proving or disproving the correctness of both hardware and software systems with respect to a certain formal specification (or property) using formal methods of mathematics. In simple words, Formal verification can be defined as the process of checking whether a design satisfies some requirements (properties) [5-7]. Therefore formal verification provides a reliable mechanism to check the correctness of its requirements for any critical application. The two most popular methods for formal verification are language containment and

* Corresponding author Tel: +201-064-849-292
E-mail address: hisham22@zu.edu.eg.

model checking. The latter have been applied in verifying ad-hoc networks, wireless network protocols, and WSN. [8-10]

PRISM tool is one of the probabilistic model checkers which can be used for modeling and analysis of systems that show a probabilistic attitude. PRISM performs probabilistic model checking based on exhaustive search and numerical solution [11]. PRISM tool has many advantages such as:

- Built on top of JAVA platform (widely used).
- Accepts very simple modeling language.
- Supports several types of probabilistic models such as: discrete-time Markov chains (DTMCs), continuous-time Markov chains (CTMCs), Markov decision processes (MDPs), and probabilistic timed automata (PTAs).
- Provides a simple graphical user interface (GUI) to simulate and model-check.
- Provides special constructs (e.g., “formula” and “cost/reward”) for writing arithmetic and conditional expressions without increasing the number of the states in the model.

This paper introduces a strategy for modeling and verification of Fuzzy Base Trilateration (FBT) localization algorithm. The paper also proves that a simple modification to the fuzzy rules used in the original FBT algorithm can enhance its accuracy. Our contribution in this paper is the introduction of a new strategy for modeling and verification of WSN localization algorithms based on trilateration technique without increasing its complexity, and enhancing the case study algorithm. The verification model is built and checked using PRISM model checker in which the complexity of the model does not increase due to using the Formulas and Rewards features of the PRISM tool.

The rest of the paper is outlined as follows: Section 2 covers background and highlights the related work on WSN localization, Fuzzy logic system and the fundamental of FBT algorithm essential for understanding the proposed verification model. In Section 3, the verification technique is described in detail. In section 4, verification results are showed. Finally, the conclusion and future work are drawn in Section 5.

2. BACKGROUND

2.1. WSN localization techniques

There are many techniques used to provide location of individual nodes in Wireless Sensor Networks. Performance of localization techniques can be evaluated by many factors. The main factors affecting the quality of localization techniques are cost, power saving and energy consideration, processing speed and, accuracy. [12]

WSN localization techniques can be classified into two categories: rang-based localization and range-free localization. Range-based localization determines the absolute distances or angles. Range-free localization provides relative distances to references. Each category is divided into two types: fully type or hybrid type. This division depends on using single localization method or combination of them. Fully and hybrid types may depend on anchor nodes or not. Anchor nodes are a small number of WSN nodes whose positions are known. Anchor nodes are aware of their position manually or using GPS. GPS cannot be used in localization of all nodes in WSN due to cost and power consumption. But it can be used in estimate anchor nodes’ positions because the number of anchor nodes is small. Hybrid type is more accurate and complex than fully type. Range-based type is more accurate, complex and scalable than range-free type. Anchor based localization techniques have less processing time than anchor free. Figure 1 shows the tree classification of localization technique in WSN. [4, 12, 13]

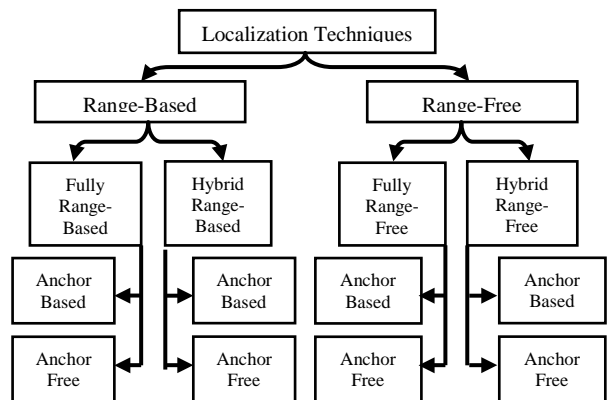


Figure 1 Tree classification of WSN localization techniques [12]

2.2. Fuzzy logic system

Any fuzzy logic system consists of four main stages which are a fuzzifier, fuzzy rules, inference engine, and defuzzifier as shown in Figure 2. [1, 14, 15]

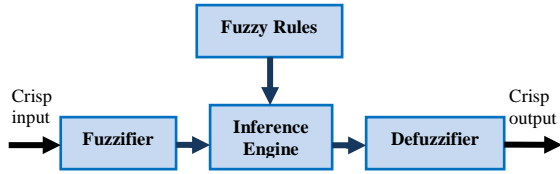


Figure 2 Fuzzy Logic System [1]

Fuzzifier receives crisp inputs and converts them into fuzzy inputs using input membership functions. Crisp inputs refer to physical parameters that are entered to the system as inputs. Fuzzy rules define a set of rules. Each rule provides a case of occurrence of the fuzzy inputs in the system. Inference engine is used to map each rule of fuzzy inputs to fuzzy output using a list of if-then conditional statements which are known as fuzzy rules. Inference engine also combines the output of each rule into a single fuzzy set of output. Finally, defuzzifier is used to transform the fuzzy output produced from the previous stage into a crisp output using output membership function. [1, 14, 15]

2.3. Fuzzy Base Trilateration (FBT)

FBT algorithm can be considered as a Hybrid range-based localization technique that depends on anchor nodes. Trilateration is a well-known localization technique that is used to calculate the position of a node in WSN. The unknown node, which position is unknown, calculates its position using trilateration by estimating the distances between it and three anchor nodes using the following equations [1, 15]:

$$d_1^2 = (x - x_1)^2 + (y - y_1)^2 \quad (1)$$

$$d_2^2 = (x - x_2)^2 + (y - y_2)^2 \quad (2)$$

$$d_3^2 = (x - x_3)^2 + (y - y_3)^2 \quad (3)$$

Where, (x,y) is the unknown node coordinate and, (x1,y1), (x2,y2), (x3,y3) are the coordinates of three anchor nodes for 2D region. Figure 3 shows two dimension trilateration localization technique.

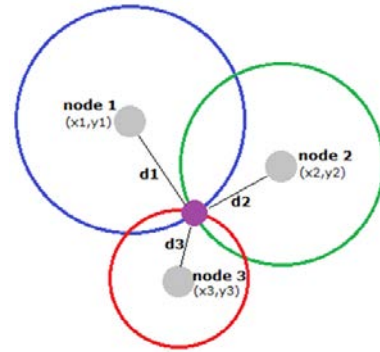


Figure 3 Two dimension trilateration localization technique

FBT technique has three inputs: Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI) and Power Level (PL). The goal of using three inputs instead of using only RSSI is to increase the accuracy of the technique. Each of the system inputs is mapped to three levels defined by three linguistic terms: low, medium, and high. Each point of any crisp input ranges from 0 to 1 as depicted by Figure 4. [1]

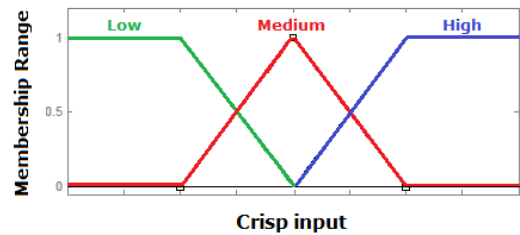


Figure 4 Input membership function of FBT system [1]

Fuzzy output is an estimate distance defined by five linguistic terms: very low, low, medium, high, and very high as depicted by Figure 5. [1]

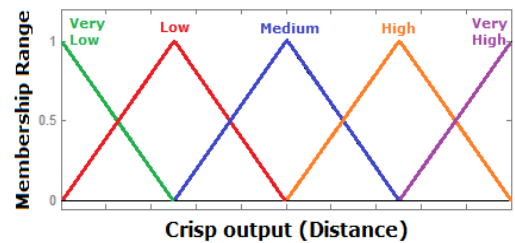


Figure 5 Output membership function of FBT system [1]

Fuzzy rules that are used to determine fuzzy distance are defined according to FBT algorithm in [1] as shown in Table 1.

Table 1 Fuzzy rules of fuzzy logic system. [1]

Row No.	Fuzzy Inputs			Fuzzy Output
	RSSI	PL	LQI	Distance
0	Low	Low	Low	High
1	Low	Low	Medium	Medium
2	Low	Low	High	Low
3	Low	Medium	Low	High
4	Low	Medium	Medium	Medium
5	Low	Medium	High	Medium
6	Low	High	Low	Very High
7	Low	High	Medium	High
8	Low	High	High	High
9	Medium	Low	Low	Medium
10	Medium	Low	Medium	Medium
11	Medium	Low	High	Low
12	Medium	Medium	Low	Medium
13	Medium	Medium	Medium	Medium
14	Medium	Medium	High	Medium
15	Medium	High	Low	High
16	Medium	High	Medium	Medium
17	Medium	High	High	Medium
18	High	Low	Low	Low
19	High	Low	Medium	Low
20	High	Low	High	Very Low
21	High	Medium	Low	Medium
22	High	Medium	Medium	Medium
23	High	Medium	High	Low
24	High	High	Low	High
25	High	High	Medium	Medium
26	High	High	High	Low

A flowchart of FBT localization algorithm is shown in Figure 6 and the steps of operation can be summarized as follows [1]:

Step1: each unknown node collects RSSI, LQI, and PL of the connected anchor nodes.

Step2: each unknown node selects the nearest three anchor nodes according to the collected data.

Step3: each crisp value of RSSI, LQI, and PL is converted into fuzzy value based on the input membership function in figure.

Step4: fuzzy rules in table 1 are applied at the three fuzzy values of RSSI, LQI, and PL to calculate fuzzy distance.

Step5: The output membership function is used to convert fuzzy distance into a crisp distance.

Step6: the position of the unknown node is determined using trilateration technique.

Step7 The calculated position is broadcast.

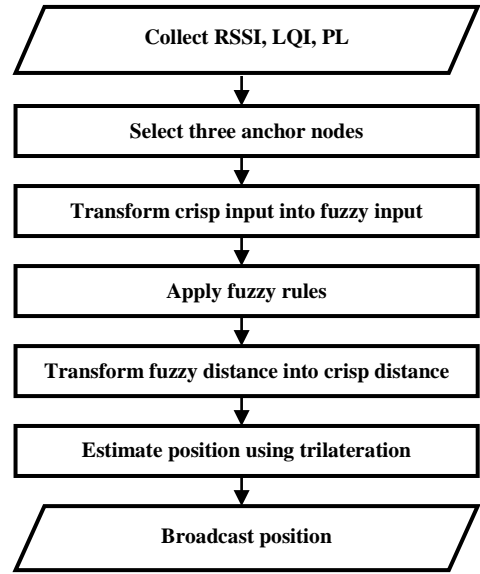


Figure 6 Flowchart of FBT localization algorithm

3. VERIFICATION TECHNIQUE

Our verification technique consists of the following three phases: model representation, formal specification and model verification as shown in Figure 7

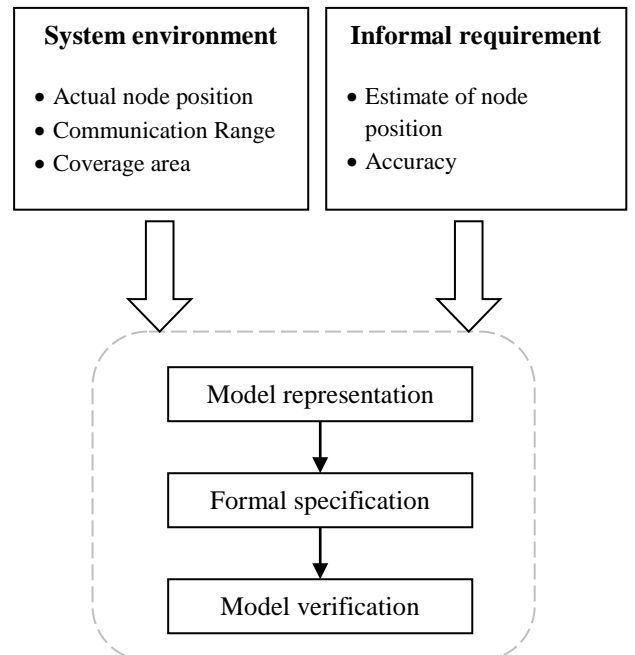


Figure 7 Verification technique phases

In model representation phase the system environment is represented using PRISM modeling language. In formal specification phase the informal requirements are translated into properties. In model verification phase the correctness of our PRISM model is checked against the formal properties. [16, 17]

3.1. The PRISM Model

We model the system and the environment in which the actual node position (x and y coordinates) are selected randomly with uniform probability distribution in the coverage area. We work in two dimension region of coverage area and so, there are two uniform distributions one for the x position of the unknown node along the x-axis and another one for the y position of it along the y-axis. This assumption can be built in PRISM tool using Discrete Time Markov Chains (DTMC) model with two levels of states (two steps) [18]. The first level is used to choose a random value of the actual x coordinate of the unknown node and the second level is used to choose a random value of actual y coordinate of that node. The rest of the calculations are expressed as PRISM formulas. Using PRISM formulas does not increase the total number of reachable states. The actual distances between anchor nodes and the unknown node are calculated. After calculating the actual distance, we select the three nearest anchor nodes to the unknown node. We calculate the actual values of RSSI, LQI and PL that are supposed to be measured by the sensor node in reality. RSSI, LQI and PL values are calculated using the following equations: [1, 15, 19, 20]

$$RSSI = -10 \times np \times \log_{10}(d) + \alpha \quad (4)$$

Where d represent distance between unknown node and anchor node, np is the open air path loss factor (np = 2) and, $\alpha = -70$ (constant for open air media). [1]

$$LQI = \begin{cases} 255 & d = 0 \\ (\ln\left(\left(\frac{R_c}{100 \times d}\right)^2\right) + 10) \times 25 & 0 < d \leq R_c \\ 0 & d > R_c \end{cases} \quad (5)$$

Where R_c represent the communication range. For more detail see. [19]

$$P_L = \frac{P_T \times L \times (4\pi d)^2}{G_t \times G_r \times \lambda^2} \quad (6)$$

Where: ($\lambda = c / f$), c represent the speed of light, and f is the carrier frequency that equal 2.4 GHz for Zigbee protocol of WSN, L = 1 (unit length), and P_T is the minimum transmission power that equal one using energy save mode. [1]

Then, we convert the crisp values of the actual inputs (RSSI, LQI, and PL) into fuzzy values. Then, we apply fuzzy rules (using conditional formulas in PRISM model checker) to estimate the distances between the unknown node and the three anchor nodes using the following equations:

$$x(x_3 - x_2) + y(y_3 - y_2) = \frac{(d_1^2 - d_2^2) - (x_1^2 - x_2^2) - (y_1^2 - y_2^2)}{2} \quad (7)$$

$$x(x_1 - x_2) + y(y_1 - y_2) = \frac{(d_1^2 - d_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)}{2} \quad (8)$$

Where, (x,y) is the unknown node coordinate, (x1,y1), (x2,y2), (x3,y3) are the coordinates of the three anchor nodes, and (d1,d2,d3) are the estimated distances. Then, we calculate the error in each reachable state (i.e. each possible position for the unknown node) based on the following equation:

$$error = \frac{P_i^e - P_i^a}{R_c} \quad (9)$$

Where P_i^e is the estimated position of node i, P_i^a represents the actual position of node i, and R_c is the communication range [1].

$$(P_i^e - P_i^a) = |x_i^e - x_i^a| + |y_i^e - y_i^a|$$

Where (x_i^e, y_i^e) are the coordinates of the estimated position and (x_i^a, y_i^a) are the coordinates of the actual position. We use the following equation to calculate the average error where N represents number of nodes:

$$error = \frac{1}{N} \sum_{i=1}^N \frac{P_i^e - P_i^a}{R_c} \quad (10)$$

We use rewards feature of PRISM tool to record all the values of the system parameters in each reachable state. [21] For example we use the following PRISM reward structure to calculate the error in each reachable state:

rewards " error "
 true : $(p_i^e - p_i^d)/R_c$;
 endrewards

We set condition of error calculation to be true for all reachable state to record error in all positions covered by the WSN. The verification strategy can be summarized as shown in Figure 8 [22] where m represents number of accessible position along the x_axis and k represents number of accessible position along the y_axis. The states in the first level have equal probability of 1/m while the states in the second level have equal probability of 1/k.

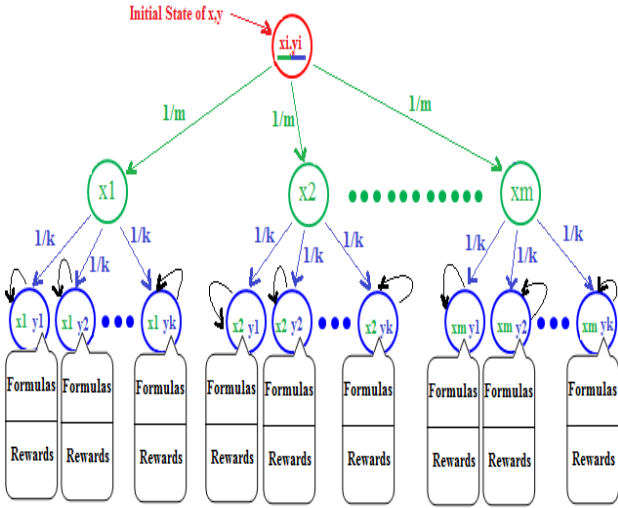


Figure 8 Verification strategy

3.2. The PRISM properties

To calculate the average error we use the following PRISM property [23]:

$$R\{\text{"error"}\}=? [S]$$

Where S indicates that, this property calculates the average error after reaching the steady state. The same result can be obtained (in less verification time) by the following PRISM property [24]:

$$R\{\text{"error"}\}=? [I = 2]$$

This property is faster than the previous property because this property requires PRISM to simulate the system for only two steps (rather than simulation until reaching the steady state). Two steps here are sufficient to choose random values for the actual position (x,y).

To estimate how likely the error will be less than 1% of communication range) we verify the following PRISM property:

$$P =? [XX \text{ error} < (0.01 * R_c)]$$

This property tells PRISM to calculate the probability for the error to be less than 1% of communication range after two steps. The same results can be obtained by using the following property:

$$S=? [\text{error} < (0.01 * R_c)]$$

4. VERIFICATION RESULTS

We simulate the same conditions, data and, requirements of the FBT localization algorithm in [1] to compare the result with our Modified version of FBT (MFBT). We examine the performance of the FBT algorithm using different communication ranges which are 150 m, 200 m, and 250 m and compare the average error (in meter) recorded in the FBT algorithm with the MFBT algorithm. The network simulation area is open space of 600*600 (m2) . We modeled 200 nodes with 8 anchor nodes. We verified two version of the FBT algorithm:

- The original FBT algorithm as presented in reference [1].
- The modified version of the FBT algorithm (MFBT) in which two of the fuzzy rules are updated as shown in Table 2.

Table 2 Updated rows of fuzzy rules

Row No.	Fuzzy Inputs			Fuzzy Output
	RSSI	PL	LQI	Distance
0	Low	Low	Low	Medium
10	Medium	Low	Medium	Low

Each of the two versions is expressed in PRISM as explained in section 3.1 and verified against the properties as shown in section 3.2. For each of the two versions FBT and MFBT algorithms, we

performed three verification tests with three different communication ranges: 150 m, 200 m, and 250 m.

Figure 9 shows average position error in the FBT algorithm and the MFBT algorithm for each of the three communication ranges that we considered. Overall in our verification result the MFBT algorithm has an average position error of 0.83 m, while the FBT has an average position error of 0.85 m. The average position error reduces when communication range increases.

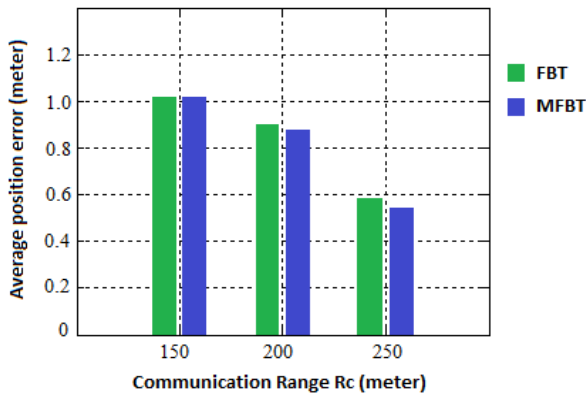


Figure 9 Average position error of the FBT and the MFBT algorithms for different communication ranges.

Figure 10 shows how likely the error in each reachable state falls within the acceptable range which is less than 1 m. On average, the probability of the localization error to be acceptable in MFBT algorithm is 89.67% while that probability in FBT is 89.16%. The probability of error to fall within the acceptable range increases when communication range increases.

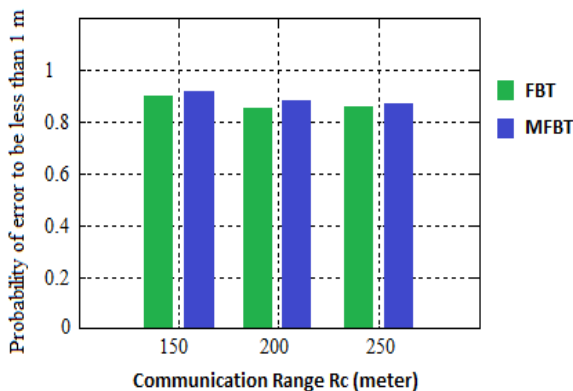


Figure 10 Probability of error less than 1 m for the FBT algorithm and the MFBT algorithm for different communication ranges.

Figure 11 shows probability of the error to be less than 0.7 m. On average, the probability of the localization error to be small (less than 0.7 m) in MFBT algorithm is 66%, while that probability in FBT is 64.3%. The probability of the error to be less than 0.7 m increases when communication range increases.

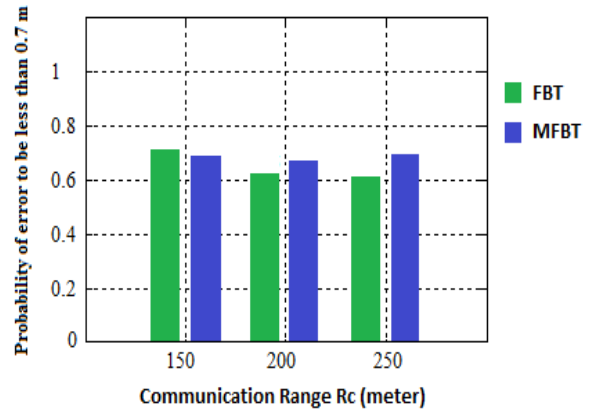


Figure 11 Probability of error less than 0.7 m for the FBT algorithm and the MFBT algorithm for different communication ranges.

5. CONCLUSION AND FUTURE WORK

In this paper, we present a strategy for modelling and verification of localization techniques that are based on trilateration. Our study provides better understanding for the accuracy of trilateration localization by a case study performed on the FBT algorithm. Our study shows that a simple modification to the FBT algorithm can improve the average position accuracy by 2.35%, the average probability of the error to be acceptable (less than 1 meter) by 0.6% and the average probability of the error to be small (less than 0.7 m) by 2.58%

There is a new issue to this verification technique can be summarized by:

- Study the performance of the FBT algorithm by changing the distribution of the possible locations of the unknown nodes.
- Provide property to detect the best modification on fuzzy rules that may achieve optimal accuracy.
- Apply verification technique on other localization method.

References

- [1] B. R. El Samadony, R. A. Ramadan, and N. M. Darwish, "Enhanced trilateration localization using fuzzy logic for WSN," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 5, pp. 378-385, 2015.
- [2] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: Overview of sensor networks," *Computer*, vol. 37, pp. 41-49, 2004.
- [3] M. A. Azim, Z. Aung, W. Xiao, and V. Khadkikar, "Localization in wireless sensor networks by cross entropy method," in *International Conference on Ad Hoc Networks*, 2012, pp. 103-118.
- [4] M. Srbinovska, V. Dimcev, C. Gavrovski, and Z. Kokolanski, "Localization techniques in wireless sensor networks using measurement of received signal strength indicator," *Electronics*, vol. 15, pp. 67-71, 2011.
- [5] J. Bowen and V. Stavridou, "Safety-critical systems, formal methods and standards," *Software Engineering Journal*, vol. 8, pp. 189-209, 1993.
- [6] P. Camurati and P. Prinetto, "Formal verification of hardware correctness: Introduction and survey of current research," *Computer*, vol. 21, pp. 8-19, 1988.
- [7] V. D'silva, D. Kroening, and G. Weissenbacher, "A survey of automated techniques for formal software verification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, pp. 1165-1178, 2008.
- [8] L. Gallina, T. Han, M. Kwiatkowska, A. Marin, S. Rossi, and A. Spanò, "Automatic energy-aware performance analysis of mobile ad-hoc networks," in *Wireless Days (WD), 2012 IFIP*, 2012, pp. 1-6.
- [9] M. Dufloy, M. Kwiatkowska, G. Norman, D. Parker, S. Peyronnet, C. Picaronny, et al., "Practical applications of probabilistic model checking to communication protocols," 2012.
- [10] M. Z. Kwiatkowska, "Formal methods for the analysis of wireless network protocols," University of Oxford, 2011.
- [11] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: probabilistic model checking for performance and reliability analysis," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, pp. 40-45, 2009.
- [12] A. Mesmoudi, M. Feham, and N. Labraoui, "Wireless sensor networks localization algorithms: a comprehensive survey," *arXiv preprint arXiv:1312.4082*, 2013.
- [13] G. Chawla and M. Khurana, "Performance and Accuracy Analysis of Range-free Localization Algorithms in Wireless Sensor Network."
- [14] A. Allahverdiyev, "Cargo Transportation Routing Under Fuzzy Conditions," *International Journal on Technical and Physical Problems of Engineering (IJTPE)*, vol. 3, pp. 45-48, 2011.
- [15] R. Y. Jyoti and N. Singh, "Localization in WSN using modified trilateration based on fuzzy optimization," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, 2013.
- [16] S. M. Babamir and M. Borhani, "Formal verification of medical monitoring software using Z language: a representative sample," *Journal of medical systems*, vol. 36, pp. 2633-2648, 2012.
- [17] D. Méry and N. K. Singh, "Trustable formal specification for software certification," in *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation*, 2010, pp. 312-326.
- [18] M. Kwiatkowska, G. Norman, and D. Parker, "Probabilistic symbolic model checking with PRISM: A hybrid approach," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2002, pp. 52-66.
- [19] R. Behnke and D. Timmermann, "AWCL: adaptive weighted centroid localization as an efficient improvement of coarse grained localization," in *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, 2008, pp. 243-250.
- [20] E. Navarro-Alvarez and M. Siller, "A node localization scheme for Zigbee-based sensor networks," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, pp. 728-733.
- [21] C. Von Essen, B. Jobstmann, D. Parker, and R. Varshneya, "Synthesizing efficient systems in probabilistic environments," *Acta Informatica*, vol. 53, pp. 425-457, 2016.
- [22] C. Mu, P. Dittrich, D. Parker, and J. E. Rowe, "Formal Quantitative Analysis of Reaction Networks Using Chemical Organisation Theory," in *International Conference on Computational Methods in Systems Biology*, 2016, pp. 232-251.
- [23] M. Kwiatkowska, D. Parker, and A. Simaitis, "Strategic analysis of trust models for user-centric networks," *arXiv preprint arXiv:1303.0791*, 2013.
- [24] F. Dannenberg, E. M. Hahn, and M. Kwiatkowska, "Computing cumulative rewards using fast adaptive uniformization," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 25, p. 9, 2015.