Dynamic Sampling Localization Algorithm Based on Virtual Anchor Node*

Wang Wei[1**], Yan Bingshu[1], Guo Min[1], Yuan Ping[2], Yin Feng[3], Luo Qian[4], and Chen Liangyin[1***]

[1] School of Computer Science, Sichuan University, Chengdu, China,
[2] School of Mathematics and Information Engineering, Chongqing University of Education, Chongqing, China,
[3] School of Computer Science and technology, Southwest University for Nationalities, Chengdu, China,
[4] Second Research Institute,General Administration of Civil Aviation of China, Chengdu, China.

Abstract. Compared with localization in the static sensor network, the node localization in dynamic sensor networks is more complicated due to the mobility of nodes. Dynamic Sampling Localization Algorithm Based on Virtual Anchor (RSMCL) is proposed in this paper, and this algorithm can implement localization for the unknown nodes in dynamic sensor networks. Firstly, RSMCL algorithm predicts the speed and movement direction of nodes in current movement to verify a sector sampling area. Secondly, a method of calculating the sampling quantity with the size of the sampling area dynamically changing is proposed in this paper. Lastly, virtual anchor node, the unknown node which got preliminary possible area (PLA) assists the other unknown node to reduce PLA, and the last PLA as a filter conditions to filter out the conflicting sample points quickly. In this way, the filtered sample is close to its real coordinates. The simulation results show that RSMCL algorithm can improve the positioning performance when ensuring the execution time is shorter and the localization coverage rate is higher. The localization error of RSMCL algorithm can be dropped to about 20%.

Keywords: Probabilistic Wireless Sensor Networks · Node Localization · Monte Carlo · Region Segmentation · RSMCL.

---

* This work was supported by the National Natural Science Foundation of China (grant no. 61373091); NNSFC&CAAC U1533203; the key Technology R&D program of Sichuan Province (grant no. 2016GZ0068, Science & Technology Department of Sichuan Province, China); the Collaborative Innovation of Industrial Cluster Project of Chengdu (grant no. 2016-XT00-00015-GX); Civil Aviation Airport United Laboratory of Second Research Institute, CAAC && Sichuan University of Chengdu (grant no. 2015-YF04-00050-JH).

** Wang Wei and Yan Bingshu have equal contribution

*** corresponding author:`chenliangyin@scu.edu.cn`

1 Introduction

The node localization in wireless sensor networks refers to a process, which a node that does not know its own location in the network can compute the exact location coordinates via a certain algorithm. The node localization is the premise of wireless sensor networks in practical application, which can make great importance in this field. According to whether it is needed to measure the physical distance or not, the technologies of node localization can be mainly divided into two categories, range-base and range-free localization algorithm. If these results in traditional condition are applied into the dynamic wireless sensor networks, the mobile characteristics of nodes will improve the positioning performance of the algorithm. It is still one of the serious challenges faced in the dynamic wireless sensor networks about how to implement the low-energy and high-accuracy node localization.

Most of localization algorithms studied in dynamic sensor networks are based on the Monte Carlo Localization(MCL) theory, whose basic concept is that it makes full use of N samples to represent the posterior probability distribution of node localization. The main innovations are as follows,

- In the sampling phase, predicting the motion trajectory of the node to be located result in a sector sampling area, which area is the place where the node is most likely to be located. The algorithm uses Newton interpolation to predict the motion velocity and motion direction of the node at the following moment, and then it obtains the sector area as the sampling area according to the error distribution of interpolation method.
- We propose a calculation approach, for changing the sampling quantity with the density of the sampling area and the unknown node, which can reduce the number of repeated sampling and the execution time of the positioning.
- In the filtering phase, use the unknown node in the preliminary possible location area as virtual anchor nodes to narrow its possible area, we use the preliminary possible location area as filter condition to obtain the valid sample. Since the filtering conditions are more strict, the filtered samples are closer to the true position coordinates. Therefore, the positioning accuracy is significantly improved, and this accuracy is good at low anchor node density (the ratio of the number of localizable nodes and unknown nodes).

This paper is organized as follows. Section 2 presents the innovations of the RSMCL algorithm in details. The performance evaluations and simulation analysis are presented in Section 3. Conclusions are given in Section 4.

2 RSMCL algorithm

As a MCL algorithm, the RSMCL algorithm obtains samples with weights by sampling and filtering. It uses samples to estimate the posterior probability distribution of node positions. RSMCL algorithm mainly includes the predicting, filtering, re-sampling and position estimation phases. The flowchart of RSMCL algorithm is shown in Fig. 1. The most important phase is the prediction and filtering. The prediction phase is to determine the sampling area and sampling. The filtering phase uses the filtering condition to filter the samples for obtaining a set of valid samples.

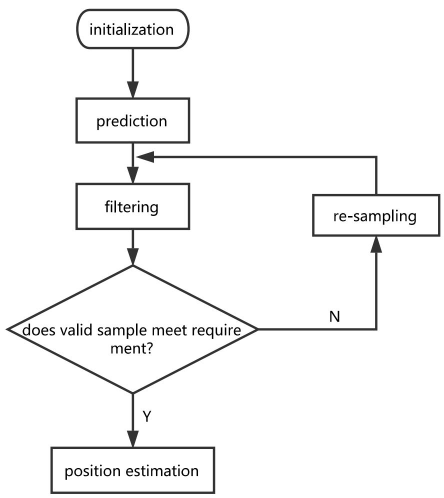

Fig. 1. RSMCL flowchart

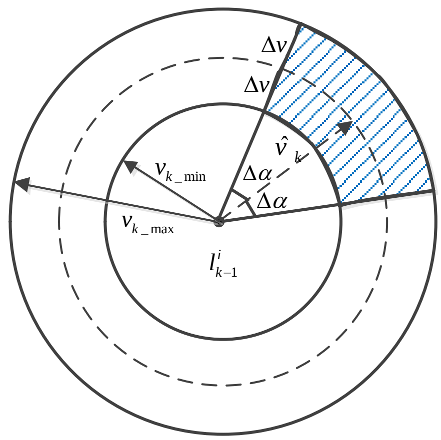

Fig. 2. Sampling Area Diagram

2.1 Sampling Area Determination

Firstly, this algorithm needs to determine the sampling area. In sampling phase of the RSMCL algorithm, the size and accuracy of sampling area are most important. The sampling area proposed in this paper is a sector segment area [8] . This paper introduces the Newton interpolation method for predicting the direction and speed of motion. The principle of the Newton interpolation method is, suppose the function $f(t)$ value $f(t_0)$,$f(t_1)$,...,$f(t_{k-1})$ at the k different time t_0,t_1,...,t_{k-1} are x_0,x_1,...,x_{k-1}. According to the Newton interpolation polynomial, the function values x_k of the previous k moment used to compute the function value X_{k+1} are as follows,

$$\begin{aligned} N_k(t) = f(t_0) + f[t_0, t_1](t - t_0) + f[t_0, t_1, t_2](t - t_0)(t - t_1) \\ +... + f[t_0, t_1, ..., t_k](t - t_0)(t - t_1)...(t - t_{k-1}) \end{aligned} \tag{1}$$

According to the collection of particles L_0,L_1,...,L_{k-1} at the previous k moment and *motion model*, the node can determine the collection of particles L_k. In this motion model, node cannot determine motion speed and direction at current moment, only knowing that maximum speed is v_{max}. The Newton interpolation operation is performed on the node in the two directions of coordinate x and coordinate y, and the coordinates of the node at time k is obtained as (x_k, y_k). So, the velocity of two direction, v_{kx} and v_{ky} at time t are predicted as,

$$v_{kx} = \frac{x_k - x_{k-1}}{k} \tag{2}$$

$$v_{kx} = \frac{y_k - y_{k-1}}{k} \tag{3}$$

k, is the size of every time slot. From the above, the velocity v_k and motion direction of the unknown node at time k can be obtained,

$$\hat{v_k} = \sqrt{v_{kx}^2 + v_{ky}^2}, \hat{\alpha_k} = arctan(\frac{y_k - y_{k-1}}{x_k - x_{k-1}}) \tag{4}$$

If you use the coordinates of all the previous moments to estimate the current position, Newton interpolation increases the amount of computing in addition to storing more data. Therefore, considering the theory and actual requirement, RSMCL algorithm uses the latest 3 position coordinates to estimate the possible motion velocity v_k of current moment. Because the motion velocity can not be estimated at $k = 0$, $k = 1$ and $k = 2$, the motion velocity is the maximum velocity which the system sets, $\hat{v_k} = v_{max}$.

As the Fig. 2 shows, the dash area is the sector sampling area in the RSMCL algorithm. There is a certain error in the interpolation calculation method, and it meets the normal distribution. The nodes to be located may be located in the surrounding area of the four directions of the predicted position, so a speed parameter Δv and an angle parameter $\Delta\alpha$ are introduced in the algorithm. The obtained sector area is the most likely location of the node to be located at the next moment. First, the algorithm causes the nodes to increase or decrease the size of Δv in the estimation direction to obtain two radii, v_{k_max} and v_{k_min}, and with the estimated position at last moment as the center of the circle to get two circles, as shown in Fig. 2. Since the motion trajectory of node is relatively smooth, the motion direction of the node does not change abruptly. Therefore, the algorithm introduces an angle parameter $\Delta\alpha$ in the estimate direction . It expands an angle $\Delta\alpha$ of the clockwise and counterclockwise directions in the estimation direction of the node. Then the shade area shown in Fig. 2 is obtained, which is the most likely area of the nodes at the current moment. System settings, $\Delta\alpha = \frac{2\pi}{9}$, $\Delta v = 0.5v_k$ (section 2.2 introduces how to obtain two parameters). In the motion model, the range of node velocity is (v_{min}, v_{max}), and when $v_{k\_max} > v_{max}$, assume $v_{k\_max} = v_{max}$; when $v_{k\_min} < v_{min}$, assume $v_{k\_min} = v_{min}$. In the sampling area, it randomly sample n sample point(n as actual sample number, as motion model introduces how to obtain n). Therefore, the shade area as the sampling area, the mathematics expression is as following,

$$L_i = \left\{ (x_k^i, y_k^i) \middle| \begin{matrix} v_{k\_min \le d[(x_k^i, y_k^i), x_{k-1}, y_{k-1}]} \le v_{k\_max} \\ \cap \left| \frac{y_k^i - y_{k-1}}{x_k^i - x_{k-1}} \right| \le tan\Delta\alpha \end{matrix} \right\} (0 \le i \le n) \tag{5}$$

2.2 Sampling Number

Algorithm needs to confirm the number of sampling in the sampling phase of RSMCL algorithm. After confirming sampling area, this paper proposes a method of sampling number according to sample area size and node density change for increasing sampling rate and saving energy consumption.

When anchor box created by the anchor node is quite small, only a few samples are needed to achieve a certain positioning accuracy. When the anchor box is large, a certain number of samples are required to achieve the required positioning accuracy. In this paper, the number of samples is calculated according to the size of the sampling area obtained in the motion prediction phase and the density of the node to be located, and the total number of sampling times is minimized on the basis of ensuring the positioning accuracy. The number of samples required for positioning is as follows,

$$n = \frac{S_{sample}}{R \times UD} \times N_U = \frac{S \times S_{sample}}{N_U \times \pi R^3} \times N \tag{6}$$

The density of nodes in networks, $UD = N_U/(\frac{S}{\pi R^2})$. n is the actual sampling numbers, R is the communication radius of the node, S_{sample} is the range of sampling area, N is maximum sampling number set by system. S is the range of whole network, N_U is the number of the unknown nodes in the networks. In order to prevent the sampling area in the sampling phase from being too small, the number of samples is extremely small, and minimum number of samples set by the system is $N_{\_min}$. If the actual number of samples calculated n is less than $N_{\_min}$, then $n = N_{\_min}$. In order to prevent excessive sampling, this topic states that when the calculated actual number of samples n is greater than the maximum number of samples N set by the system, then $n = N$.

2.3 Filtering Condition Obtaining

In RSMCL algorithm, each node to be located continuously reduces the nodes own possible location area by using its anchor node neighbor and the virtual anchor node neighbor with respect to its own possible virtual communication circle and the inevitable communication circle of its neighbor node, using the idea of the Bounding-Box algorithm [4].

The virtual communication circle of the anchor node is used to segment the possible location area of the positioning node, as shown in Fig. 3. The polygon $ABCD$ is the preliminary possible location area obtained by the node $U1$ to be located, and the circle in the figure is the virtual communication circle of the anchor node $A1$. It is assumed that $U1$ is the two-hop neighbor of $A1$, and the radius of the virtual communication circle is $2R$. According to the communication principle of network, $U1$ is located in the virtual communication circle. So the node $U1$ to be located can be located at a possible position outside the virtual communication circle and segment this area. Firstly, the vertices of the possible localization areas outside the virtual communication circle are respectively connected, C, D and the anchor node $A1$, and two intersection points obtained by

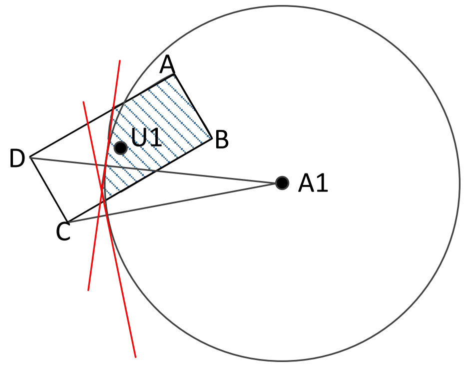

Fig. 3. $A1$ virtual communication circle

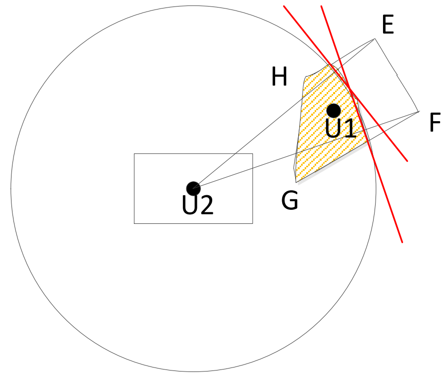

Fig. 4. $U2$ virtual communication circle

the two lines respectively intersect the virtual communication circle. Then the virtual communication circle is respectively made through the two intersection points. The two tangent lines divide the possible position area of the node $U1$ to be located into a shaded portion as shown in Fig. 3, as an updated possible position area of $U1$. The virtual communication circle of the virtual anchor node is used to segment the possible location areas of the positioning node, as shown is Fig. 4. The polygon $EFGH$ is a possible location area of the node $U1$ to be located, and the circle in the figure is the virtual communication circle of the virtual anchor node $U2$. In the same way as shown in Fig. 3, $U2$ is used to segment the possible location area of $U1$, and the shaded portion in Fig. 4 is the updated position location area of the node $U1$ to be located.

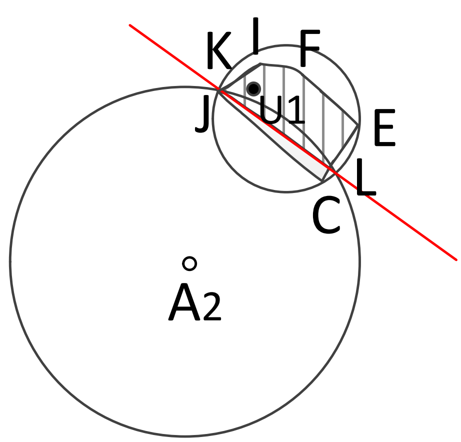

Fig. 5. utilize the inevitable communication circle of $A2$

After the virtual communication circle is utilized, the positioning node is segmented by using the inevitable communication circle of the neighbor node. Fig. 5 is a schematic diagram of the inevitable communication circle assisted positioning using the anchor node neighbor. It is assumed here that U1 is the two-hop neighbor of $A2$, the polygon $CLEFIKJ$ is the current possible location area of the node $U1$ to be located, the circle on the left is the inevitable communication circle of the anchor node $A2$, and the circle on the right is the circumcircle of the possible location area of $U1$. As shown in the figure, the two circles intersect as two intersection points, and the two intersection points are connected to obtain a straight line which cuts the possible position area of $U1$ into two parts. It can be known from the communication principle that the node $U1$ to be located is not located in the inevitable communication circle of $A2$, because if it is within the inevitable communication circle, $U1$ is a one-hop neighbor node of $A2$, which does not conform to the assumption. Therefore, the node $U1$ can segment the possible location area within the inevitable communication circle of $A2$, and the shaded portion in Fig.. 5 is the possible location area after the $U1$ update. Similarly, it is the same to use the the virtual communication node neighbor's inevitable communication circle for segmenting the area, and it will not be described in detail here. After the node to be located obtains the final possible location area, as a filtering condition of the filtering phase, the samples that do not meet the conditions are removed. If the number of filtered samples is less than n, re-sampling is performed until the number satisfies n. At the same time, the average weighting calculation is performed on all the samples, and the coordinates of the node to be located are estimated.

3 Simulation and Analysis

In order to compare the fairness of different algorithm performances, this topic compares the various performances of MCL, MCB, WMCL and RSMCL algorithms under the same network parameter settings. Assume that 100(quantity variation range[50−150])anchor nodes and 640(quantity range[400−1300])unknown nodes are randomly distributed in the sensor network region of $1000m \times 1000m$, and both anchor nodes and unknown nodes are adopted the Random Path Point Motion Model(RWPM) moves randomly within the network area. The moving node determines a target position according to the movement model in the network area, and randomly selects a speed among $[v_{min}, v_{max}]$. After reaching the target position, the node pauses for a period of time t_{pause}. In the simulation, if the node is always in motion, set the pause time t_{pause} to 0. The maximum communication radius of the node is set to $100m$, and the communication irregularity DOI varies from $[0 - 0.3]$. The simulation result of the algorithm is the average value calculated after 20 simulations.

3.1 Performance Analysis

Speed and angle parameter selection In the RSMCL algorithm, the size and accuracy of the sampling area concern the final positioning error. The algorithm needs to confirm a speed parameter Δv and an angle parameter $\Delta\alpha$, so that the positioning error of the algorithm is the lowest.

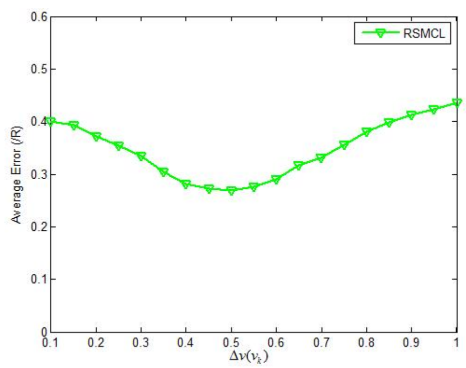

Fig. 6. the influence of speed parameter

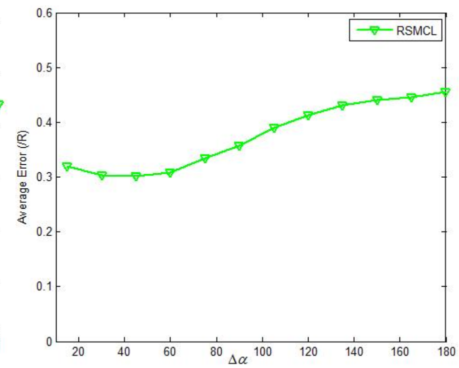

Fig. 7. the influence of angle parameter

As the Fig. 6 shows, the positioning error is observed with the speed parameter Δv at the range of $[0.4v_k, 0.6v_k]$. $\Delta\alpha$ is less error about the localization error at around 40 degrees, as shown in Fig. 7. In summary, the speed parameter set $0.5v_k$ in the RSMCL algorithm is that the angle parameter is $2\pi/9$ 40 degrees.

Performance analysis The performance analysis mainly analyzes the re-sampling rate, computing time, total coverage, positioning error and communication energy consumption of the algorithm. The specific analysis is as follows.

Fig. 8 shows the re-sampling rate under the change of anchor node density. The actual number of samples in the RSMCL algorithm varies with the area of the sampling area, reducing many unnecessary re-sampling, so the re-sampling rate is low.

Fig. 9 shows the average positioning time as a function of the density of unknown nodes. It can be seen that when the density of unknown nodes increase, the computing time of several algorithms decreases to varying degrees. The computing time of the RSMCL algorithm is significantly lower than other algorithms, indicating that the sampling efficiency of the RSMCL algorithm is higher than other algorithms.

Fig. 10 shows the location coverage under the change of the density of the anchor node, where the density of the unknown node remains at 20. From the analysis, the unknown nodes in the dynamic wireless sensor network can be located. It is seen that the average positioning error of the RSMCL algorithm is greatly improved.

Fig. 12 shows the positioning coverage as a function of the maximum speed of the node. It is seen that the RSMCL algorithm estimates the motion speed of the node. The maximum motion speed has little influence on the positioning of

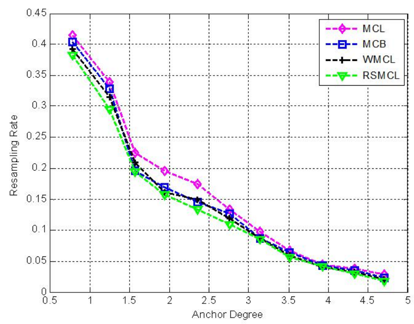

Fig. 8. influence of anchor to re-sampling

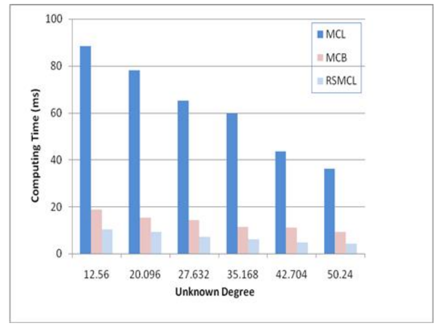

Fig. 9. influence of unknown nodes to average time

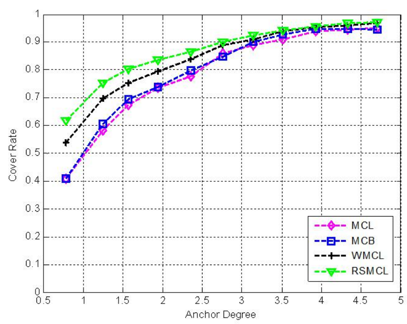

Fig. 10. the influence of anchor density to localization coverage rate

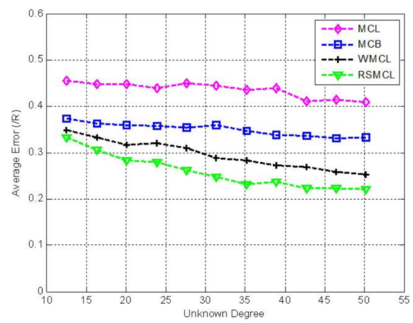

Fig. 11. the influence of unknown nodes density to average localization error

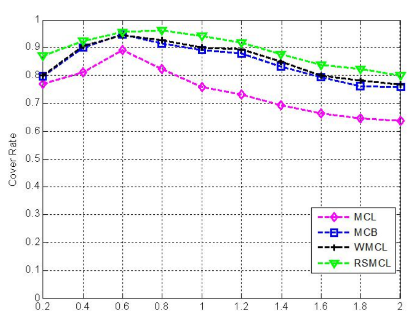

Fig. 12. the influence of maximum motion speed to coverage rate

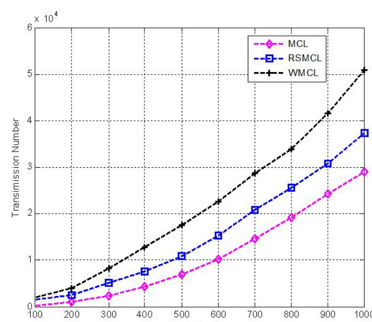

Fig. 13. Comparison of the communication consumption

the algorithm, so the location coverage rate is the highest. As can be seen from Fig. 13, when the number of nodes in the network increases and the nodes to be located increases, so traffic of the network increases. Among these algorithms, the RSMCL algorithm is modest.

4 Conclusion

In this paper, we proposed a RSMCL algorithm used for node localization in the dynamic sensor network. The improvements include determination of sampling area, sampling number and filtering condition. After determining the sampling area and sampling number, we made full usage of the unknown node in the preliminary possible localization area as the filtering condition. The experimental results show that the RSMCL algorithm effectively shortens positioning time and improve sampling efficiency and positioning accuracy. Because of many other mobility models, analyzing the effect of the proposed algorithm in other models is the future work.

References

1. Taponecco, L., D'amico, A., Mengali, U.: Joint TOA and AOA estimation for UWB localization. IEEE Transaction on Wireless Communications **10**(7), 2207–2217 (2011). https://doi.org/10.1109/TWC.2011.042211.100966
2. Xu, B., Sun, G., Yu, R.: High-Accuracy TDOA-Based Localization without Time Synchronization. IEEE Transaction on Parallel and Distributed Systems **24**(8), 1567–1576 (2013). https://doi.org/10.1109/TPDS.2012.248
3. Kulakowski, P., Vales-Alonso, J., Egea-Lopez, E., Ludwin, W., Garca-Haro., J.: Angle-of-arrival localization based on antenna arrays for wireless sensor networks. Computers & Electrical Engineering **36**(6), 1181–1186 (2010). https://doi.org/10.1016/j.compeleceng.2010.03.007
4. Tomic, S., Beko, M., Dinis, R.: Distributed RSS-based localization in wireless sensor networks using convex relaxation. 2014 International Conference on Computing, Networking and Communications (ICNC) 853–857(2014). https://doi.org/10.1109/ICCNC.2014.6785449
5. Wang, W., Zhu, Q.: RSS-based Monte Carlo localisation for mobile sensor networks. Communications, IET **2**(5), 673–681(2008). https://doi.org/10.1049/iet-com:20070221
6. Martins, M., Chen, H., Sezaki, K.: OTMCL: Orientation tracking-based Monte Carlo localization for mobile sensor networks. Proceedings of the 6th International Conference on Networked Sensing Systems (INSS), 1–8(2009). https://doi.org/10.1109/INSS.2009.5409931
7. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research . Wireless communications and mobile computing, **2**(5), 483–502(2002). https://doi.org/10.1002/wcm.72
8. Chen, L., Li, M., Liu, K., Liu, Z., Zhang, J., Peng, T., Liu, Y., Xu, Y., Luo, Q., He, T.: Distributed range-free localisation algorithm for wireless sensor networks. ELECTRONICS LETTERS, **50**(12), 894-896(2014). https://doi.org/10.1049/el.2014.0787