Extended Min-Max Algorithm for Position Estimation in Sensor Networks

Jorge Juan Robles*, Javier Superv√≠a Pola** and Ralf Lehnert*
Chair for Telecommunications
Technische Universität Dresden, Germany
*{robles|lehnert}@ifn.et.tu-dresden.de
**jsuperviapola@gmail.com

***Abstract*—A localization algorithm that is often used by sensor nodes is Min-Max [1][2]. This algorithm can be easily executed due to the fact that it principally consists of few additions, subtractions and logical comparisons. However, Min-Max provides a coarse position estimation. In our proposal we improve the accuracy of Min-Max by including simple extra operations. We compare the accuracy of our extended Min-Max (E-Min-Max) with other algorithms by using simulation.**

I. Introduction

The knowledge about the position of a sensor node can be exploited by many applications, like access control, navigation systems as well as monitoring.

The position can be estimated in a centralized way. Here, the sensor nodes take measurements and send them to a central computer, which is in charge of the estimation. The big advantage of this proposal is that the central computer can execute complex and accurate localization algorithms because of its high processing power. On the other hand, the drawback is that a considerable traffic can be generated reducing the scalability of the network.

Another option is to use decentralized calculation, where a sensor node is able to estimate the position for itself without sending localization information to a central computer. Due to the limited processing power of the sensor nodes, low complexity localization algorithms are required in this case.

Actually, there are several methods for estimating the position of a node. This paper deals with distance-based localization algorithms, where distances to reference nodes are used. These reference nodes are called anchors (ANs) and their positions are known. There are exact and approximate distance-based localization algorithms. The exact methods provide the exact coordinates of the blind node (BN) while there is no error in the distance determination. On the contrary, the approximate algorithms can provide a coarse position in the above mentioned conditions. Note that the main advantage of the approximate algorithms is the robustness against the error and, in general, their low complexity [1]. We investigate two exact methods: Range-based Least Square (R-LS) [3] and Linear Least Square (L-LS) [1][5] to compare the performance of the approximate algorithms Min-Max and E-Min-Max.

II. Localization algorithms

A. Range-Based Least Square

Consider a 2D scenario with n anchors placed at the coordinates $[x_i, y_i]$ $(i = 1...n)$ and one BN located at $[x_0, y_0]$. In order to calculate the BN position, the distances d_i from this node to each anchor i have to be estimated. Thus, by using the Pythagoras theorem it is possible to obtain the following system of equations:

$$\begin{aligned} (x_1 - x_0)^2 + (y_1 - y_0)^2 &= d_1^2 \\ \vdots \qquad\qquad & \quad \vdots \\ (x_n - x_0)^2 + (y_n - y_0)^2 &= d_n^2 \end{aligned} \tag{1}$$

One option to estimate the BN position is to solve the following Range-Based Least Square problem [3]. The final position of the BN is taken as those that minimizes (2).

$$R\text{−}LS: \min_{x_0, y_0} \sum_{i=0}^{n} \left(\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - d_i\right)^2 \tag{2}$$

However, this minimization is not trivial due to the fact that this function is not convex.

An inefficient method to solve (2) is to analyze the function in all possible points in the scenario. This method is called brute-force or exhaustive method. In this paper we implement an algorithm that operates in a similar way, but it is faster than the exhaustive method because only some points of the scenario are evaluated to find the minimum value.

For this purpose, an iterative algorithm has been implemented. In the first iteration, the scenario is divided into N_1 sections forming a square grid. In the intersection points of the grid the above mentioned least square function is evaluated. Then, a minimum value is obtained and the next iteration starts. Here, the new search space is reduced to a square with side two times the previous inter-grid spacing. The center of the square is located at the coordinates where the minimum value was obtained. Again, this square is divided into N_1 sections, where the inter-grid spacing is now $\sqrt{N_1}/2$ times smaller than the previous one. Thus, the resolution increases and a new minimum value can be obtained. This process is repeated until the maximum number of allowed iterations N_2 is reached.

978-1-4673-1439-8/12/$31.00 ©2012 IEEE

This algorithm can fail if the initial inter-grid spacing is large due to the fact that the algorithm cannot converge to the minimum global value. We obtained satisfactory results by setting N_1=100 and N_2=5 in our reference scenario described in section III-B.

Note that due to its large processing time, it is not optimal to implement this technique in a sensor node. In this paper we use this algorithm as reference to compare the performance of our proposed algorithms in terms of position accuracy.

B. Linear Least Square Algorithm

The set of equations (1) can be linearized if its last equation is subtracted from the others [1] [5]. Thus, after rearranging terms, the resulting set of equations can be expressed in matrix form as $At = b$, where:

$$A = \begin{pmatrix} 2(x_n - x_1) & 2(y_n - y_1) \\ \vdots & \vdots \\ 2(x_n - x_{n-1}) & 2(y_n - y_{n-1}) \end{pmatrix} \tag{3}$$

$$b = \begin{pmatrix} (d_1^2 - d_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ \vdots \\ (d_{n-1}^2 - d_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{pmatrix} \tag{4}$$

$$t = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \tag{5}$$

The position of the BN can be estimated by minimizing the following function:

$$L{-}LS: \min_t\{\|At - b\|^2\} \tag{6}$$

The solution of (6) can be easily computed by solving:

$$t = (A^T A)^{-1} A^T b \tag{7}$$

The complexity of L-LS strongly decreases compared with the previous algorithm allowing a computation-limited sensor node to execute this exact algorithm. However, the achieved solution is suboptimal in case the estimated distances contain errors [3].

C. Min-Max Algorithm

This algorithm builds a square (bounding box) around each anchor. As shown in Fig.1, each side of the bounding box is two times the measured distance d_i between the BN and the corresponding AN. The bounding boxes are used to delimit a region where the position of the BN will be estimated. This region called ”definition zone” is a square whose vertices are placed at the following coordinates:

$$\begin{aligned} P_1 &= [max(x_i - d_i), max(y_i - d_i)] \\ P_2 &= [max(x_i - d_i), min(y_i + d_i)] \\ P_3 &= [min(x_i + d_i), min(y_i + d_i)] \\ P_4 &= [min(x_i + d_i), max(y_i - d_i)] \\ &\forall i \in [1, \ldots, n] \end{aligned} \tag{8}$$

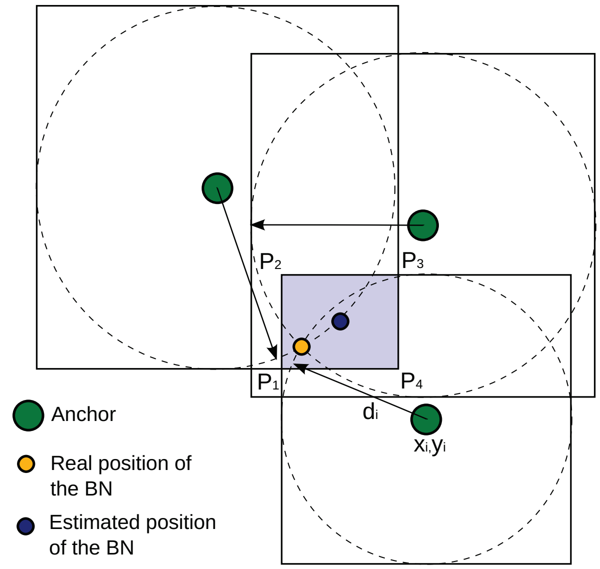

Fig. 1. Operation of Min-Max with three anchors

where $min()$ and $max()$ are the minimum and maximum function, respectively.

In the original Min-Max algorithm, the final position of the BN is approximated by calculating the geometric average of the vertices of the definition zone [1].

$$\begin{aligned} x_0 &= \frac{min(x_i + d_i) + max(x_i - d_i)}{2} \\ y_0 &= \frac{min(y_i + d_i) + max(y_i - d_i)}{2} \\ &\forall i \in [1, \ldots, n] \end{aligned} \tag{9}$$

In general, the definition zone is formed by the intersection of the bounding boxes as represented in Fig.1. However, it is possible that there is no overlapping zone between the bounding boxes. This can occur when the estimated distances are too short compared to the real case. In such scenarios Min-Max operates without problems providing an coarse position.

Note, that Min-Max can produce a high position error in some cases. For example in Fig.2., assume that there is an ”internal zone”, which is delimited by the ANs of the periphery. In case the BN is located outside of this internal zone, Min-Max trends to locate its estimation inside the internal zone. Therefore, the bigger the separation between the BN and the internal zone the higher the error in the position estimation. Another approximate algorithm that has a similar behavior is Weigthed Centroid Localization (WCL) [4]. On the contrary, the above mentioned exact algorithms do not present this problem.

D. Extended Min-Max Algorithm

The difference of E-Min-Max and its original version is the fact that the E-Min-Max does not locate the estimation at the center of the definition zone. On the contrary, the estimated BN's position can be located in any point inside the defined

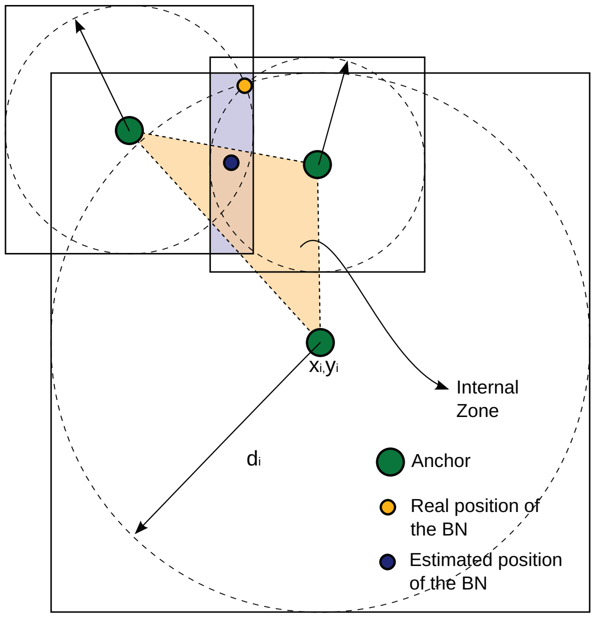

Fig. 2. Internal zone problem

region. Thus, the described problem of the internal zone can be minimized. For this purpose, a weight W_a is given to each vertex j of the definition zone. This weight indicates how similar the BN coordinates can be with respect to the coordinates of the vertices $[x_j, y_j]$. In this work, we evaluate E-Min-Max with four different weights:

$$W_1(j) = \frac{1}{\sum_{i=1}^{n} |D_{i,j} - d_i|} \tag{10}$$

$$W_2(j) = \frac{1}{\sum_{i=1}^{n} (D_{i,j} - d_i)^2} \tag{11}$$

$$W_3(j) = \frac{1}{\sum_{i=1}^{n} |M_{i,j} - d_i|} \tag{12}$$

$$W_4(j) = \frac{1}{\sum_{i=1}^{n} |D_{i,j}^2 - d_i^2|} \tag{13}$$

where $D_{i,j}$ and $M_{i,j}$ are the Euclidean distance and the Manhattan distance between AN i and vertex j, respectively.

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{14}$$

$$M_{i,j} = |x_i - x_j| + |y_i - y_j| \tag{15}$$

In order to estimate the final BN's position, a weighted centroid is calculated with the weights and the coordinates of the vertices as in (16).

$$[x_0, y_0] = \left[\frac{\sum_{j=1}^{4} W_a(j) \cdot x_j}{\sum_{j=1}^{4} W_a(j)}, \frac{\sum_{j=1}^{4} W_a(j) \cdot y_j}{\sum_{j=1}^{4} W_a(j)} \right] \tag{16}$$

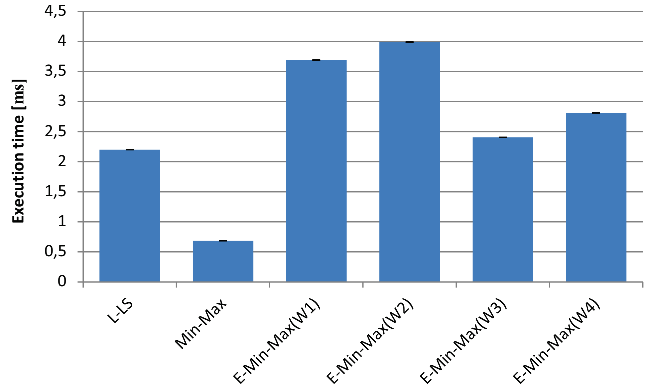

Fig. 3. Computation time of low-complexity localization algorithms in a 8-bit microcontroller (n=4). The confidence intervals (95%) are smaller than 10 μs in all cases.

TABLE I
NUMBER OF ARITHMETIC OPERATIONS

Algorithm	Additions or Subtractions	Mult.	Div.	Square roots
L-LS	n·13-16	n·11-2	4	—
Min-Max	n·4+2	—	2	—
E-Min-Max(W_1)	n·20+9	n·8+8	6	n·4
E-Min-Max(W_2)	n·20+9	n·12+8	6	n·4
E-Min-Max(W_3)	n·20+9	8	6	—
E-Min-Max(W_4)	n·20+9	n·9+8	6	—

III. ANALYSIS

A. Complexity

The method used for solving the R-LS function (2) requires the execution of many operations and therefore it should not be used by computation-limited sensor nodes. The mentioned low-complexity algorithms were investigated by considering the number of arithmetic operations executed in each case. Table I shows this information, which is classified in four groups: additions or subtractions, multiplication (Mult.), divisions (Div.) and square roots. In all cases, the complexity increases with increasing number of ANs (n).

The required computation time strongly depends on the hardware architecture of the sensor nodes. In general, the execution of square roots is not desirable because of its significant computation time. It can be also considered that divisions, cost more time that multiplications, additions or subtractions.

As mentioned, the original Min-Max contains very few arithmetic operations. However, note that the comparisons done by the functions $min()$ and $max()$ have to be considered additionally.

The exact solution of L-LS seems to be efficient in terms of complexity. Here, the inverse of the square matrix $A^T A$ (see (7)) can be easily obtained by multiplying the determinant of $A^T A$ by its adjugate. The operations required in determining the adjugate of this square matrix are not considered in the table I because of its simplicity. It consists of exchanging the place of elements of the diagonal and the negation of the other two elements of the matrix.

E-Min-Max requires extra operations to estimate the weights for the vertices. Specifically, E-Min-Max(W_1) and E-Min-Max(W_2) include square roots to estimate $D_{i,j}$ between the vertices and the anchors as shown in (14). In order to avoid the execution of square roots and minimize the computation time, E-Min-Max(W_4) uses $D_{i,j}^2$ in the estimation of the weights. E-Min-Max(W_3) uses Manhattan distances reducing the required number of multiplications.

By generating timestamps at the beginning and at the end of the execution of the algorithms, it was possible to measure their corresponding computation times in a sensor node. We used the 802.15.4 sensor node RCB230 [7], which contains the 8-bit microcontroller Atmega1281. In our experiments, uniform random numbers (double precision) between 0 and 150000 were generated for representing the coordinates of 4 ANs and their corresponding distances to the BN. More than 1500 execution times were averaged for each algorithm.

As expected, the method used for solving the R-LS function registered an average computation time of circa 487ms (N_1=100 and N_2=5). Fig.3 refers to the values obtained from the other algorithms. It can be seen, that the more operations an algorithm executes the longer its execution time, being these results consistent with table I.

With the information about the average duration of the algorithms it is possible to estimate their energy consumptions. In [10] we measured the energy consumption of the node RCB230 in different operation modes. This node consumes circa 30.42 mW when the transceiver sleeps and the microcontroller is active.

Note that if the energy consumption of the entire localization process should be calculated, the energy consumed in the distance determination should be also considered. There are different techniques to approximate the distance between two nodes, like the Received Signal Strength Indicator (RSSI), Time of Arrival (TOA) as well as Time Difference of Arrival (TDOA). More information about the signaling required in such techniques can be found in [8] [9].

B. Position accuracy

By using simulation (MATLAB), the localization algorithms were analyzed in a reference scenario (100m x100m), which contains four anchor placed at $[33m, 67m]$, $[33m, 33m]$, $[67m, 33m]$ and $[67m, 67m]$, respectively. The blind node took 10000 different positions. In each position more than 50 estimations were averaged. The position error is considered as the Euclidean distance between the real coordinates of the node and the estimated position.

We used a generic model for considering the distance error. Here, the estimated distance (d_i) is modeled by adding a zero mean normaly distributed random variable (χ) to the real distance (d_r). The standard deviation σ is defined as a percentage (p) of the real distance.

$$d_i = d_r + \chi_\sigma \tag{17}$$

$$\sigma = p \cdot d_r \tag{18}$$

In this way the distance error increases with the measured distance. A similar relationship between error and distance is also described in other distance error models, like in [6]. In our evaluation p takes values from 0% up to 20%.

The average position error in the complete scenario is depicted in Fig.4. As expected, when there is no error in the distances, the exact localization algorithms do not produce position errors.

In case of error in the distance determination, the high complexity R-LS can achieve the best performance in terms of average position accuracy.

The original Min-Max presents the highest error because of the described internal zone problem. E-Min-Max improves Min-Max in all cases. Furthermore, E-Min-Max with weight W_2 outperforms L-LS when p is greater than 5%.

The advantage of the use of Manhattan distance is its low-complexity, which is very attractive for resource-limited sensor nodes. However, the results indicate that E-Min-Max(W_3) does not produce satisfactory results in terms of average position accuracy.

Fig.5 shows the average position error but only in the internal zone delimited by the ANs. Here, the original Min-Max improves the performance of L-LS when $p > 10\%$. E-Min-Max with weights W_1 and W_4 not only achieves a similar accuracy as R-LS, but also outperforms L-LS even for small distance errors (p=5%). When the distance errors are very big, E-Min-Max and the original Min-Max provide similar results. This is due to the fact that the weights used by E-Min-Max do not correctly represent the similarity between the coordinates of the vertices and the real coordinates of the BN.

IV. Conclusion

In this paper we present an extended version of the low-complexity localization algorithm Min-Max. By using simulation we compared four versions of E-Min-Max with the original version of Min-Max and two exact localization algorithms: R-LS and L-LS.

The exact algorithm R-LS has the better performance related to position accuracy. However, the method used in R-LS means the execution of a lot of operations, which is not efficient in case of decentralized calculation. On the contrary, L-LS executes few operations and given its accuracy, this should be used when the distance errors are very small.

In the complete scenario, E-Min-Max improves the position accuracy compared with the original one. This is principally due to the fact that Min-Max trends to locate its estimation inside of the "internal zone", which is the region delimited by the ANs of the periphery. This can lead to a high position error when the BN is outside of this region. E-Min-Max can minimize this problem but at expense of the execution of extra operations.

E-Min-Max(W_2) can improve the accuracy of Min-Max and L-LS, when the relative distance error is higher than 5%. The drawback is that it needs the execution of square roots and more arithmetic operations compared to L-LS.

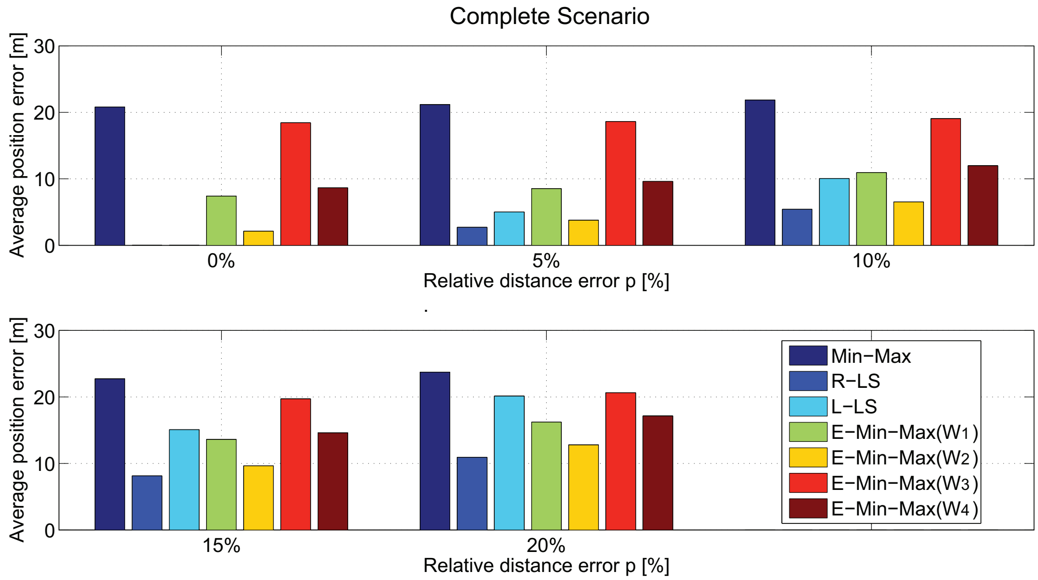

Fig. 4. Average position error in the scenario. The confidence intervals (95%) are smaller than 0.05m in all cases [11]

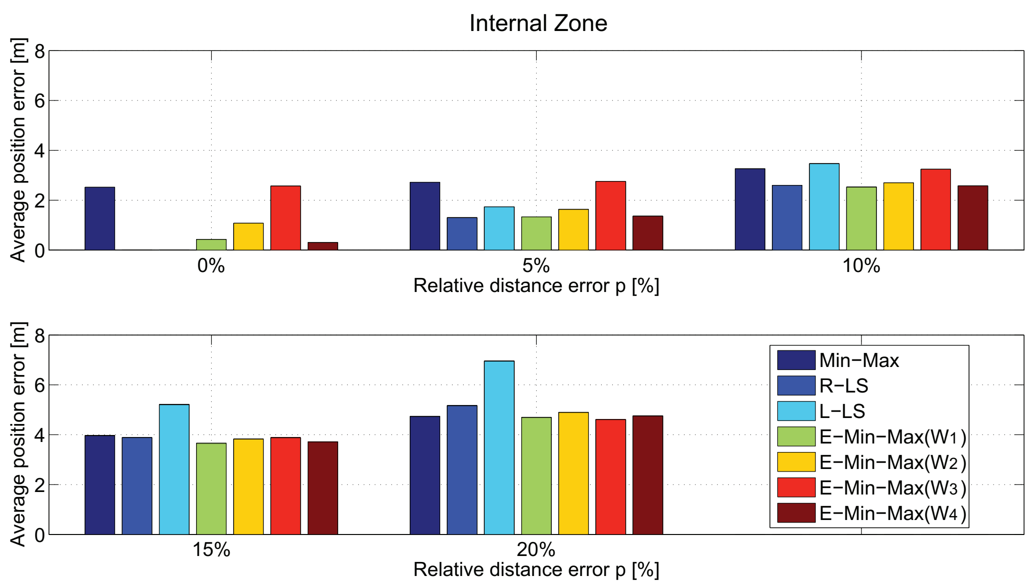

Fig. 5. Average position error in the internal zone. The confidence intervals (95%) are smaller than 0.05m in all cases [11]

In the internal zone, Min-Max operates better than L-LS when the relative distance error p is higher than 10%. Furthermore, E-Min-Max(W_4) and E-Min-Max(W_1) can improve L-LS even for small distance errors. Note that E-Min-Max(W_4) is more efficient than E-Min-Max(W_1) because it does not include square roots in the calculation.

In our reference scenario E-Min-Max(W_3) did not show an significant improvement in terms of position accuracy.

REFERENCES

[1] K. Langendoen and N. Reijers, *Distributed Localization in Wireless Sensor Networks: a Quantitative Comparison*, The International Journal of Computer and Telecommunications Networking - Special issue: Wireless sensor networks archive, Volume 43 Issue 4, 15 November 2003.

[2] A. Savvides, H. Park and M. Srivastava, *The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems*, First ACM International Workshop on Wireless Sensor Networks and Application (WSNA), Atlanta, GA, 2002, pp. 112121.

[3] A. Beck, P. Stoica, and J. Li, *Exact and Approximate Solutions of Source Localization Problems*, IEEE Transactions on Signal processing, Vol. 56, No. 5, May 2008

[4] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, *Weighted Centroid Localization in WLAN-based Sensor Networks*, In Proceedings of the IEEE International Symposium on Intelligent Signal Processing (WISP'2007)

[5] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2005.

[6] J. Chóliz, A. Hernández-Solana, A. Valdovinos, *Evaluation of Algorithms*

for UWB Indoor Tracking, Workshop on Positioning Navigation and Communication (WPNC), Dresden, April 2011.

[7] *Datasheet sensor node RCB230*, available at:http://www.dresden-elektronik.de/shop/prod68.html.

[8] G. Mao and B. Fidan, *Localization Algorithms and Strategies for Wireless Sensor Networks*, Information Science Reference, 2009.

[9] Z. Sahinoglu,S. Gezici and I. GuvencG, *Ultra-wideband Positioning Systems*, Cambridge University, 2008.

[10] J.J. Robles, S. Tromer, M. Quiroga and R. Lehnert, *Enabling Low-power Localization for Mobile Sensor Nodes*, In Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN) 2010, Zürich, Switzerland, Sep 2010.

[11] J. Supervía, *Investigation of the Optimal Anchor Location in Indoor Scenarios*, Student thesis, Supervisor: J. J. Robles, Chair for Telecommunications, Technische Universität Dresden, Nov 2011.