Particle Swarm Optimization for Auto-localization of Nodes in Wireless Sensor Networks

Stefania Monica and Gianluigi Ferrari

Wireless Ad-hoc and Sensor Networks Laboratory
Department of Information Engineering
University of Parma, 43124 Parma, Italy
stefania.monica@studenti.unipr.it, gianluigi.ferrari@unipr.it
http://wasnlab.tlc.unipr.it/

Abstract. In this paper, we consider the problem of auto-localization of the nodes of a static Wireless Sensor Network (WSN) where nodes communicate through Ultra Wide Band (UWB) signaling. In particular, we investigate auto-localization of the nodes assuming to know the position of a few initial nodes, denoted as "beacons". In the considered scenario, we compare the location accuracy obtained with the widely used Two-Stage Maximum-Likelihood algorithm with that achieved with an algorithm based on Particle Swarming Optimization (PSO). Accurate simulation results show that the latter can significantly outperform the former.

Keywords: Auto-localization, Particle Swarm Optimization, Maximum-Likelihood Algorithms.

1 Introduction

The location of sources in an indoor environment is of great interest because it has applications in many areas, such as monitoring of people in hospitals or in high security areas, search of victims or firefighters in emergency situations, home security, and finding people or vehicles in a warehouse. Wireless Sensor Networks (WSNs) are a leading option to address this problem, since they combine low to medium rate communications with positioning capabilities [1]. As a matter of fact, radio signal exchanges between nodes enables them to estimate the distance to each other, by extracting some physical quantities, such as the Received Signal Strength (RSS), the Angle Of Arrival (AOA), or the Time Of Flight (TOF) from the signals travelling between them.

Assuming to know the exact positions of a sufficiently large number of nodes, the position of a new node can be estimated by measuring its distances from a few nodes with known positions. Of course, wireless communications are affected by noise, especially in indoor environments, because of non-line-of-sight, multipath and multiple access interference. Ultra Wide Band (UWB) signaling

M. Tomassini et al. (Eds.): ICANNGA 2013, LNCS 7824, pp. 456–465, 2013.

seems a promising technology in this area, since the large bandwidth allows to resolve multipath components and the high time resolution improves their ranging capability (thus making the position estimate more accurate) [2].

In this paper, we consider an indoor scenario, which may be, for instance, a warehouse, in which a certain number of fixed Anchor Nodes (ANs) is used to locate Target Nodes (TNs), such as moving people and vehicles. In order to guarantee accurate TN estimation in every accessible point inside the building, which might be very large, a huge number of accurately positioned ANs would be necessary and this could be very demanding also from an economic point of view. Moreover, if the geometry of the warehouse changes, (e.g., for varying quantities of stored goods) the ANs might be replaced and/or new fixed ANs might be added. In order to overcome this problem, we consider auto-localization of the ANs under the assumption of initially knowing exactly the positions of only a few ANs, denoted as "beacons". UWB signaling is used and we consider a TOF approach to estimate the distances between pairs of nodes. In particular, we focus on a Time Difference Of Arrival (TDOA) approach, which is based on the estimation of the difference between the arrival times of signals traveling between each node to locate and nodes with known position (beacons or nodes whose position has already been estimated).

Many location estimate techniques, based on range measurement, have been proposed in the literature. Among them, it is worth recalling iterative methods, such as those based on Taylor-series expansion [3], or the steepest-descent algorithm [4]. These methods guarantee fast convergence for an initial value close to the true solution (which is often difficult to obtain in practice), but they are computationally expensive and convergence is not guaranteed, since, for instance, ignoring higher order terms in the Taylor-series expansion may lead to significant errors. To overcome these limitations, closed-form algorithms have been studied, such as least-square methods, approximated maximum-likelihood method [5], the plane intersection method [6] and the Two-Stages Maximum-Likelihood (TSML) method [7] [8]. In particular, the TSML method is one of the most commonly used, since it has been proved that it can attain the Cramer-Rao lower bound [9]. By observing that the initial system of equations of the TSML can be re-interpreted as an optimization problem, we thus solve it through the use of Particle Swarming Optimization (PSO). The proposed approach is shown to perform better than the TSML method.

This paper is organized as follows. In Section 2 the TSML method and the PSO algorithm are described. In Section 3 numerical results are presented. Section 4 concludes the paper.

2 Description of the Scenario

Throughout the paper, it is assumed that all the ANs lay on a plane, e.g., the ceiling of a warehouse. As anticipated in Section 1, a sufficient (but small) number of known "beacons" is used to get the position estimate of each AN with unknown position. Define: $\underline{s}_i = [x_i, y_i]^T, \forall i = 1, \ldots, M$ as the (known) positions of M ANs (the "beacons"); $\underline{u}_e = [x_e, y_e]^T$ as the true position of a generic AN whose position needs to be estimated; $\underline{\hat{u}}_e = [\hat{x}_e, \hat{y}_e]^T$ as its estimated position. The true and estimated distances between the $i-$th beacon and the AN with position to be estimated are, respectively:

$$r_i = \sqrt{(\underline{u}_e - \underline{s}_i)^T(\underline{u}_e - \underline{s}_i)} \qquad \hat{r}_i = \sqrt{(\underline{\hat{u}}_e - \underline{s}_i)^T(\underline{\hat{u}}_e - \underline{s}_i)}. \tag{1}$$

As we are considering UWB signaling, it can be shown that $\hat{r}_i = r_i + \nu_i$, where $\nu_i = \varepsilon_i + b$ where $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ and ε_i is independent from ε_j if $i \neq j$, and b is the synchronization bias [10]. Moreover, according to [10], the standard deviation of the position error estimation between two UWB nodes can be approximated as a linear function of the distance between them, namely

$$\sigma_i = \sigma_0 r_i + \beta. \tag{2}$$

In the following, the values $\sigma_0 = 0.01$ m and $\beta = 0.08$ m are considered. These values are obtained in [10] by considering Channel Model 3 described in [11] and the energy detection receiver presented in [12], which is composed by a band-pass filter followed by a square-law device and an integrator, in which the integration interval was set equal to $T_s = 1$ s. Therefore, the results presented in the following hold under these channel and receiver conditions.

2.1 TSML Method

The position estimation is carried on by using a simplification of the two step algorithm proposed in [8]. Note that at least 4 ANs positions must be known in order to start applying the following algorithm. Defining $\Delta_{1i} = r_i - r_1$, $\forall\ i = 2, \ldots, M$, and observing that $r_i^2 = (\Delta_{1i} + r_1)^2$, the following TDOA non-linear equations can be derived:

$$\Delta_{1i}^2 + 2\Delta_{1i} r_1 = -2x_i x_e - 2y_i y_e + x_e^2 + y_e^2 - K_i - r_1^2 \qquad i = 2, \ldots, M \tag{3}$$

where $K_i = x_i^2 + y_i^2$. When using estimated distances instead of real ones, the set of equations (3) can be written as

$$\underline{\underline{\hat{G}}}\,\underline{\hat{u}}_e = \underline{\hat{h}} \tag{4}$$

where

$$\underline{\underline{\hat{G}}} = -\begin{pmatrix} x_2 - x_1 & y_2 - y_1 & \hat{\Delta}_{12} \\ x_3 - x_1 & y_3 - y_1 & \hat{\Delta}_{13} \\ \vdots & \vdots & \vdots \\ x_M - x_1 & y_M - y_1 & \hat{\Delta}_{1M} \end{pmatrix} \qquad \underline{\hat{h}} = \frac{1}{2}\begin{pmatrix} K_1 - K_2 + \hat{\Delta}_{12}^2 \\ \vdots \\ K_1 - K_M + \hat{\Delta}_{1M}^2 \end{pmatrix} \tag{5}$$

and $\hat{\Delta}_{1i} = \hat{r}_i - \hat{r}_1$, $\forall i = 2, \ldots, M$. The system (4) would be a linear system in the three unknowns x_e, y_e and r_1 if r_1 did not depend on x_e and y_e (which is instead the case, since by definition $r_1 = \sqrt{(x_e - x_1)^2 + (y_e - y_1)^2)}$. In order to take into account this dependence, the solution is determined in 2 steps. First, it is supposed that x_e, y_e, and r_1 are three independent variables and the (linear) system is solved by using the Least Square (LS) technique. Defining $\hat{\underline{\phi}}_1 = [\hat{x}_e, \hat{y}_e, \hat{r}_1]^T$ and $\underline{\phi}_1 = [x_e, y_e, r_1]^T$ leads to the error vector

$$\underline{\psi} = \hat{\underline{\underline{G}}}(\hat{\underline{\phi}}_1 - \underline{\phi}_1). \tag{6}$$

The Maximum Likelihood (ML) solution of (6) is

$$\hat{\underline{\phi}}_1 = (\hat{\underline{\underline{G}}}^T \underline{\underline{\Psi}}^{-1} \hat{\underline{\underline{G}}})^{-1} \hat{\underline{\underline{G}}}^T \underline{\underline{\Psi}}^{-1} \hat{\underline{h}} \tag{7}$$

where

$$\underline{\underline{\Psi}} = \mathbb{E}[\underline{\psi}\,\underline{\psi}^T] = \underline{\underline{B}}\,\underline{\underline{Q}}\,\underline{\underline{B}}, \tag{8}$$

and $\underline{\underline{B}} = \mathrm{diag}(r_2, \ldots, r_M)$, $\underline{\underline{Q}} = \mathbb{E}[\underline{\varepsilon}_1 \underline{\varepsilon}_1^T]$ where $(\underline{\varepsilon}_1)_j = \hat{\Delta}_{1j} - \Delta_{1j}$ [8]. Omitting non linear perturbation, from (7) one obtains

$$\hat{\underline{u}}_e - \underline{u}_e = (\underline{\underline{G}}^T \underline{\underline{\Psi}}^{-1} \underline{\underline{G}})^{-1} \underline{\underline{G}}^T \underline{\underline{\Psi}}^{-1} (\hat{\underline{h}} - \hat{\underline{\underline{G}}}\,\underline{u}_e) = (\underline{\underline{G}}^T \underline{\underline{\Psi}}^{-1} \underline{\underline{G}})^{-1} \underline{\underline{G}}^T \underline{\underline{\Psi}}^{-1} (\underline{\psi}).$$

Since $\mathbb{E}[\underline{\psi}] = \underline{0}$, it can be noticed that $\mathbb{E}[\hat{\underline{u}}_e] = \underline{u}_e$ and the covariance matrix of $\hat{\underline{\phi}}_1$ is then [9]

$$\mathrm{cov}(\hat{\underline{\phi}}_1) \triangleq \mathbb{E}[(\hat{\underline{\phi}}_1 - \underline{\phi}_1)(\hat{\underline{\phi}}_1 - \underline{\phi}_1)^{\mathrm{T}}] = (\underline{\underline{\mathrm{G}}}^{\mathrm{T}} \underline{\underline{\Psi}}^{-1} \underline{\underline{\mathrm{G}}})^{-1}. \tag{9}$$

Now taking into account the relation between x_e, y_e, and r_1 the following set of equations is obtained:

$$\underline{\psi}' = \underline{h}' - \underline{\underline{G}}' \underline{\phi}_2 \tag{10}$$

where

$$\underline{h}' = [([\hat{\underline{\phi}}_1]_1 - x_1)^2, ([\hat{\underline{\phi}}_1]_2 - y_1)^2, [\hat{\underline{\phi}}_1]_3^2]^T \qquad \underline{\underline{G}}' = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\underline{\phi}_2 = [(x_e - x_1)^2, (y_e - y_1)^2]^T.$$

The ML solution of (10) is

$$\hat{\underline{\phi}}_2 = (\underline{\underline{G}}'^T \underline{\underline{\Psi}}'^{-1} \underline{\underline{G}}')^{-1} \underline{\underline{G}}'^T \underline{\underline{\Psi}}'^{-1} \underline{h}' \tag{11}$$

where $\underline{\underline{\Psi}}' \triangleq \mathbb{E}[\underline{\psi}' \underline{\psi}'^T] = 4\underline{\underline{B}}' \mathrm{cov}(\hat{\underline{\phi}}_1) \underline{\underline{\mathrm{B}}}'$, $\underline{\underline{B}}' = \mathrm{diag}(x_e - x_1, y_e - y_1, r_1)$ and [9]

$$\mathrm{cov}(\hat{\underline{\phi}}_2) = (\underline{\underline{\mathrm{G}}}'^{\mathrm{T}} \underline{\underline{\Psi}}'^{-1} \underline{\underline{\mathrm{G}}}')^{-1}. \tag{12}$$

This leads to

$$\underline{u}_e = \underline{\underline{U}} \left[\sqrt{[\hat{\underline{\phi}}_2]_1}, \sqrt{[\hat{\underline{\phi}}_2]_2} \right]^T + \underline{s}_1$$

where $\underline{\underline{U}} = \text{diag}[\text{sgn}(\underline{\hat{\phi}}_1 - \underline{s}_1)]$. The covariance matrix of the error is then given by

$$\underline{\underline{\Phi}} = \frac{1}{4}(\underline{\underline{B}}''^{-1}\text{cov}(\underline{\hat{\phi}}_2)\underline{\underline{\text{B}}}''^{-1}) \tag{13}$$

where $B'' = \text{diag}(\underline{u} - \underline{s}_1)$ [9].

2.2 PSO Algorithm

The starting point for the previous method was the system (4) defined in Section 2.1. Through simple algebraic manipulations, it can be written as

$$\underline{\underline{A}}\,\underline{\hat{u}}_e = \underline{\hat{t}} \tag{14}$$

where

$$\underline{\underline{A}} = -2\begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ \vdots & \vdots \\ x_M - x_1 & y_M - y_1 \end{pmatrix} \qquad \underline{\hat{t}} = \begin{pmatrix} \hat{r}_2^2 - \hat{r}_1^2 + K_1 - K_2 \\ \hat{r}_3^2 - \hat{r}_1^2 + K_1 - K_3 \\ \vdots \\ \hat{r}_M^2 - \hat{r}_1^2 + K_1 - K_M \end{pmatrix}. \tag{15}$$

Notice that in this way, the measurements affected by noise only appear in vector $\underline{\hat{t}}$, while matrix $\underline{\underline{A}}$ cointains known parameters. On the contrary, in (4) both the matrix $\underline{\underline{\hat{G}}}$ and the vector $\underline{\hat{h}}$ contain noisy data. The solution of the system (14) can be found by formulating it as an optimization problem, with the following solution:

$$\underline{u}_e = \text{argmin}_{\underline{u}} F(\underline{u}) \tag{16}$$

where

$$F(\underline{u}) \triangleq ||\hat{t} - \underline{\underline{A}}\,\underline{u}||.$$

To solve this problem, the PSO algorithm, introduced in [13], can be used. According to this algorithm, the set of potential solutions of an optimization problem can be modeled as a swarm of particles, and the aim is to produce computational intelligence (thus to guide all the particles towards the optimal solution of the given problem), by exploiting social interactions between individuals [14]. It is assumed that the swarm is composed by S individuals, and that every particle i, $i = 1, \ldots, S$ at any given instant t is associated with a position $\underline{x}^i(t)$ in the region of interest and with a velocity $\underline{v}^i(t)$, which are both randomly initialized at the beginning with values $\underline{x}^i(0)$ and $\underline{v}^i(0)$ and which are updated at each iteration [15]. Moreover, it is supposed that the entire system has a memory, which allows each particle to know, at every step, not only its own best position reached so far, but also the best position among the ones reached by any other particle in the swarm (or by any other particle in a given neighbourhood of the swarm) in previous iterations. Each particle also keeps track of the values of the function to optimize corresponding both to its best position and to the global best position. These values are used to update the velocity (and thus the position) of every particle in each step. The updating rule for the velocity of particle i is [16]

$$\underline{v}^i(t+1) = \omega(t)\underline{v}^i(t) + c_1 R_1(t)(\underline{y}^i(t) - \underline{x}^i(t)) + c_2 R_2(t)(\underline{y}(t) - \underline{x}^i(t)) \qquad i = 1, \ldots, S \tag{17}$$

where $\omega(t)$ is a weight called *inertial factor*, c_1 and c_2 are positive real parameters called *cognition parameter* and *social parameter*, respectively, $R_1(t)$ and $R_2(t)$ are random variable drawn at each step from the uniform $(0, 1)$ distribution and $\underline{y}^i(t)$ and $\underline{y}(t)$ are the position of the $i-$th particle with the best objective function and the position of the best (among all particles) objective function reached until instant t [14]. They can be described as

$$\begin{aligned} \underline{y}^i(t) &= \mathrm{argmin}_{\underline{z}\in\{\underline{x}^i(0),\ldots\,\underline{x}^i(t)\}} F(\underline{z}) \\ \underline{y}(t) &= \mathrm{argmin}_{\underline{z}\in\{\underline{y}^1(t),\ldots\,\underline{y}^S(t)\}} F(\underline{z}). \end{aligned} \tag{18}$$

The meaning of formula (17) is to add to the previous velocity (which is weightened by means of a multiplicative factor) a stochastic combination of the direction to the best position of the $i-$th particle and to the best global position. The definition of the velocity given in (17) is then used to update the position of the $i-$th particle, according to the rule

$$\underline{x}^i(t+1) = \underline{x}^i(t) + \underline{v}^i(t) \qquad i = 1, \ldots, S.$$

This process is iterated until a stopping criterion (which might be the achievement of a satisfying value of F or a maximum number of iterations) is met. The position of the particle which best suits the optimization requirements in the last iteration is then considered as the optimal solution.

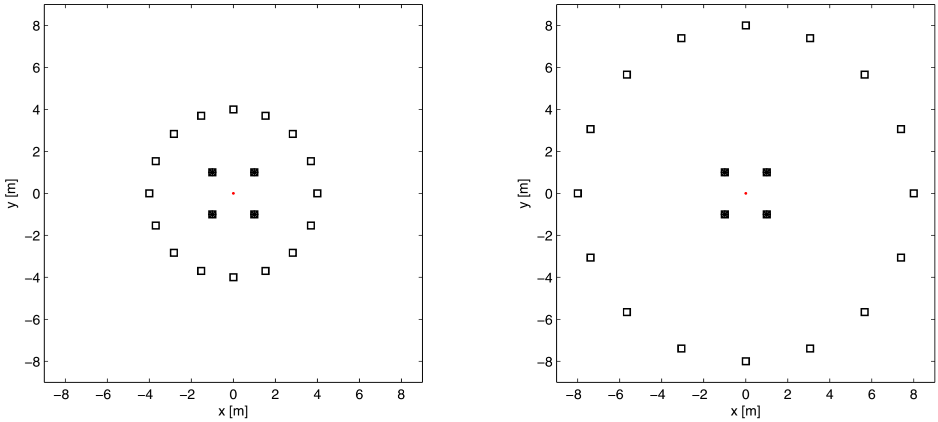
In Section 3, this algorithm is applied to the function $F(\underline{u})$ defined in (16). The number of particles in the simluations is $S = 40$, and the number of iterations has been set to 50. The parameters c_1 and c_2 in (17) are both set equal to 2, which is a recommended choice since it makes the weights for social and cognition parts to be 1 on average [13]. The inertial factor $\omega(t)$ has been chosen to be a decreasing function in the number of iterations. As a matter of fact, a large value of the inertial factor corresponds to low dependece of the solution on initial population, and any good optimization algorithm should possess more exploitation ability at the beginning. Decreasing the value of $\omega(t)$ reduces the capability of PSO to exploit new areas, thus making the method more simliar to a local search as the number of iteration increases, which is a good property for an optimization algorithm. In the following simulations it is assumed that the initial value of the inertial factor is $\omega(0) = 0.9$ and that it decreases linearly to 0.5, reached in the last iteration.

3 Simulation Results

In this section, we compare, through simulations, the two localization approaches, namely TSML and PSO, described in Section 2. The performances of the algorithms are evaluated in terms of the Mean Square Error (MSE), which is defined as follows:

$$\mathrm{MSE} \triangleq \mathbb{E}[(\hat{x}_e - x_e)^2 + (\hat{y}_e - y_e)^2]. \quad (19)$$

While in [17] the impact of the number of beacons is investigated, we now want to investigate the impact of the distance between beacons and ANs to be estimated. Two scenarios are considered, in which 4 beacons are used to estimate 16 unknown ANs around them. In the first scenario, the distance between the baricenter of the beacons and the remaining ANs is 4 m, while in the second case a longer distance (8 m) is considered. This two scenarios are shown in Fig. 1 (a.) and Fig. 1 (b.), respectively.



(a.) (b.)

Fig. 1. Possible scenarios where beacons (*full squares*) and ANs with unknown position (*empty squares*) are represented. The distance between the baricenter of the beacons (*red dot*) and the ANs is 4 m in (a.), and 8 m in (b.).

Fig. 2 (a.) represents the MSE relative to each AN when using only the beacons to get the location estimate. It can be noticed that the TSML method is far too unreliable, while the accuracy obtained using the PSO algorithm is satisfactory. The resulting estimated positions of the ANs are represented in Fig. 2 (b.), both in case of TSML method and PSO algorithm. Instead of considering only the beacons, a possible idea which can improve the accuracy is to consider the already estimated ANs as known ones, and to use them to calculate the position of the remaining nodes. Fig. 2 (c.) represents the MSE relative to each AN when using this second strategy and comparing the results with the ones of Fig. 2 (a.) shows that performances are improved. In particular, the accuracy of the location estimate when using TSML method significantly improves as the number of ANs assumed to be known increases. Less significant but still remarkable improvement can be noticed also in the behaviour of the MSE when using the PSO algorithm.

Notice that in both cases the PSO gives a better approximation of the real position of the ANs. The position estimates of the ANs obtained when following this strategy are represented in Fig. 2 (d.).

Fig. 3 represents the analogous results when the scenario is the one described in Fig. 1 (b.), so when the ANs are further from the beacons. Once again, from Fig. 3 (a.) and Fig. 3 (c.) it can be noticed that the PSO algorithm outperforms the TSML mehtod. Comparing these results with the analogous ones represented in Fig. 2 (a.) and Fig. 2 (c.), respectively, shows that a bigger distance between beacons and ANs leads to worst accuracy, especially when the TSML method is used. As a matter of fact, even when also the already estimated ANs are used to localize the remaining ones (as in Fig. 3 (c.)), the accuracy obtained with this method is not satisfactory. On the other hand, the accuracy obtained when using the PSO algorithm is still good, and in this case the distance between ANs does not seem to impact much on the solutions. Fig. 3 (b.) and Fig. 3 (d.) represent the obtained position estimate, analogous to Fig. 2 (b.) and Fig. 2 (d.).

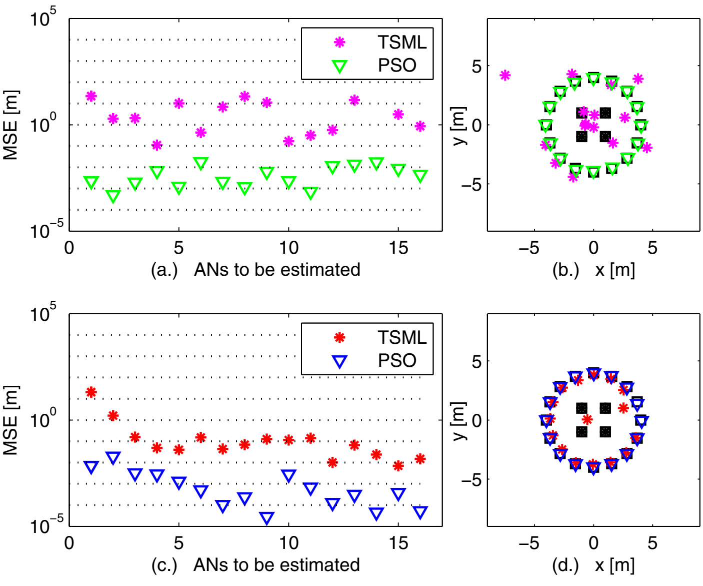


Fig. 2. The MSE of the ANs relative to the scenario described in Fig. 1 (a.) is plotted in (a.) and (c.). Fig. 2 (a.) refers to the case when only the beacons are used to estimate the position of all the ANs, both when TSML method is used (*magenta dots*) and when PSO algorithm is used (*green triangles*). The resulting position estimate are represented in Fig. 2 (b.), both in case of TSML method (*magenta dots*) and PSO algorithm (*green triangles*). Fig. 2 (c.) refers to the case in which also already estimated ANs are used to get the position of the remaining ones, both when TSML method is used (*red dots*) and when PSO algorithm is used (*blue triangles*). The resulting position estimate are represented in Fig. 2 (d.), both in case of TSML method (*red dots*) and PSO algorithm (*blue triangles*).

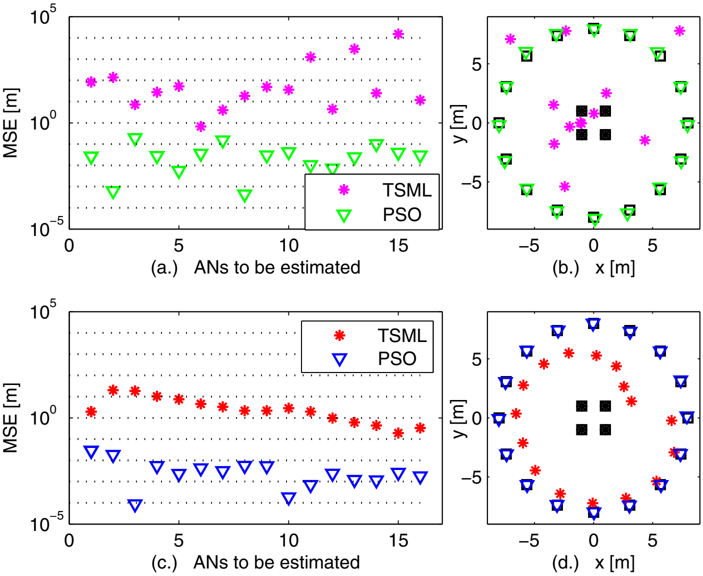


Fig. 3. The MSE of the ANs relative to the scenario described in Fig. 1 (a.) is plotted in (a.) and (c.). Fig. 2 (a.) refers to the case when only the beacons are used to estimate the position of all the ANs, both when TSML method is used (*magenta dots*) and when PSO algorithm is used (*green triangles*). The resulting position estimate are represented in Fig. 2 (b.), both in case of TSML method (*magenta dots*) and PSO algorithm (*green triangles*). Fig. 2 (c.) refers to the case in which also already estimated ANs are used to get the position of the remaining ones, both when TSML method is used (*red dots*) and when PSO algorithm is used (*blue triangles*). The resulting position estimate are represented in Fig. 2 (d.), both in case of TSML method (*red dots*) and PSO algorithm (*blue triangles*).

4 Conclusion

Two different approaches to UWB-signaling-based auto-localization of nodes in a static WSN have been considered. Besides solving the non-linear system of the localization equations by means of the TSML method, which is widely used for this kind of application, the trasformation of the original problem into an optimization one allows to solve it by means of the PSO algorithm. Our results show that the novel approach based on the use of the PSO algorithm allows to achieve better accuracy in the position estimate.

Acknowledgment. This work is supported by Elettric80 (http://www.elettric80.it).

References

1. Gezici, S., Poor, H.V.: Position estimation via ultra-wide- band signals. Proc. IEEE 97(2), 386–403 (2009)
2. Zhang, J., Orlik, P.V., Sahinoglu, Z., Molisch, A.F., Kinney, P.: UWB systems for wireless sensor networks. Proc. IEEE 97(2), 313–331 (2009)
3. Wade, H.F.: Position-location solutions by Taylor-series estimation. IEEE Trans. Aerosp. Electron. Syst. AES-12(2), 187–194 (1976)
4. Mensing, C., Plass, S.: Positioning algorithms for cellular networks using TDOA. In: Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2006, vol. 4 (May 2006)
5. Shen, G., Zetik, R., Thomä, R.S.: Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment. In: Proceedings of the 5th Workshop on Positioning, Navigation and Communication, WPNC 2008 (2008)
6. Schmidt, R.O.: A new approach to geometry of range difference location. IEEE Trans. Aerosp. Electron. Syst. AES-8(6), 821–835 (1972)
7. Chan, Y., Ho, K.C.: A simple and efficient estimator for hyperbolic location. IEEE Trans. Signal Process 42(8), 1905–1915 (1994)
8. Ho, K.C., Xu, W.: An accurate algebraic solution for moving source location using TDOA and FDOA measurements. IEEE Trans. Signal Process. 52(9), 2453–2463 (2004)
9. Ho, K.C., Lu, X., Kovavisaruch, L.: Source localization using TDOA and FDOA measurements in the presence of receiver location errors: analysis and solution. IEEE Trans. Signal Process. 55(2), 684–696 (2007)
10. Busanelli, S., Ferrari, G.: Improved ultra wideband-based tracking of twin-receiver automated guided vehicles. Journal of Integrated Computer-Aided Engineering 19(1), 3–22 (2012)
11. Molisch, A.F., Cassioli, D., Chong, C.-C., Emami, S., Fort, A., Kannan, B., Karedal, J., Kunisch, J., Schantz, H.G., Siwiak, K., Win, M.Z.: A comprehensive standardized model for ultrawideband propagation channels. IEEE Trans. Antennas Propagat. 54(11), 3151–3166 (2006)
12. Dardari, D., Chong, C.C., Win, M.Z.: Threshold-based time-of-arrival estimators in uwb dense multipath channels. IEEE Trans. Commun. 56(8), 1366–1378 (2008)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE International Conf. on Neural Networks, Perth, Australia, IEEE Service Center, Piscataway (1995)
14. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm Intelligence Journal 1(1) (2007)
15. Eberhart, R., Kermedy, J.: A new optimizer using particles swarm theory. In: Proc. Sixth International Symposium on Micro Machine and Hmm Science, Nagoya, Japan, IEEE Service Center, Piscataway (1995)
16. Shi, Y., Eberhart, R.: A modied particle swarm optimizer. In: Proc. IEEE International Conference on Evolutionary Computation, Piscataway, NJ, pp. 69–73 (1999)
17. Monica, S., Ferrari, G.: Impact of the number of beacons in PSO-based auto-localization in UWB networks. To appear in Proceedings of EvoApplications (2013)