Impact of the Number of Beacons in PSO-Based Auto-localization in UWB Networks

Stefania Monica and Gianluigi Ferrari

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

Abstract. In this paper, we focus on auto-localization of nodes in a static wireless network, under the assumption of known position of a few initial nodes, denoted as "beacons". Assuming that Ultra Wide Band (UWB) signals are used for inter-node communications, we analyze the impact of the number of beacons on the location accuracy. Three different approaches to localization are considered, namely: the Two-Stage Maximum-Likelihood (TSML) method ; the Plane Intersection (PI) method, and Particle Swarming Optimization (PSO). Simulation results show that PSO allows to obtain accurate postion estimates with a small number of beacons, making it an attractive choice to implement effective localization algorithm.

Keywords: Particle Swarm Optimization (PSO), Auto-localization, Two-Stage Maximum-Likelihood (TSML) Algorithms, Least Square (LS) Method, Ultra Wide Band (UWB) Signaling.

1 Introduction

The problem of locating sources in an indoor environment has been widely studied since it has many applications in various areas, such as: monitoring of people in hospitals or in high security areas; search for victims or firefighters in emergency situations; home security; and locating people or vehicles in a warehouse. The use of wireless networks is an attractive option in this field, as they combine low-to-medium rate communications with positioning capabilities [6]. As a matter of fact, the distance between each pair of nodes can be estimated by sending signals between them and by extracting from these signals some physical quantities, such as the received signal strength, the angle of arrival, or the time of flight. The position of a node can then be estimated by using the distance measurements from a certain number of nodes with known positions, denoted as "beacons." The accuracy of the obtained position estimate depends on the

A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2013, LNCS 7835, pp. 42–51, 2013.

errors that affect wireless communications between nodes, which, in indoor environments, are mainly due to non-line-of-sight, multipath, and multiple access interference. To reduce the impact of these sources of errors (thus obtaining a more accurate position estimate), Ultra Wide Band (UWB) signaling is a promising technology, since, on one hand, the large bandwidth allows to penetrate through obstacles and to resolve multipath components and, on the other hand, the high time resolution improves the ranging capability [14].

In this paper, the considered scenario is a warehouse in which fixed Anchor Nodes (ANs) with known positions are used to locate Target Nodes (TNs), such as people and vehicles. A very large number of ANs might be necessary to guarantee accurate TN estimation in every accessible point inside a large building, and their accurate positioning 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. To overcome this problem, we focus on the auto-localization of the ANs assuming to know the exact positions of only a few beacons. The number of beacons should be small, in order to reduce the cost of installation, but sufficiently large to guarantee a reliable position estimate of other ANs. The focus of this work is to investigate the impact of the number of beacons on the system performance.

We assume to use UWB signaling. The distances between pairs of nodes are estimated by means of a time-based approach. More precisely, we consider 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 beacons.

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 [5], or the steepest-descent algorithm [9]. These techniques 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 (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 the Plane Intersection (PI) method [12] and the Two-Stage Maximum-Likelihood (TSML) method [2]. These methods can be re-interpreted as possible approaches to solve a minimization problem. According to this perspective, the location estimate can then be found by means of optimization techniques. More precisely, by re-formulating the initial system of equations of the TSML in terms of an optimization problem, we solve it through the use of Particle Swarming Optimization (PSO). In this work we show that the proposed approach can perform better than the PI and the TSML methods.

This paper is organized as follows. In Section 2, the PI and TSML methods and the PSO algorithm are described. In Section 3 numerical results, relative to the impact of the number of beacons on the performances of the different algorithms, are presented. Section 4 concludes the paper.

2 Scenario Description

Throughout the paper, we assume that all the ANs lay on a plane, which could be, for instance, the ceiling of a warehouse. We suppose that M beacons, whose coordinates are denoted by $\underline{s}_i = [x_i, y_i]^T, \forall i = 1, \ldots, M$ are used to get the position estimate of each AN with unknown position. In order to apply the algorithms outlined in the remainder of this section, a necessary condition is that $M \geq 4$.

If we define $\underline{u}_e = [x_e, y_e]^T$ as the true position of a generic AN (whose position needs to be estimated) and $\underline{\hat{u}}_e = [\hat{x}_e, \hat{y}_e]^T$ as its estimated position, then the true and estimated distances between the $i-$th beacon and the AN of interest are, respectively:

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

Since we are considering UWB signaling, it can be shown that $\hat{r}_i \simeq r_i + \nu_i$, where $\nu_i = \varepsilon_i + b$, $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, ε_i is independent from ε_j if $i \neq j$ $(j = 1, \ldots, M)$, and b is a synchronization bias [1]. Moreover, according to [1], the standard deviation σ_i of the position error estimation between two UWB nodes can be approximated as a linear function of the distance between them, namely

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

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

In the remainder of this section, the following notation will be used:

$$\Delta_{1i} = r_i - r_1, \quad \forall\, i = 2, \ldots, M \qquad K_i = x_i^2 + y_i^2 \quad \forall\, i = 1, \ldots, M. \tag{3}$$

2.1 TSML Method

According to the TSML method, each TDOA measurement identifies a hyperbola which the source has to belong to. Therefore, given a set of TDOA measurements, the position estimate can be determined by solving the system of equations corresponding to these hyperbolas using a Least Square (LS) technique [2]. Observing that $r_i^2 = (\Delta_{1i} + r_1)^2$, from (1) and (3) the following TDOA non-linear equations can be derived:

$$\Delta_{1i}^2 + 2\Delta_{1i} r_1 = -2x_i x_e - 2y_i y_e + x_e^2 + y_e^2 - K_i - r_1^2 \qquad i = 2, \ldots, M. \tag{4}$$

When using estimated distances instead of the real ones, defining $\underline{\hat{\phi}}_1 = [\underline{\hat{u}}_e^T, \hat{r}_1]^T$, the set of equations (4) can be written as

$$\underline{\underline{\hat{G}}}\,\underline{\hat{\phi}}_1 = \underline{\hat{h}} \tag{5}$$

where

$$\underline{\underline{\hat{G}}} = -\begin{pmatrix} x_2 - x_1 & y_2 - y_1 & \hat{\Delta}_{12} \\ x_3 - x_1 & y_3 - y_1 & \hat{\Delta}_{13} \\ \vdots & \vdots & \vdots \\ x_M - x_1 & y_M - y_1 & \hat{\Delta}_{1M} \end{pmatrix} \qquad \underline{\hat{h}} = \frac{1}{2}\begin{pmatrix} K_1 - K_2 + \hat{\Delta}_{12}^2 \\ \vdots \\ K_1 - K_M + \hat{\Delta}_{1M}^2 \end{pmatrix} \tag{6}$$

and $\hat{\Delta}_{1i} = \hat{r}_i - \hat{r}_1$, $\forall i = 2, \ldots, M$. This is a non-linear system because, according to (1), $\hat{r}_1$ depends on $\hat{x}_e$ and $\hat{y}_e$. The solution of (5) is determined in 2 steps. First, $\hat{x}_e$, $\hat{y}_e$, and $\hat{r}_1$ are assumed to be three independent variables and the (linear) system is solved by using the LS method. Consider the error vector

$$\underline{\psi} \triangleq \underline{\underline{\hat{G}}}(\underline{\hat{\phi}}_1 - \underline{\phi}_1) \tag{7}$$

where $\underline{\phi}_1 = [x_e, y_e, r_1]^T$. The Maximum Likelihood (ML) estimate of $\underline{\hat{\phi}}_1$ is

$$\underline{\hat{\phi}}_1 = (\underline{\underline{\hat{G}}}^T \underline{\underline{\Psi}}^{-1} \underline{\underline{\hat{G}}})^{-1} \underline{\underline{\hat{G}}}^T \underline{\underline{\Psi}}^{-1} \underline{\hat{h}} \tag{8}$$

and $\underline{\underline{\Psi}} \triangleq \mathrm{cov}(\underline{\psi}) = \underline{\underline{B}}\,\underline{\underline{Q}}\,\underline{\underline{B}}$ where $\underline{\underline{B}} = \mathrm{diag}(r_2, \ldots, r_M)$, $\underline{\underline{Q}} = \mathbb{E}[\underline{\varepsilon}_1 \underline{\varepsilon}_1^T]$ and $(\underline{\varepsilon}_1)_j = \hat{\Delta}_{1j} - \Delta_{1j}$ [7]. It can be shown that cov $(\underline{\hat{\phi}}_1) = (\underline{\underline{G}}^T \underline{\underline{\Psi}}^{-1} \underline{\underline{G}})^{-1}$ [2]. Taking into account the relation between $\hat{x}_e$, $\hat{y}_e$, and $\hat{r}_1$, i.e., equation (1), the following set of equations can be obtained:

$$\underline{\psi}' = \underline{\hat{h}}' - \underline{\underline{G}}' \underline{\hat{\phi}}_2 \tag{9}$$

where

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

$$\underline{\hat{\phi}}_2 = [(\hat{x}_e - x_1)^2, (\hat{y}_e - y_1)^2]^T.$$

The ML solution of (9) is

$$\underline{\hat{\phi}}_2 = (\underline{\underline{G}}'^T \underline{\underline{\Psi}}'^{-1} \underline{\underline{G}}')^{-1} \underline{\underline{G}}'^T \underline{\underline{\Psi}}'^{-1} \underline{\hat{h}}' \tag{10}$$

where $\underline{\underline{\Psi}}' \triangleq \mathrm{cov}(\underline{\psi}') = 4\underline{\underline{B}}' \mathrm{cov}(\underline{\hat{\phi}}_1) \underline{\underline{B}}'$ and $\underline{\underline{B}}' = \mathrm{diag}(x_e - x_1, y_e - y_1, r_1)$ [7]. This leads to the following position estimate

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

where $\underline{\underline{U}} = \mathrm{diag}[\mathrm{sgn}(\underline{\hat{\phi}}_1 - \underline{s}_1)]$.

2.2 PI Method

According to the PI method, introduced in [12], any triple of ANs (which leads to a pair of TDOA measurements) identifies the major axes of a conic, a focus of which is the position of the source. Given at least two triples of ANs, the position estimate can then be determined by solving the system given by the equations of the corresponding axes. By considering the axes identified by $\{\underline{s}_1, \underline{s}_2, \underline{s}_k\}$, $k = 3, \ldots, M$ the system can be written as

$$\underline{\underline{\hat{A}}}\, \hat{\underline{u}}_e = \hat{\underline{b}} \tag{11}$$

where

$$\underline{\underline{\hat{A}}} = \begin{pmatrix} x_{21}\hat{\Delta}_{13} - x_{31}\hat{\Delta}_{12} & y_{21}\hat{\Delta}_{13} - y_{31}\hat{\Delta}_{12} \\ x_{21}\hat{\Delta}_{14} - x_{41}\hat{\Delta}_{12} & y_{21}\hat{\Delta}_{14} - y_{41}\hat{\Delta}_{12} \\ \vdots & \vdots \\ x_{21}\hat{\Delta}_{1M} - x_{M1}\hat{\Delta}_{12} & y_{21}\hat{\Delta}_{1M} - y_{M1}\hat{\Delta}_{12} \end{pmatrix} \tag{12}$$

and

$$\hat{\underline{b}} = \frac{1}{2} \begin{pmatrix} -\hat{\Delta}_{12}\hat{\Delta}_{13}(\hat{\Delta}_{13} - \hat{\Delta}_{12}) + (K_1 - K_2)\hat{\Delta}_{13} - (K_1 - K_3)\hat{\Delta}_{12} \\ -\hat{\Delta}_{12}\hat{\Delta}_{14}(\hat{\Delta}_{14} - \hat{\Delta}_{12}) + (K_1 - K_2)\hat{\Delta}_{14} - (K_1 - K_4)\hat{\Delta}_{12} \\ \vdots \\ -\hat{\Delta}_{12}\hat{\Delta}_{1M}(\hat{\Delta}_{1M} - \hat{\Delta}_{12}) + (K_1 - K_2)\hat{\Delta}_{1M} - (K_1 - K_M)\hat{\Delta}_{12} \end{pmatrix}. \tag{13}$$

where $x_{j1} \triangleq x_1 - x_j$, $y_{j1} \triangleq y_1 - y_j$, $j = 2, \ldots, M$, and K_j and $\hat{\Delta}_{1j}$ are defined in (3). The LS solution of (11) is then given by

$$\hat{\underline{u}}_e = (\underline{\underline{\hat{A}}}^T \underline{\underline{\hat{A}}})^{-1} \underline{\underline{\hat{A}}}^T \hat{\underline{b}}. \tag{14}$$

2.3 PSO Algorithm

The starting point for the TSML method was the system (5) in Subsection 2.1. Through simple algebraic manipulations, this system can be written as

$$\underline{\underline{B}}\, \hat{\underline{u}}_e = \hat{\underline{t}} \tag{15}$$

where

$$\underline{\underline{B}} = -2 \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ \vdots & \vdots \\ x_M - x_1 & y_M - y_1 \end{pmatrix} \qquad \hat{\underline{t}} = \begin{pmatrix} \hat{r}_2^2 - \hat{r}_1^2 + K_1 - K_2 \\ \hat{r}_3^2 - \hat{r}_1^2 + K_1 - K_3 \\ \vdots \\ \hat{r}_M^2 - \hat{r}_1^2 + K_1 - K_M \end{pmatrix}. \tag{16}$$

Notice that, while in (5) both the matrix $\underline{\underline{\hat{G}}}$ and the vector $\hat{\underline{h}}$ contain noisy data, in (15) the measurements affected by noise only appear in vector $\hat{\underline{t}}$, while the matrix $\underline{\underline{B}}$ cointains known parameters. By interpreting the system (15) as an optimization problem, its solution can be expressed as follows:

$$\underline{\hat{u}}_e = \text{argmin}_{\underline{u}} ||\hat{t} - \underline{\underline{B}}\,\underline{u}||. \tag{17}$$

The PSO algorithm, introduced in [8], can be used to solve this problem. According to this algorithm, the set of potential solutions of an optimization problem is modeled as a swarm of S particles, which are guided towards the optimal solution of the given problem, by exploiting "social" interactions between individuals [11]. It is assumed that every particle i in the swarm ($i = 1, \ldots, S$) 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 [4]. It is also assumed that the system has memory, so that, at every instant, each particle knows 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 in the previous iterations. Each particle also keeps track of the values of the function to optimize in correspondence to both its best position and the global best position. These values are used to update the velocity and the position of every particle at each iteration. More precisely, the velocity of particle i is updated at consecutive iterations, according to the rule [13]

$$\underline{v}^{(i)}(t+1)=\omega(t)\underline{v}^{(i)}(t)+c_1 R_1(t)(\underline{y}^{(i)}(t)-\underline{x}^{(i)}(t))+c_2 R_2(t)(\underline{y}(t)-\underline{x}^{(i)}(t)) \quad i = 1, \ldots, S \tag{18}$$

where: $\omega(t)$ is denoted as *inertial factor*; c_1 and c_2 are positive real parameters denoted as *cognition* and *social* parameters, respectively; $R_1(t)$ and $R_2(t)$ are random variables uniformly distributed in $(0, 1)$; 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 particle with the best (among all particles) objective function reached until instant t [11]. In the considered minimization problem (17), they can be described as

$$\begin{aligned} \underline{y}^{(i)}(t) &= \text{argmin}_{\underline{z} \in \{\underline{x}^{(i)}(0), \ldots \underline{x}^{(i)}(t)\}} ||\hat{t} - \underline{\underline{B}}\,\underline{z}|| \\ \underline{y}(t) &= \text{argmin}_{\underline{z} \in \{\underline{y}^{(1)}(t), \ldots \underline{y}^{(S)}(t)\}} ||\hat{t} - \underline{\underline{B}}\,\underline{z}||. \end{aligned} \tag{19}$$

The idea behind the iterative step (18) is to add to the previous velocity of particle i (which is weighted by means of a multiplicative factor) a stochastic combination of the direction to its best position and to the best global position. The definition of the velocity given in (18) is then used to update the position of the $i-$th particle, according to the following rule:

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

Possible stopping conditions for the PSO algorithm can be the achievement of a satisfying value of the function to be minimized or a given (maximum) number of iterations. At the end of the algorithm, the solution is the position of the particle which best suits the optimization requirements in the last iteration.

The application of PSO to the considered localization problem is better explained in the next section.

3 Simulation Results

In this section, the three localization approaches described in Section 2, namely TSML, PI, and PSO, are compared through MATLAB based simulations compliant with the propagation model introduced in Section 2. In all cases, the performance is evaluated in terms of Mean Square Error (MSE) between true and estimated positions, i.e.:

$$\text{MSE} \triangleq \mathbb{E}[(\hat{x}_e - x_e)^2 + (\hat{y}_e - y_e)^2]. \tag{20}$$

In the following simulations, the MSE is obtained from the average of 100 independent runs.

The PSO algorithm has been implemented by setting both the parameters c_1 and c_2 in (18) to 2. This choice makes the weights for social and cognition parts to be, on average, equal to 1. The inertial factor $\omega(t)$ has been chosen to be a decreasing function of the number of iterations, in order to guarantee low dependence of the solution on the initial population and to reduce the exploitation ability of the algorithm, making the method more simliar to a local search, as the number of iterations increases [13]. In the following, it is assumed that the initial value of the inertial factor is $\omega(0) = 0.9$ and that it decreases linearly to 0.4, reached at the 50-th iteration, i.e. the last one according to the stopping criterion we chose. A population of 40 particles is considered since previous simulations showed that this value is large enough to guarantee an accurate solution and that incrementing it does not lead to significant improvements.

We investigate through simulation the minimum number of beacons that are needed to obtain a reliable estimate of the ANs positions. First, we consider a partition of the entire plane on which the ANs lay into squares, whose edges are 10 m long. Without loss of generality, we restrict our analysis to a single square. The considered scenario is shown in Fig. 1 (b) and Fig. 1 (d), where circles represent beacons while squares represent ANs with unknown positions. In the scenario shown in Fig. 1 (b), 8 out of the total 36 ANs are assumed to be beacons and their known coordinates are then used to get the position estimate of the remaining 28 ANs. In Fig. 1 (a), the MSE corresponding to each AN is represented. In this case, by comparing the MSE of the three algorithm, it can be noticed that there are no significant differences in the order of magnitude of the error and the three algorithms guarantee an accurate position estimate of all the ANs. On the other hand, in the scenario represented in Fig. 1 (d) only 4 beacons are assumed to be used to estimate the positions of the remaining 32 ANs. As can be noticed from Fig. 1 (c), in this case both the TSML and the PI methods lead to a far inaccurate positioning estimate for many ANs while the accuracy obtained when using the PSO algorithm is still good. Moreover, by comparing the behaviour of PSO algorithm when 8 beacons are used with the one obtained with only 4 beacons shows that the MSE has the same order of magnitude in both cases. It can then be concluded that 4 beacons are not enough to obtain a reliable estimate when using the TSML and the PI methods, but they are sufficient to guarantee an accurate postion estimate when the PSO algorithm is used.

We now consider a scenario composed by a corridor 40 m long and 5 m wide, as shown in Fig. 2 (b) and Fig. 2 (d). In the scenario represented in Fig. 2 (b), there are 16 beacons out of the total 44 ANs and this allows to obtain an accurate position estimate with all the three approaches previously described, namely the TSML and the PI method and the PSO algorithm, as shown in Fig. 2 (a).

As in the previous scenario, reducing the number of beacons, as in Fig. 2 (d), leads to significantly worse values of the MSE corresponding both to TSML and PI method, without changing the accuracy of the position estimate obtained via the PSO algorithm, as shown in Fig. 2 (c).

Therefore, we can observe that the PSO algorithm can be successfully applied with at least half the number of beacons, which allows to save money and time in the accurate positioning of fixed ANs.

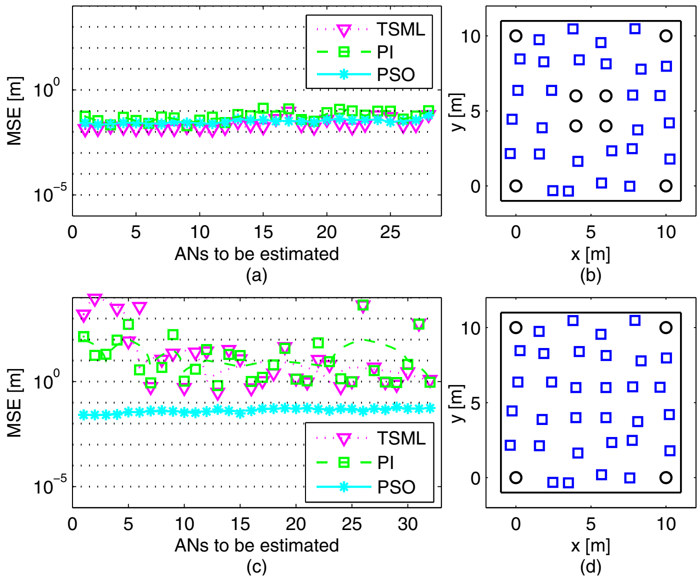

Fig. 1. Fig. 1 (b) and Fig. 1 (d) represent the beacons (*circles*) and the ANs whose positions need to be estimated (*squares*). In Fig. 1 (a.) the MSE of the ANs relative to the scenario described in Fig. 1 (b.) is plotted, corresponding to TSML method (*triangles*), PI method (*squares*) and PSO algorithm (*dots*). In each case, the interpolation lines (*dotted* lines for TSML, *dashed* lines for PI, *solid* lines for PSO) are shown. Fig. 1 (c.) represents the MSE realtive to each AN when the considered scenario is the one described in Fig. 1 (d.). In this case the PSO algorithm outperforms the TSML and the PI method, showing that 4 beacons are enough to obtain an accurate position estimate when using PSO.

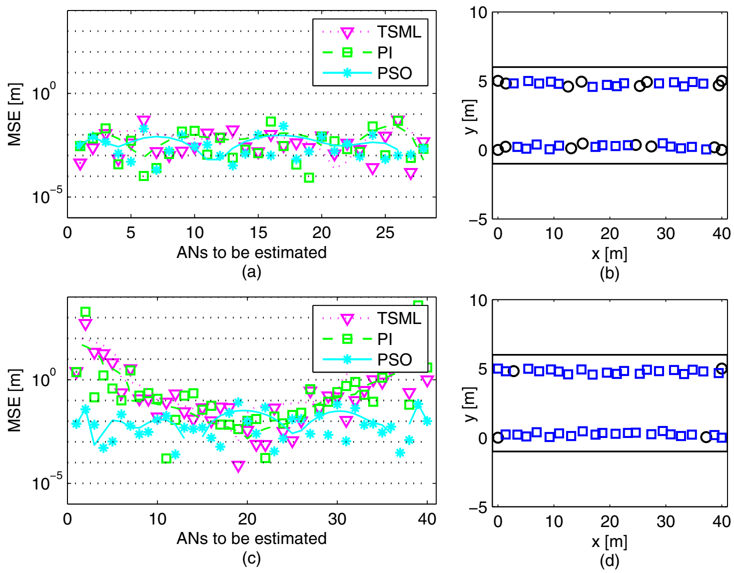

Fig. 2. Fig. 2 (b) and Fig. 2 (d) represent the beacons (*circles*) and the ANs whose positions need to be estimated (*squares*). In Fig. 2 (a) the estimation MSEs of the ANs in the scenario described in Fig. 2 (b) are shown, using TSML method (*triangles*), PI method (*squares*) and PSO algorithm (*dots*). In this case the order of magnitude of the error is the same for all the three algorithms. Fig. 2 (c) represents the MSE realtive to each AN when the considered scenario is the one described in Fig. 2 (d). In this case the PSO algorithm outperforms the TSML and the PI method, showing that 4 beacons are enough to obtain an accurate position estimate when using PSO.

4 Conclusion

In order to evaluate the impact of the number of beacons on localization accuracy, three approaches to UWB-signaling-based auto-localization of nodes in a static wireless network have been considered. Besides solving the localization problem by means of the TSML and the PI methods, which are widely used for this purpose, the original system of non-linear equations of the TSML method has been re-formulated in terms of an optimization problem, which is then solved by means of PSO. Our results show that the PSO approach guarantees a good accuracy in the position estimate with a smaller number of known beacons, allowing to reduce the installation cost of the entire localization system.

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

References

1. Busanelli, S., Ferrari, G.: Improved ultra wideband-based tracking of twin-receiver automated guided vehicles. Integrated Computer-Aided Engineering 19(1), 3–22 (2012)
2. Chan, Y., Ho, K.C.: A simple and efficient estimator for hyperbolic location. IEEE Trans. Signal Process. 42(8), 1905–1915 (1994)
3. 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)
4. 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)
5. Foy, W.H.: Position-location solutions by Taylor-series estimation. IEEE Trans. Aerosp. Electron. Syst. AES-12(2), 187–194 (1976)
6. Gezici, S., Poor, H.V.: Position estimation via ultra-wide- band signals. Proc. IEEE 97(2), 386–403 (2009)
7. 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)
8. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE International Conf. on Neural Networks, Perth, Australia. IEEE Service Center, Piscataway (1995)
9. Mensing, C., Plass, S.: Positioning algorithms for cellular networks using TDOA. In: 2006 IEEE International Conference on Proceedings of the Acoustics, Speech and Signal Processing, ICASSP 2006, vol. 4 (May 2006)
10. 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)
11. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm Intelligence Journal 1(1) (2007)
12. Schmidt, R.O.: A new approach to geometry of range difference location. IEEE Trans. Aerosp. Electron. Syst. AES-8(6), 821–835 (1972)
13. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proc. IEEE International Conference on Evolutionary Computation, Piscataway, NJ, pp. 69–73 (1999)
14. 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)