A Swarm Intelligence Approach to 3D Distance-based Indoor UWB Localization

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 the application of Ultra Wide Band (UWB) technology to the problem of locating static nodes in three-dimensional indoor environments, assuming to know the positions of a few nodes, denoted as "beacons." The localization algorithms which are considered throughout the paper are based on the Time Of Arrival (TOA) of signals traveling between pairs of nodes. In particular, we propose to apply the Particle Swarm Optimization (PSO) algorithm to solve the localization problem and we compare its performance with that of the Two-Stage Maximum-Likelihood (TSML) algorithm. Simulation results show that the former allows achieving accurate position estimates even in scenarios where, because of ill-conditioning problems associated with the network topology, TSML fails.

Keywords: Wireless Sensor Networks, Localization, Time Of Arrival (TOA), Maximum-Likelihood, Particle Swarm Optimization

1 Introduction

In the last years, the interest on Wireless Sensor Networks (WSNs) has become more and more evident, as they promise to be a scalable and low-cost technology which can be used to solve many practical challenges. They consist of a number of wirelessly connected nodes, each of which has one or more sensors to measure some physical quantities such as temperature, pressure, acceleration, etc. Typical applications of WSNs include home security, military surveillance, environmental monitoring, assistance for old people and patients in hospitals, and industrial process control [1].

Among the wide variety of fields in which WSNs are involved, in this paper we focus on accurate target localization, through wireless communications, in indoor environments. This is a challenging and, at the same time, very interesting problem: the use of WSNs is an attractive option to address it, as they combine low-to-medium rate communications and low power consumption with positioning capabilities. As a matter of fact, it is possible to estimate the distances between pairs of nodes by extracting some physical quantities—such as the Time Of Flight (TOF), the Received Signal Strength (RSS), or the Angle Of Arrival (AOA), from the signals travelling between them.

Many localization estimate techniques have been proposed in the literature, such as: iterative methods based on Taylor series expansion [2] or the steepest-descent algorithm [3], which guarantee fast convergence only for an initial estimate value close to the true solution (often difficult to obtain in real applications); closed-form methods, such as the Circumference Intersection (CI) algorithms [4], the Plane Intersection (PI) algorithm [5] [6], and the Two-Stage Maximum-Likelihood (TSML) algorithm [7] [8]. These "geometrical" methods typically involve linear or non-linear systems of equations, which can become ill-conditioned (for instance, if the considered beacons lay on the same line or plane) and, thus, lead to wrong position estimates. In [9] [10], it is shown that the initial system of equations of the TSML algorithm can be reformulated in terms of an optimization problem and solved through the use of Particle Swarm Optimization (PSO). Even if the TSML algorithm is particuarly interesting, as it can attain the Cramer-Rao lower bound [11], in [9] [10] the PSO algorithm is shown to outperform the TSML algorithm. In [12], the use of the PSO algorithm is investigated to estimate the positions, with the use of a few "beacons" and considering Time Difference Of Arrival (TDOA) approaches, of nodes laying on a plane. We remark that the use of the PSO algorithm for localization purposes is not novel [13].

In this paper, we extend [12] by considering three dimensional node position estimation. In order to keep the derivation more tractable, we investigate Time Of Arrival (TOA) localization strategies. In particular, under the assumption of knowing the positions of a few nodes, denoted as beacons, the positions of other nodes, placed in three dimensional scenarios, are estimated. As already observed in two-dimensional scenarios, the PSO algorithm outperforms the TSML algorithm. The impact of the swarm size on the performance of the PSO-based localization strategy is investigated.

This paper is organized as follows. In Section 2, the considered scenario is described. In Section 3, the considered localization algorithms are described. In Section 4, the simulation-based performance results are presented. Section 5 concludes the paper.

2 Scenario

Assume to have M beacons and denote their coordinates are $\underline{s}_i = [x_i, y_i, z_i]^T, \forall i \in \{1, \ldots, M\}$—we remark that the condition $M \geq 4$ needs to be satisfied in order to apply the localization algorithms described later. We indicate as K_i the square of the Euclidean norm of the vector which identifies the position of the $i-$th node, namely:

$$K_i = x_i^2 + y_i^2 + z_i^2 \qquad i \in \{1, \ldots, M\}. \tag{1}$$

We denote as $\underline{u} = [x, y, z]^T$ the true position, unknown and to be estimated, of a generic node and as $\hat{\underline{u}} = [\hat{x}, \hat{y}, \hat{z}]^T$ its estimated position. Then, the true and estimated distances between the $i-$th beacon ($i \in \{1, \ldots, M\}$) and the currently considered node are, respectively:

$$r_i = \sqrt{(\underline{u} - \underline{s}_i)^T(\underline{u} - \underline{s}_i)} \qquad \hat{r}_i = \sqrt{(\hat{\underline{u}} - \underline{s}_i)^T(\hat{\underline{u}} - \underline{s}_i)}. \tag{2}$$

In order to provide a statistical model for the errors on the estimated distance, the wireless communication protocol comes into the picture. Since we consider indoor environments, where the main sources of errors are non-line-of-sight propagation, multipath, and multiple access interference, we propose to use Ultra Wide Band (UWB) signaling, which can theoretically reduce the impact of these phenomena. As a matter of fact, UWB signals' large bandwidth allows penetrating through obstacles and resolving multipath components. Moreover, the high time resolution typical of UWB signals improves the ranging capability [14]. According to [15], the estimated distances can be modeled as

$$\hat{r}_i \simeq r_i + \nu_i \qquad i \in \{1, \ldots, M\} \tag{3}$$

where:

$$\nu_i = \varepsilon_i + b$$

$\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$; ε_i is independent from ε_j if $i \neq j$, $j \in \{1, \ldots, M\}$; and b is a synchronization bias. Moreover, the standard deviation σ_i of the position error estimate between two UWB nodes can be approximated as a linear function of the distance between them, namely $\sigma_i \simeq \sigma_0 r_i + \beta$. In the following, the values $\sigma_0 = 0.01$ m and $\beta = 0.08$ m are considered. These values are obtained in [15] by considering Channel Model 3 described in [16] and the energy detection receiver presented in [17], 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.

3 Localization Algorithms

In this section, we first describe the TOA-based TSML algorithm. Then, we formulate the PSO algorithm applied to the minimization problem obtained by properly reinterpreting the TOA equations of the TSML algorithm.

3.1 Two-Stage Maximum-Likelihood Algorithm

Each measurement of the distance between a given beacon and a node provides the equation of the circumference, centered at the beacon, on which the node may lay. Therefore, considering all measurements of the distances between the node of interest and the beacons, the following quadratic system can be obtained:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = \hat{r}_1^2 \\ \ldots \\ (x - x_M)^2 + (y - y_M)^2 + (z - z_M)^2 = \hat{r}_M^2. \end{cases} \tag{4}$$

Defining a new variable as $\hat{n} \triangleq ||\underline{\hat{u}}||^2$ (where $||\cdot||$ is the Euclidean norm), the system (4) can be written in matrix notation as

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

where

$$\underline{\underline{G}} \triangleq -2 \begin{pmatrix} x_1 & y_1 & z_1 & -0.5 \\ x_2 & y_2 & z_2 & -0.5 \\ \vdots & \vdots & \vdots & \vdots \\ x_M & y_M & z_M & -0.5 \end{pmatrix} \qquad \underline{\hat{w}} \triangleq \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{n} \end{pmatrix} \qquad \underline{\hat{h}} \triangleq \begin{pmatrix} \hat{r}_1^2 - K_1 \\ \hat{r}_2^2 - K_2 \\ \vdots \\ \hat{r}_M^2 - K_M \end{pmatrix}. \tag{6}$$

The solution of (5) can be found in two steps and we refer the interested reader to [8] for all details. The idea behind the solution is that (5) can be interpreted as a linear system (which is not the case since $\hat{n} = \hat{x}^2 + \hat{y}^2 + \hat{z}^2$, i.e., the last component of the solution vector depends on the first three ones) and, then, be solved by using a Maximum Likelihood (ML) appraoch, obtaining [8]:

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

where $\underline{\underline{\mathit{\Psi}}} \triangleq 4\underline{\underline{B}}\,\underline{\underline{Q}}\,\underline{\underline{B}}$, $\underline{\underline{B}} = \mathrm{diag}(r_1, \ldots, r_M)$, and $\underline{\underline{Q}} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_M^2)$.

Then, in order to take into account the fact that the elements of $\underline{\hat{w}}$ are not independent, a second step is necessary, and corresponds to solving the following system:

$$\underline{\underline{G}}'\,\underline{\hat{w}}' = \underline{\hat{h}}'$$

where

$$\underline{\underline{G}}' \triangleq \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad \underline{\hat{h}}' \triangleq \begin{pmatrix} \underline{\hat{w}}_1^2 \\ \underline{\hat{w}}_2^2 \\ \underline{\hat{w}}_3^2 \\ \underline{\hat{w}}_4 \end{pmatrix} \qquad \underline{\hat{w}}' \triangleq \begin{pmatrix} \underline{\hat{x}}^2 \\ \underline{\hat{y}}^2 \\ \underline{\hat{z}}^2 \end{pmatrix}.$$

The ML estimate of $\underline{\hat{w}}'$ ca be finally written as [8]

$$\underline{\hat{w}}' = (\underline{\underline{G}}^T \underline{\underline{\mathit{\Phi}}}^{-1} \underline{\underline{G}})^{-1} \underline{\underline{G}}^T \underline{\underline{\mathit{\Phi}}}^{-1} \underline{\hat{h}'} \tag{8}$$

where $\underline{\underline{\mathit{\Phi}}} \triangleq 4\underline{\underline{B}}'(\underline{\underline{G}}^T \underline{\underline{\mathit{\Psi}}}^{-1} \underline{\underline{G}})^{-1} \underline{\underline{B}}'$ and $\underline{\underline{B}}' \triangleq \mathrm{diag}(\hat{w}_1, \hat{w}_2, \hat{w}_3, 0.5)$. Finally, the position estimate is found to be [8]

$$\underline{\hat{u}} = \underline{\underline{U}} \left[\sqrt{\underline{\hat{w}}_1'}, \sqrt{\underline{\hat{w}}_2'}, \sqrt{\underline{\hat{w}}_3'} \right]^T$$

where $\underline{\underline{U}} = \mathrm{diag}(\mathrm{sgn}(\underline{\hat{w}}))$.

3.2 Particle Swarm Optimization

The system (4) can be written, in matrix notation, as

$$\underline{I}_1 \underline{\hat{u}}^T \underline{\hat{u}} + \underline{\underline{A}}\,\underline{\hat{u}} = \underline{\hat{k}} \tag{9}$$

where: $\underline{I}_1$ is a $M \times 1$ vector with all elements equal to 1; $\underline{\underline{A}}$ is a $M \times 3$ matrix whose columns are the first three columns of the matrix $\underline{\underline{G}}$ in (5); and the $i-$th element of $\underline{\hat{k}}$ is $\hat{r}_i^2 - K_i$. Instead of solving the quadratic system (9), we re-interpret it as the following minimization problem:

$$\underline{\hat{u}} = \mathrm{argmin}_{\underline{u}} F(\underline{u}) \tag{10}$$

where $F(\underline{u})$ is the fitness function defined as follows:

$$F(\underline{u}) \triangleq ||\hat{k} - (\underline{I}_1 \underline{\hat{u}}^T \underline{\hat{u}} + \underline{\underline{A}}\, \underline{\hat{u}})||.$$

The PSO algorithm, introduced in [18], is an iterative method which can be used to solve optimization problems such as the one in (10). The set of potential solutions of the considered problem is modeled as a swarm of S particles. At any given instant t, each particle $i \in \{1, \ldots, S\}$ in the swarm, 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 $\underline{v}^{(i)}(0)$. The particles are "guided" towards the optimal solution. In fact, the updating rules for the position and the velocity of each particle are meant to simulate "social" interactions between individuals [19]. More precisely, according to the most general formulation of the PSO algorithm, the velocity of particle i is updated, at each iteration, according to the following rule [20]:

$$\begin{aligned} \underline{v}^{(i)}(t+1) = \omega(t)\underline{v}^{(i)}(t) &+ c_1 R_1(t)(\underline{y}^{(i)}(t) - \underline{x}^{(i)}(t)) \\ &+ c_2 R_2(t)(\underline{y}(t) - \underline{x}^{(i)}(t)) \qquad i \in \{1, \ldots, S\} \end{aligned} \tag{11}$$

where: $\omega(t)$ is denoted as *inertial factor*; c_1 and c_2 are positive real parameters denoted as *cognition* and *social* parameters, respectively; and $R_1(t)$ and $R_2(t)$ are random variables uniformly distributed in $(0, 1)$. Finally, $\underline{y}^{(i)}(t)$ and $\underline{y}(t)$ are the positions of the $i-$th particle with the best fitness function and the position of the particle with the best (among all particles) fitness function reached until instant t [19].

The idea behind the updating rule (11) for the velocities is to add to the previous velocity of each particle in the swarm (weighted by means of a multiplicative factor $\omega(t)$) a stochastic combination of the direction to its best position (corresponding to the second addend in (11)) and to the global best position (third addend in (11)). The position of each particle is then updated at each step by adding to the previous position the velocity obtained according to (11):

$$\underline{x}^{(i)}(t+1) = \underline{x}^{(i)}(t) + \underline{v}^{(i)}(t) \qquad i \in \{1, \ldots, S\}. \tag{12}$$

Possible stopping conditions for the PSO algorithm can be the achievement of a sufficiently low value of the fitness function 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.

In the simulation-based performance analysis in Section 4, the stopping condition will be the reach of 50 iterations. The population size S is set, in most of

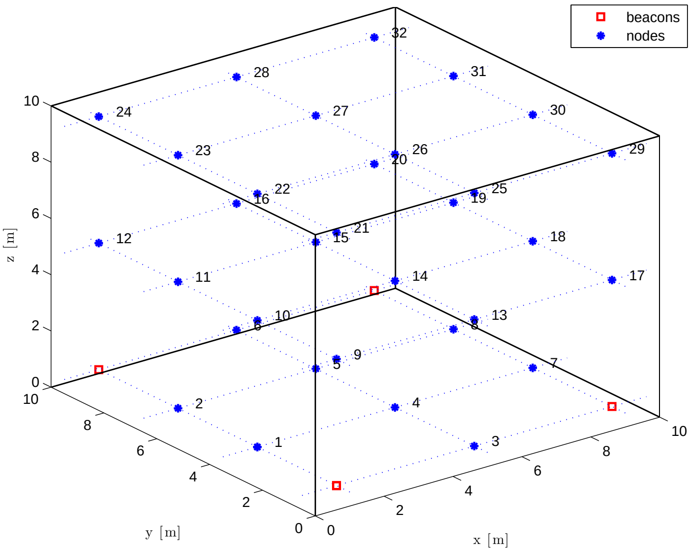

Fig. 1. The beacons (*red squares*) and the nodes with unknown positions (*blue dots*) are shown in a (three-dimensional) cubic region.

the presented results, to 200 and the position of each particle is randomly initialized in the three dimensional cube region, with side equal to 10 m where the nodes are placed—this is the natural choice under the assumption of unknown node's positions. Finally, the parameters c_1 and c_2 in (11) are both set to 2, so that the weights for social and cognition parts to are, on average, equal to 1 and the inertial factor is $\omega(t) = 0.5 - 0.01t$ (i.e., it starts from 0.5 when $t = 0$ and decreases to 0 in the last iteration, i.e., when $t = 50$).

4 Simulation-based Performance Analysis

In this section, we compare, through simulations, the performances of the two localization algorithms described in Section 3. Both TSML and PSO algorithms are implemented in a Matlab simulator. First, we consider the node configuration shown in Fig. 1, where the nodes are placed on regular spatial grid such that the coordinates of a generic point can be expressed as (x_k, y_k, z_k), $k \in \{1, \ldots, 36\}$ with: $x_k \in \{1\text{ m}, 5\text{ m}, 9\text{ m}\}$, $y_k \in \{0.5\text{ m}, 3.5\text{ m}, 6.5\text{ m}, 9.5\text{ m}\}$, and $z_k \in \{0.5\text{ m}, 5\text{ m}, 9.5\text{ m}\}$. In Fig. 1, the four vertices in the bottom plane are the positions where beacons are placed. Then, we consider the node configuration in Fig. 2, where the positions of the 36 nodes are obtained by "perturbating" the positions of the nodes in Fig. 1. More precisely, the coordinates of the nodes can be expressed as: (X_k, Y_k, Z_k), $k \in \{1, \ldots, 36\}$, where $X_k = x_k + \theta_k^{(x)}$,

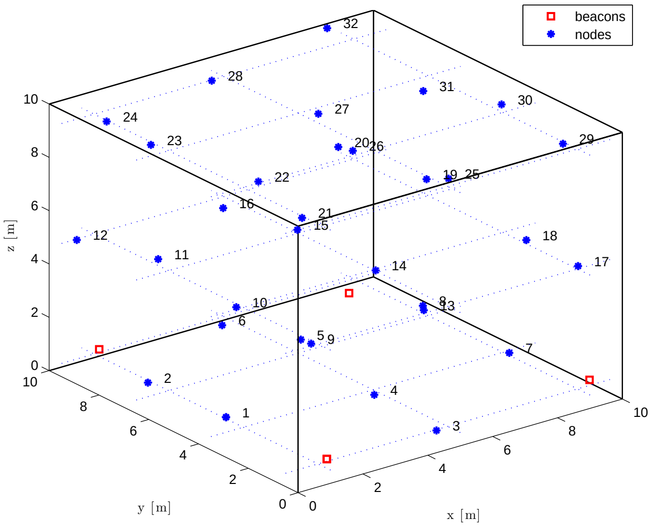

Fig. 2. The beacons (*red squares*) and the nodes with unknown positions (*blue dots*) are shown in a (three-dimensional) cubic region.

$Y_k = y_k + \theta_k^{(y)}$, and $Z_k = z_k + \theta_k^{(z)}$, with $\theta_k^{(x)}, \theta_k^{(y)}, \theta_k^{(z)} \sim \mathcal{N}(0 \text{ m}, 1 \text{ m})$ an independent of each other.

For each node whose position needs to be estimated, we assume to make 100 independent estimates of the distance between the node and each beacon, assuming that the measurement error follows the statistical characterization in (3). Finally, each distance $\hat{r}_i$, $i \in \{1, \ldots, M\}$, used in the considered localization algorithms, is obtained by averaging over the 100 estimates, i.e.,

$$\hat{r}_i = \frac{\sum_{j=1}^{100} \hat{r}_i^{(j)}}{100}$$

where $\hat{r}_i^{(j)}$ is the $j-$th estimate of the i-th distance. We remark that this approach is different from the one followed in [10], where the localization algorithms were applied to each set of range measurements $\{r_i\}_{i=1}^{M}$ and the final position estimates were obtained by averaging the 100 position estimates. In each scenario, we assume that the number of beacons is $M = 4$, i.e, the minimum number which allows applying the localization techniques described in Section 3.

In Fig. 3, we investigate the performance of the scenario shown in Fig. 1, i.e., four beacons (red squares) laying on the same plane are considered. In this case, the matrix $\underline{\underline{G}}$ involved in the TSML algorithm is not full rank, making this method inapplicable. At the opposite, the PSO algorithm guarantees accurate position estimates for all the nodes, as shown in Fig. 3.

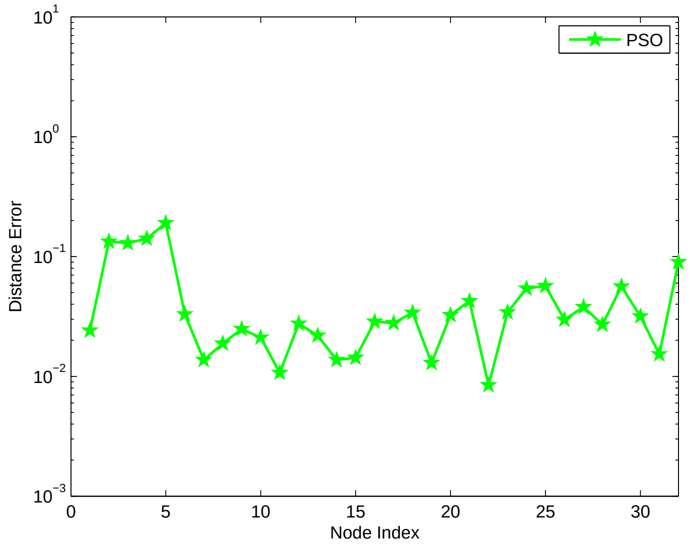

Fig. 3. The distance error, using the PSO algorithm, is shown as a function of the node index.

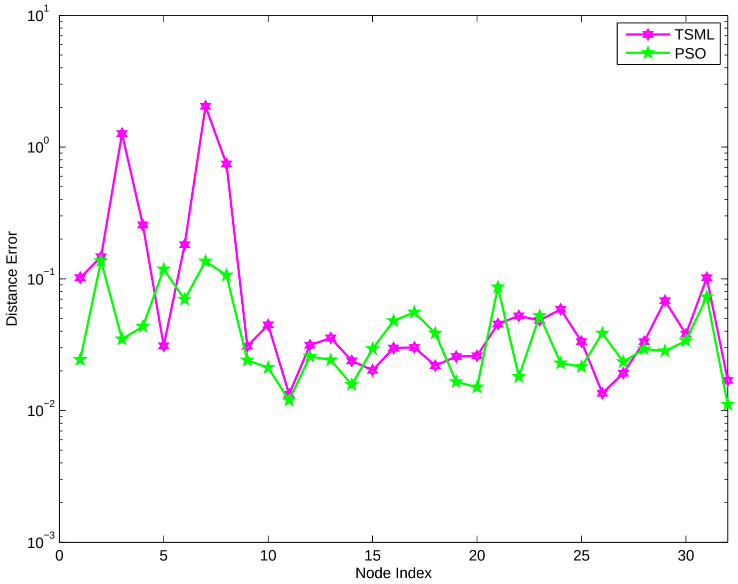

Fig. 4. The distance error on the position estimate of each node when using the TSML algorithm (*magenta hexagrams*) and the PSO algorithm (*green stars*) are shown.

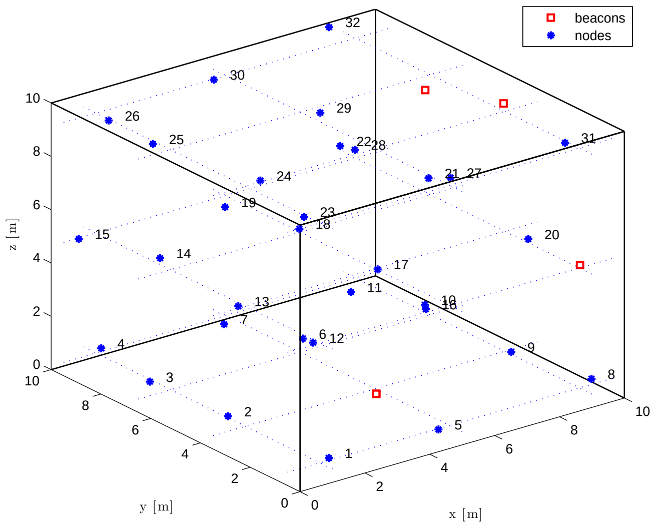

Fig. 5. The beacons (*red squares*) and the nodes with unknown positions (*blue dots*) are shown in a (three-dimensional) cubic region.

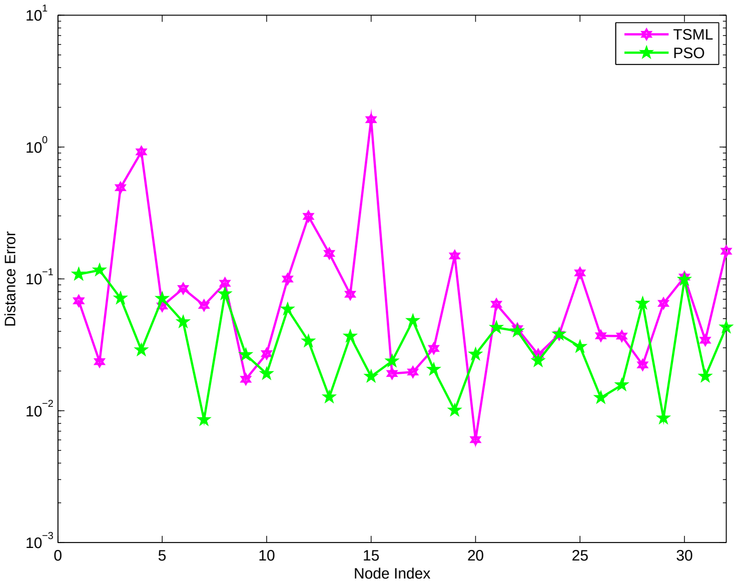

Fig. 6. The distance error on the position estimate of each node when using the TSML algorithm (*magenta hexagrams*) and the PSO algorithm (*green stars*) are shown.

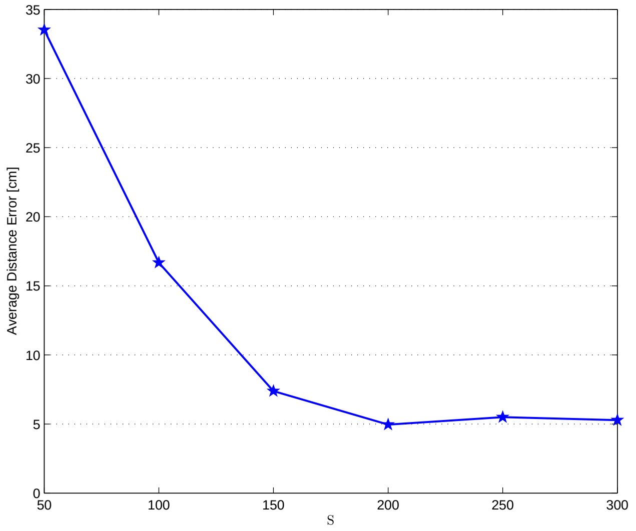

Fig. 7. Average distance error as a function of the swarm size S.

In Fig. 4, the performance in the scenario of Fig. 2 is investigated. Since, in this case, the nodes do not lie exactly on parallel planes, as in Fig. 1, even though the matrix $\underline{\underline{G}}$ of the TSML algorithm is full rank, it is ill-conditioned, leading to wrong position estimates of some nodes. In particular, the "critical" nodes are 3, 4, 7, 8, in correspondence to which the position estimation error is significantly higher than with the PSO algorithm. Overall, the PSO algorithm allows locating all the nodes with a satisfying accuracy.

In Fig. 5, we consider once again the scenario described in Fig. 2, but with a more general configuration of the beacons, which are now randomly placed. Once again, the corresponding performance results, shown in Fig. 6, show that the PSO algorithm leads to sufficiently accurate position estimates for all nodes, while the TSML algorithm leads to inaccurate estimates of the positions of some nodes. Therefore, it can be concluded that the performance of the PSO algorithm is better not only in particular cases, i.e., when the beacons (almost) lay on the same plane, but also with generic positions of the beacons.

Considering the same scenario of Fig. 5, we now investigate the impact of the swarm size S on the localization accuracy. More precisely, we are interested in evaluating the average distance errors, averaged over the 32 nodes with unknown positions, for different values of the swarm size S. Once again, the stopping condition for the PSO algorithm corresponds to reaching 50 iterations. In Fig. 7, the average distance error is shown as a function of the swarm size S. When considering values of the swarm size between 50 and 200, the average distance error is a decreasing function of S. This is an expected behaviour, since the higher is the swarm size, the faster is the convergence. At the opposite, when considering values of S higher than 200, the average distance error is almost constant, showing that increasing the population size above 200 individuals does not improve the localization performance. Observe that the fact that the average distance error does not attain zero is due to the fact that we are using the estimated distances (and not the true ones) as an input for the PSO algorithm. Finally, one can conclude that setting S equal to 200 is a good choice, as it is the lowest value of the population size which allows to obtain an accuracy around 5 cm—this, in particular, has motivated the choice of $S = 200$ in the previous performance results.

5 Conclusion

In this paper, we have proposed a swarm intelligent approach to UWB-signaling-based position estimation of nodes in a static three dimensional indoor scenario. Besides solving the non-linear system of the localization equations by means of the TSML algorithm, which is a "geometric" algorithm, the original problem has been re-written in terms of an optimization one, which is then solved by means of the PSO algorithm. Our results show that, while the accuracy of the TSML algorithm depends on the topology of the network and on relative positions of beacons and nodes, the approach based on the PSO algorithm allows to achieve a good accuracy in the position estimate, regardless of the configuration of nodes and beacons.

References

1. S. Monica and G. Ferrari, "Accurate indoor localization with UWB wireless sensor networks," *Proceedings of the* 23^{rd} *IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises* (WETICE 2014), track on Capacity-Driven Processes and Services for the Cyber-Physical Society (CPS), June 2014, Parma, Italy, pp. 287-289.
2. Wade H. Foy, "Position-location solutions by Taylor-series estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-12, no.2, March 1976, pp.187-194.
3. Christian Mensing and Simon Plass, "Positioning algorithms for cellular networks using TDOA," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP'06), vol. 4, Toulouse, France, May 2006, pp. 513-516.
4. S. Monica and G. Ferrari, "An experimental model for UWB distance measurements and its application to localization problems," *Proceedings of the IEEE International Conference on Ultra Wide Band* (ICUWB), September 2014, Paris, France, pp. 297-302.
5. Ralph O. Schmidt, "A new approach to geometry of range difference location," *IEEE Trans. Aerosp. Electron.*, vol. AES-8, no. 6, Nov. 1972, pp. 821-835.
6. S. Monica and G. Ferrari, "Optimized anchors placement: An analytical approach in UWB-based TDOA localization," *Proceedings of the* 9^{th} *International Wireless Communications & Mobile Computing Conference* (IWCMC), July 2013, Cagliari, Italy, pp. 982-987.

7. Y. Chan and K. C. Ho, “A simple and efficient estimator for hyperbolic location,” *IEEE Trans. Signal Process.*, vol. 42, no. 8, Aug. 1994, pp. 1905-1915.
8. G. Shen, R. Zetik, and R. S. Thomä, “Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment,” *Proceedings of the 5th workshop on positioning, navigation and communication 2008* (WPNC'08), Hannover, Germany, March 2008, pp. 71-78.
9. S. Monica and G. Ferrari, “Particle Swarm Optimization for Auto-localization of Nodes in Wireless Sensor Networks,” *Proc. 11th International Conference on Adaptive and Natural Computing Algorithms* (ICANNGA'13), April 2013, Lausanne, Switzerland, pp. 456-465.
10. S. Monica and G. Ferrari, “Impact of the Number of Beacons in PSO-based Auto-localization in UWB Networks,” *Proc. International Conference on the Applications of Evolutionary Computation* (EvoApplications13)*, Track on Nature-inspired Techniques for Communication Networks and other Parallel and Distributed Systems* (EvoCOMNET 2013), April 2013, Vienna, Austria, pp. 42-51.
11. S. Monica and G. Ferrari, “UWB-based localization in large indoor scenarios: Optimized placement of anchor nodes,” *IEEE Transactions on Aerospace and Electronic Systems*, to appear, 2015.
12. S. Monica and G. Ferrari, “Swarm intelligent approaches to auto-localization of nodes in static UWB networks,” *Applied Soft Computing*, vol. 25, December 2014, pp. 426-434, DOI:10.1016/j.asoc.2014.07.025.
13. E. Okamoto, M. Horiba, K. Nakashima, T. Shinohara, and K. Matsumura, “Particle swarm optimization-based low-complexity three-dimensional UWB localization scheme,” *Proceedings of the International Conference on Ubiquitous and Future Networks* (ICUFN), July 2014, pp. 120-124.
14. J. Zhang, P. V. Orlik, Z. Sahinoglu, A. F. Molisch, and P. Kinney, “UWB systems for wireless sensor networks,” *Proc. IEEE*, vol. 97, no. 2, Feb. 2009, pp. 313-331.
15. S. Busanelli and G. Ferrari, “Improved ultra wideband-based tracking of twin-receiver automated guided vehicles,” *Journal of Integrated Computer-Aided Engineering*, vol. 19, no. 1, 2012, pp. 3-22.
16. A.F. Molisch, D. Cassioli, Chia-Chin Chong, S. Emami, A. Fort, B. Kannan, J. Karedal, J. Kunisch, H.G. Schantz, K. Siwiak, and M.Z. Win, “A comprehensive standardized model for ultrawideband propagation channels,” *IEEE Trans. Antennas Propagat.*, vol. 54, no. 11, Nov. 2006, pp. 3151-3166.
17. D. Dardari, C. C. Chong, and M. Z. Win, “Threshold-based time-of-arrival estimators in UWB dense multipath channels,” *IEEE Trans. Commun.*, vol. 56, no.8, Aug. 2008, pp. 1366-1378.
18. J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proc. IEEE International Conf. on Neural Networks,* (IJCNN 2005), Nov. 1995, Perth, Australia, pp. 1942-1948.
19. R. Poli, J. Kennedy, T. Blackwell, “Particle swarm optimization, an overview,” *Swarm Intelligence Journal*, vol. 1, no.1, Aug. 2007, pp. 33-57.
20. Y. Shi and R. Eberhart, “A modied particle swarm optimizer,” *Proc. IEEE International Conference on Evolutionary Computation* (ICEC'98), May 1998, Anchorage, AK, USA, pp. 69-73.