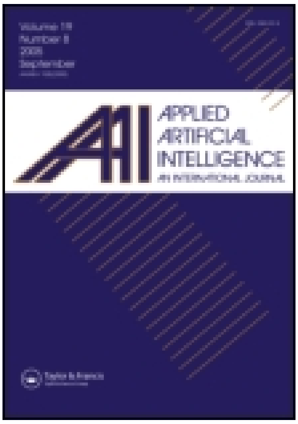


Applied Artificial Intelligence: An International Journal
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uaai20

INVESTIGATION ON THE EVOLUTION OF AN INDOOR ROBOTIC LOCALIZATION SYSTEM BASED ON WIRELESS NETWORKS

Gustavo Pessin [a c], Fernando S. Osório [a], Jefferson R. Souza [a], Jó Ueyama [a], Fausto G. Costa [a], Denis F. Wolf [a], Desislava Dimitrova [b], Torsten Braun [b] & Patrícia A. Vargas [c]

[a] Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP), São Carlos, São Paulo, Brazil

[b] Institute of Computer Science and Applied Mathematics, University of Bern, Bern, Switzerland

[c] School of Mathematical and Computer Sciences (MACS), Heriot-Watt University, Edinburgh, UK

To cite this article: Gustavo Pessin, Fernando S. Osório, Jefferson R. Souza, Jó Ueyama, Fausto G. Costa, Denis F. Wolf, Desislava Dimitrova, Torsten Braun & Patrícia A. Vargas (2013) INVESTIGATION ON THE EVOLUTION OF AN INDOOR ROBOTIC LOCALIZATION SYSTEM BASED ON WIRELESS NETWORKS, Applied Artificial Intelligence: An International Journal, 27:8, 743-758, DOI: 10.1080/08839514.2013.823328

To link to this article: http://dx.doi.org/10.1080/08839514.2013.823328

Applied Artificial Intelligence, 27:743–758, 2013

ISSN: 0883-9514 print/1087-6545 online
DOI: 10.1080/08839514.2013.823328

Taylor & Francis
Taylor & Francis Group

INVESTIGATION ON THE EVOLUTION OF AN INDOOR ROBOTIC LOCALIZATION SYSTEM BASED ON WIRELESS NETWORKS

Gustavo Pessin[1,3], Fernando S. Osório[1], Jefferson R. Souza[1], Jó Ueyama[1], Fausto G. Costa[1], Denis F. Wolf[1], Desislava Dimitrova[2], Torsten Braun[2], and Patrícia A. Vargas[3]

[1]*Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP), São Carlos, São Paulo, Brazil*
[2]*Institute of Computer Science and Applied Mathematics, University of Bern, Bern, Switzerland*
[3]*School of Mathematical and Computer Sciences (MACS), Heriot-Watt University, Edinburgh, UK*

□ *This work addresses the evolution of an artificial neural network (ANN) to assist in the problem of indoor robotic localization. We investigate the design and building of an autonomous localization system based on information gathered from wireless networks (WN). The article focuses on the evolved ANN, which provides the position of a robot in a space, as in a Cartesian coordinate system, corroborating with the evolutionary robotic research area and showing its practical viability. The proposed system was tested in several experiments, evaluating not only the impact of different evolutionary computation parameters but also the role of the transfer functions on the evolution of the ANN. Results show that slight variations in the parameters lead to significant differences on the evolution process and, therefore, in the accuracy of the robot position.*

INTRODUCTION

Mobile robot navigation is one of the most fundamental and challenging directions in the mobile robotics research field, and it has received great attention in recent years (Fu, Hou, and Yang 2009). Intelligent navigation often depends on mapping schemes, which depend on the localization

The authors would like to acknowledge the financial support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), processes ID 573963/2008-8 and 08/57870-9. The first author would like to acknowledge the Capes Foundation, Ministry of Education of Brazil, process BEX 4202-11-2 and the fourth author would like to thank FAPESP (process ID 2008/05346-4), CNPq (process ID 474803/2009-0) and RNP for funding the bulk of his research.

Address correspondence to Gustavo Pessin, Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP), Po Box 668, São Carlos, SP 13560-970, Brazil. E-mail: pessin@icmc.usp.br

scheme. For either indoor or outdoor environments, the mapping and localization schemes have their own features. Thereby, localization is a key problem in mobile robotics and it plays a pivotal role in various successful mobile robot systems (Thrun et al. 2001).

This article describes an investigation on the evolution of a system for indoor localization, which estimates the position of one robot based on information gathered from wireless networks (WNs). The signal strength of several wireless nodes are used as input in the ANN in order to determine the position of one robot in an indoor space. The evolution of the ANN, detailed in the section "Evolutionary Localization" is done using particle swarm optimization (PSO) (Eberhart, Kennedy, and Shi 2001; Engelbrecht 2005). We show the complete hardware and software architecture for the robotic system developed so far. Our main focus in this work is to report findings that corroborate the use of evolutionary computation techniques to create autonomous intelligent robots (Fogel 2006). This article is an extended version of our previous work (Pessin et al. 2012). It aims to describe in a more thorough way the applied methodology, the experiments, and the discussion about the results. The main novel aspect in this article is the investigation on several new parameters of the PSO.

In indoor spaces, sensors such as laser range finders and cameras might be used for pose estimation (Napier, Sibley, and Newman 2010), but they require landmarks (or maps) in the environment and a fair amount of computation to process complex algorithms. These sensors also have a limited field of vision, which makes the localization task more difficult. In the case of video cameras, the variation of light is also a serious issue. Another commonly used sensor is the encoder, which provides odometry. Odometry is a useful source of information in some cases (Martinelli 2002) but it has an incremental error that usually invalidates its use in real systems.

WNs are widely available in indoor environments and might allow efficient global localization while requiring relatively low computing resources. Other advantages of this technology are that it may provide high degrees of scalability and robustness. However, the inherent instability of the wireless signal does not allow it to be used directly for accurate position estimation. One machine learning technique that could reduce the instability of the signals of the WN is the artificial neural Network (ANN) because of its capacity to learn from examples, as well as the generalization and adaptation of the outputs (Mitchell 1997).

In Elnahrawy, Li, and Martin (2004), it has been shown that obtaining an absolute performance in localization, by means of a WN, depends on the environmental configuration. This means that different approaches are required for different environments, such as using different kinds of signals and filters. Evaluations in large indoor areas (such as a building) present specific difficulties not always the same as in small indoor areas (such as a room). These difficulties are related to the problem of attenuation and reflection of the signals on the walls and the different sources of interferences. The use of WNs addressing localization inside a building can be seen in Espinace, Sota, and Torres-Torriti (2008) and Ladd and colleagues (2004). Another approach for localization uses the wireless sensor network (WSN) in which a large number of small sensors are used to pick up information from the environment. The information acquired by the sensors can be regarded as a fingerprint (Robles, Deicke, and Lehnert 2010).

This article has the following structure: "Methodology" outlines the methodology that is employed to set up and to evaluate the experiments. Experimental Results and Discussion describes all the evaluations and discussions that have been carried out. The final section makes some concluding comments and examines the future prospects of this area of research.

METHODOLOGY

The indoor localization system uses an evolved ANN.[1] The inputs of the ANN are signals strength measurements from WNs (802.11b/g) received by the robot[2] from eight statically positioned access points (AP) as shown in Figure 1(a) and 1(b). The signal obtained from the WN is the received signal strength indication (RSSI). This value is obtained with the aid of the GNU/Linux command iwlist.[3] Because we have used the iwlist command, there was no need to establish a connection (or login) with different specific networks. The scan of the networks, without a connection, provides

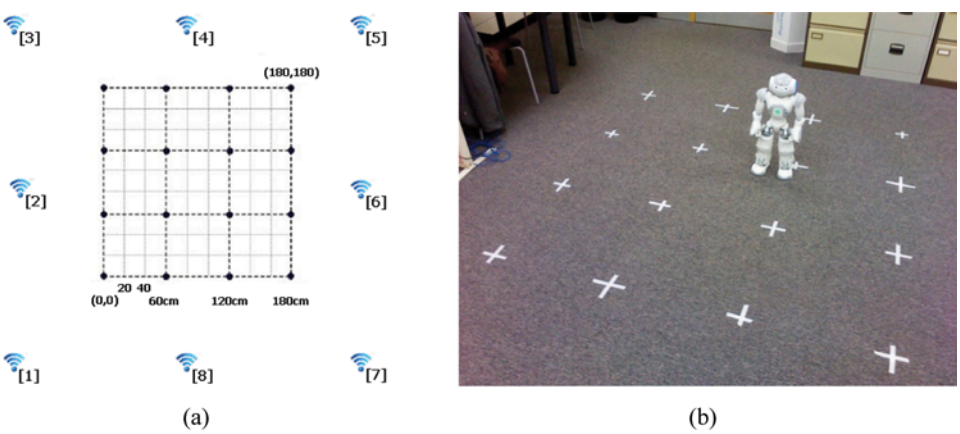


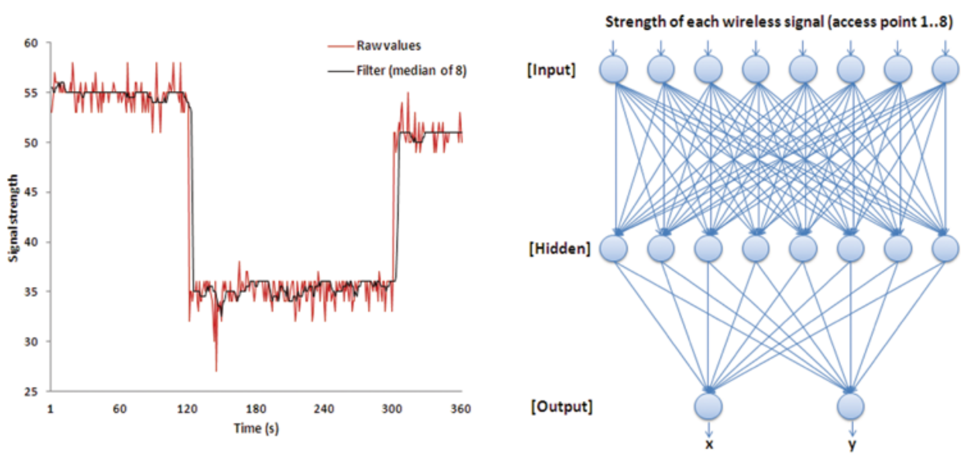
FIGURE 1 (a) Graphical representation of the working area. It represents an area of 180 cm by 180 cm. (b) Picture of the working area with the robot, similar to what is represented in Figure 1(a). Each small cross is placed 60 cm apart (color figure available online).

enough information for this evaluation. Without a connection, the system becomes easier to use, more lightweight and flexible. Furthermore, as the Nao robot has an operating system based on GNU/Linux, this approach may be generalized to any other GNU/Linux based system.

The evolution of the ANN is carried out using data collected by the robot. We use the robot within the working area (Figure 1(b)) and collected readings over 3 minutes (i.e., ≈180 readings) at each marked point. With a displacement of 60 cm, mapping out a plane of 180 cm by 180 cm, it means 16 points to read resulting in ≈2880 readings altogether.

Our approach relies on the ANN learning and generalization capabilities in an attempt to reduce the effect of unstable data (due to signal strength oscillation), and to increase the accuracy of the position estimation of the robot. However, as the values obtained from the reading of the APs are quite unstable, we improve the learning capability using the noise filter previously evaluated in Pessin and colleagues (2011). The behavior of the noise filter can be seen in Figure 2(a), where two lines represent scanning of one AP over a period of time. The red line shows the raw value and the black line shows how the median filter removes some of the noise. Although it generates a delay of ∼8 seconds in acquiring the new position, it was shown in Pessin and coauthors (2011) that the accuracy turns out to be widely better.

The number of neurons in the input layer is equivalent to the number of available APs—because we use 8 APs, the inputs of the ANN use one neuron for each network signal. The order is important, and hence, AP 1 was linked to neuron 1, AP 2 with neuron 2, and so on. The outputs of



(a) (b)

FIGURE 2 (a) Sample of filter behavior. The red line shows the raw value from one of the access points. The black line shows how the median filter removes some of the noise. (b) Example of ANN topology (color figure available online).

the network are two values, in other words, the coordinates (x, y). We measure the output errors by using the Euclidean distance, as shown in Equation 1. The value d is the error (distance, in centimeters), (x_1, y_1) are the expected values from the ANN validation set, and (x_2, y_2) are the obtained value while using the ANN.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

Evolutionary Localization

As we seek to evaluate ANN characteristics and also the evolutionary process, we started with a simple ANN topology as can be seen in Figure 2(b). We evaluate changes in the ANN topology and in the role of the transfer function. Furthermore, because the evolution is carried out using PSO (Eberhart, Kennedy, and Shi 2001; Engelbrecht 2005), we evaluate several aspects that influence the search efficiency in the PSO. These aspects are related to confidence models, neighborhood topology, and inertia, among others.

We use PSO to evolve two different structures. In the first, we use the PSO to evolve just the ANN weights. In the second, we evolve the ANN weights plus the slope of the transfer function. As an example, the ANN in Figure 2(b) has 80 connections plus 10 weights for bias, hence, the PSO particle has 90 positions (i.e., it is a vector with 90 positions). For the slope, we consider its use just in the hidden layer, and it is the same value for all neurons. Hence, it adds only one more value to the PSO particle.

The PSO is a stochastic optimization technique, inspired by the social behavior of birds flocking and fish schooling (Eberhart, Kennedy, and Shi 2001; Engelbrecht 2005). The optimization process occurs in two different ways simultaneously: through cooperation (group learning) and competition (individual learning) among particles (individuals) from a swarm (population). It shares many concepts with evolutionary computation techniques such as genetic algorithms (GA), in which there is an initial population (where each individual represents a possible solution) and a fitness function (whose value represents how far an individual is to an expected solution). However, unlike GA, PSO has no explicit concepts of evolution operators such as crossover or mutation. In the PSO, there is a swarm of randomly created particles. At each algorithm iteration, each particle is updated as follows: (1) the best population fitness; (2) the best fitness found by the particle (considering past generations of the particle). Each particle has a position $\boldsymbol{x}$ (or a position vector) and a velocity $\boldsymbol{v}$ (or velocity vector). The position represents a solution for the problem, and the velocity defines the particle's displacement direction weight.

The new particle's position is given by Equation (2). Where x_k^i is the position of particle i at instant k, and v_k^i is the particle's i velocity at k moments. The particle's velocity is updated according to Equation (3).

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{x}_k^i + \boldsymbol{v}_{k+1}^i \tag{2}$$

$$\boldsymbol{v}_{k+1}^i = w \cdot \boldsymbol{v}_k^i + c_1 \cdot r_1(\text{pbest} - \boldsymbol{x}_k^i) + c_2 \cdot r_2(\text{gbest} - \boldsymbol{x}_k^i) \tag{3}$$

In Equation (3), v_k^i is the particle actual velocity, w represents a particle inertia parameter, pbest is the best position among all positions found by the individual (particle best), gbest is the best position among all positions found by the group (group best), c_1 and c_2 are trust parameters, r_1 and r_2 are random numbers between 0 and 1. Parameters (w, c_1, c_2, r_1 e r_2) are detailed following.

The velocity is the optimization's process guide parameter (Engelbrecht 2005) and reflects both the particle's individual knowledge and group knowledge. Individual knowledge is known as the *cognitive component* whereas group knowledge is known as the *social component*. Velocity consists of a three-term sum: (1) previous speed, utilized as a displacement direction memory and can be seen as a parameter that avoids drastic direction changes; (2) cognitive component, directs the individual to the best position found so far (i.e., memory of the particle); (3) social component, directs the individual to the best particle in the group.

Parameters c_1 and c_2 (confidence or trust) are used to define individual or social tendency importance. Default PSO works with static and equal trust values ($c_1 = c_2$), which means that the group experience and the individual experience are equally important (called *Full Model*). When parameter c_1 is zero and parameter c_2 is higher than zero, PSO uses only group information (called *Social Model*). When parameter c_2 is zero and parameter c_1 is higher than zero, PSO uses only particle's information, disregarding the group experience (called *Cognitive Model*). Random value introduction (r_1 and r_2) on velocity adjustment allows PSO to explore the search space in a better way (Engelbrecht 2005). The inertia parameter aims to balance local or global search. As the value approximates to 1.0, search gets close to global while lower values allow local search. Usually this value is between 0.4 and 0.9. Some authors suggest its linear decay, but they warn that it is not always the best solution. Most parameters are problem dependent (Engelbrecht 2005; Yao 1999).

EXPERIMENTAL RESULTS AND DISCUSSION

This section describes the six evaluations we have performed, considering several changes in the ANN and PSO techniques. Results are always presented from the validation dataset (optimum generalization point).

TABLE 1 Evaluation Set Used to Investigate the Swarm Size and the Number of Generations in the PSO

	Swarm size		
Generations	200	500	1,000
500	E1	E2	E3
1,000	E4	E5	E6
2,000	E7	E8	E9

We performed 25 runs for each parameter set, considering different random seeds on each initialization. The evolutionary process considers 2/3 of the dataset as training data and 1/3 as validation data.

Range of Velocity and Position

Velocity is the optimization's process guide parameter (Engelbrecht 2005) and reflects both the particle's individual knowledge and group knowledge. In mathematical terms, it specifies the weight of the displacement vector. Higher velocity leads to larger displacement in the position and it is an issue related to global and local search (Pant, Thangaraj, and Abraham 2009; Herrera and Zhang 2009). PSO's position, in this work, is regarded as the synaptic weight of the connections.

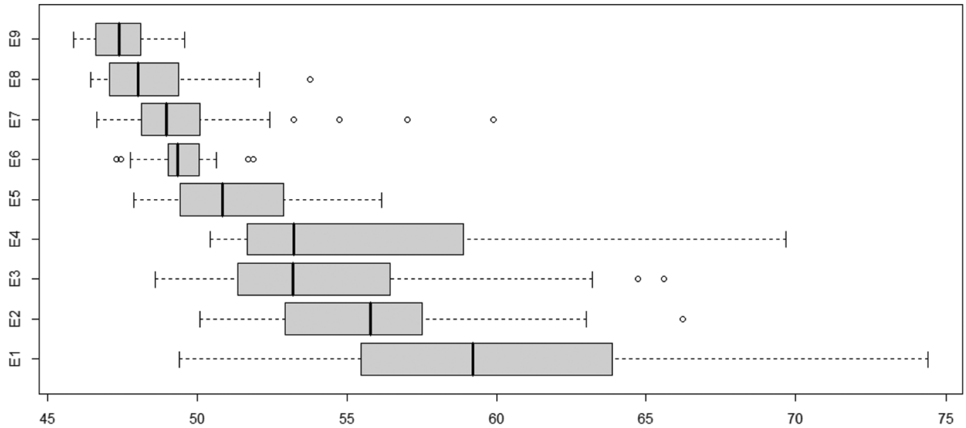
We evaluate the impact of using different range values in the initialization of PSO's velocity and position. Furthermore, we also evaluate how the number of generations and the swarm size impact the evolution. Table 1 shows the evaluation set related to swarm size and the number of generations. Table 2 shows the evaluation set for weights of connections (i.e., position) and velocity parameters in the PSO. Results can be seen in Figure 3.

We can see in Figure 3(a) that more generations and larger swarm size provides better results (E9). However, even using a swarm size equal to 1,000 and number of generations equal to 2,000, the evolutionary process was not reaching learning stabilization (the learning curve still had slight improvements).

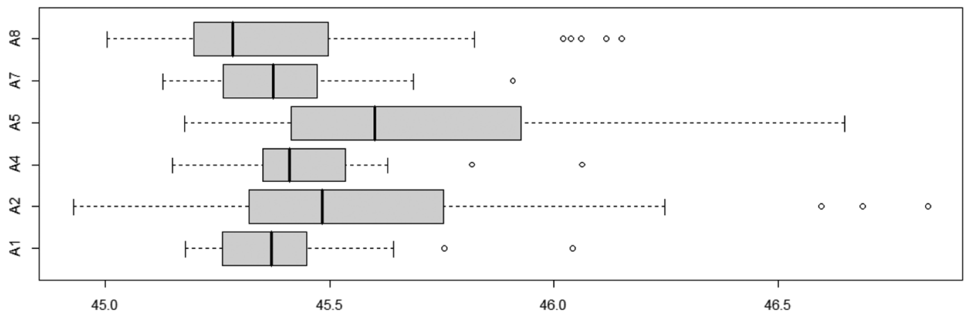
In Figure 3(b) we can see that the two sets that obtained the best (lower error) results considering the range of velocity and position are A8 and A2.

TABLE 2 Evaluation Set Used to Investigate the Weights of Connections (i.e., Position) and Velocity Parameters in the PSO

	Position		
Velocity	{−2.0;2.0}	{−5.0;5.0}	{−20.0;20.0}
{−2.0;2.0}	A1	A2	A3
{−5.0;5.0}	A4	A5	A6
{−20.0;20.0}	A7	A8	A9



(a)



(b)

FIGURE 3 (a) Results using different swarm sizes and numbers of generations (an shown in Table 1). We can see that more generations and larger swarm size provides better results (E9). (b) Results using different initialization in PSO velocity and position (an shown in Table 2). Best (lower error) can be seen in A8 and A2. The graphs are in different scales. The *x* axes represent the error in centimeters. Sets A3, A6, and A9 are not presented in the figure because of average errors larger than 55 cm.

Both use positions between $\{-5.0; 5.0\}$ but have different velocity ranges. We can see that although A2 has the lowest minimum error, it has the widest dispersion among all. The A8 set has the lowest median and a minimum error close to A2. Although, considering the graph scale (2 cm), all results are not much different. Thereafter, for the next steps, we maintain the set A8 in the PSO. Moreover, given that the evolutionary process was not reaching complete stabilization, we extrapolate E9 with 10,000 generations instead of 2,000 generations.

Confidence Models, Inertia, and the Role of the Transfer Function

The confidence model is an issue directly related to how one particle is influenced by another particle of the swarm or by itself. It is very closely

TABLE 3 Evaluation Set Used to Investigate the Inertia and Confidence Models (in the PSO) and the Transfer Function (in the ANN)

	Full		Social		Cognitive	
Inertia	Linear	Logistic	Linear	Logistic	Linear	Logistic
0.3	Fi3	Fo3	Si3	So3	Ci3	Co3
0.5	Fi5	Fo5	Si5	So5	Ci5	Co5
0.7	Fi7	Fo7	Si7	So7	Ci7	Co7

related to the diversity of the population. Greater diversity may lead to better global search, whereas small diversity may lead to premature convergence (Ozcan and Mohan 1999; Kennedy 1997).

The inertia, as presented in the section "Evolutionary Localization" aims to balance local or global search. As the value approximates to 1.0, search gets close to global while lower values allow local search. The inertia weight (w) controls the momentum of the particle by weighting the previous velocity (Malik et al. 2007).

Transfer functions are related to the production of a scalar neuron output (Dorofki et al. 2012; Haykin 2008). Usually, a transfer function (or activation function) is used to limit or smooth the output of a neuron. It is related to the learning and generalization capabilities of the ANN.

We seek to understand the behavior of the PSO evolving the ANN considering confidence models, the inertia, and the role of the transfer function. Table 3 shows the evaluated parameter set. Linear and logistic are the two types of transfer function used in the ANN hidden layer.

We can see that results (Figure 4) using the Cognitive Model or the Social Model were worse than using the Full Model. The four best sets

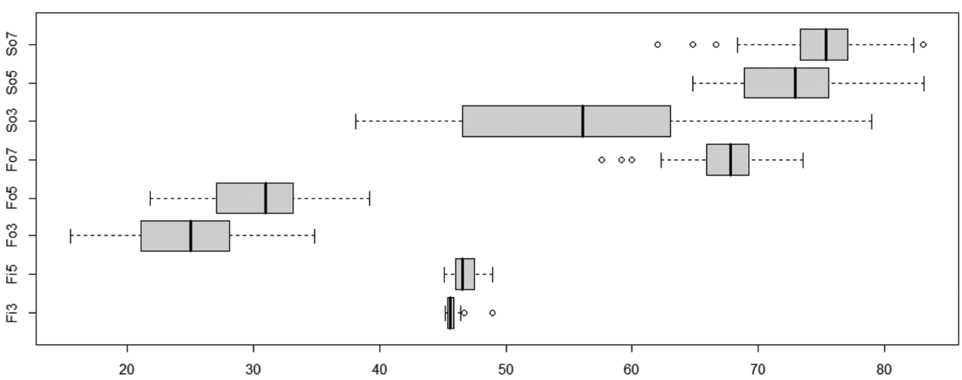


FIGURE 4 Results using different PSO confidence models and inertia, as shown in Table 3. We show the sets where the average error was lower than 80 cm – 8 of 18 sets. The x axis represents the error in centimeters. The best set (lowest median) is Fo3 (PSO full model. Transfer function logistic and inertia equal to 0.3).

are {Fi3, Fi5, Fo3, Fo5}. We can see that the sets with the logistic transfer function obtained best results near to 15 cm, whereas the sets with linear transfer function obtained best results near to 45 cm. In these sets, we can also see that when inertia was used as 0.3, we have better results than using inertia equal to 0.5 or 0.7. It makes sense because the lower the inertia, the better the local search (fine tuning). Hence, the best parameter set that we maintain in order to go to the next steps is Fo3.

We have used the logistic transfer function with a slope of 0.02. We also had a different PSO evolving the slope, however, the results when we leave the PSO to evolve the slope were similar to the use of the predefined value. Of course, the finding of slope equal to 0.02 was not trivial because it was encountered by analyzing the output of the sum of the hidden layers. Such a situation may encourage the use of the PSO in order to find the slope of the transfer function. Some evaluations taking into account nonlinear inertia will be show in the Section titled "Nonlinear Inertia."

Number of Neurons in the Hidden Layer

The number of neurons in the ANN hidden layer is related to the learning and generalization capabilities. Too few neurons can lead to underfitting (lack of learning) whereas too many neurons can lead to overfitting (learning too well the training data and lacking generalization capabilities) and waste of computational resources (Teoh, Tan, and Xiang 2006; Huang 2003).

We evaluate different numbers of neurons in the hidden layer to understand the learning and generalization capabilities. Hence, we evaluate the use of 2, 4, 8, 12, and 16 neurons. Results can be seen in Figure 5. We can see poorest results using 2 and 4 neurons. Using 8 and 12 neurons leads to good minimum errors but high dispersion. The ANN with 16 neurons presents the lowest error and the more homogeneous results. Statistical analyses (Welch Two Sample t-test[4]) on N12 and N16 showed p-value of

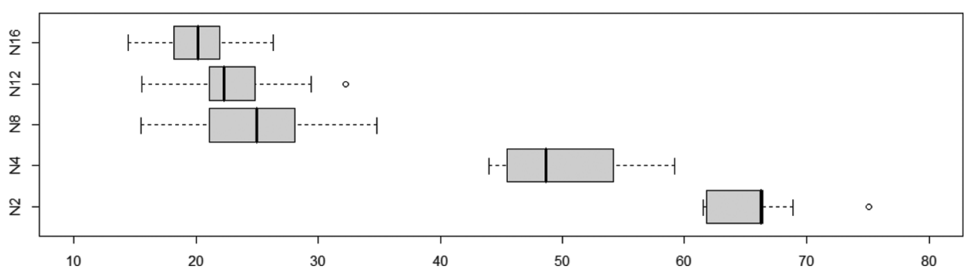


FIGURE 5 Results using a different number of neurons in the ANN hidden layer. The x axis represents the error in centimeters. We can see poorest results using 2 and 4 neurons. Using 8 and 12 neurons shows good minimum errors but high dispersion. The ANN with 16 neurons presents the lowest error and the more homogeneous results, being ≈12% better than N12.

0.011, i.e. with 95% of confidence, the hypothesis that the two population means are equal is rejected, in other words, the use of 16 neurons does improve the system accuracy in ≈12%.

Neighborhood Topologies

Neighborhood topologies are related to the choice of the best particle to follow. In the star model, all particles follow the best among all (example in Figure 6(a)). Nevertheless, it may sometimes be more susceptible to local minima. Other neighborhood topologies may be less susceptible to local minima, in which the particle does not follow the global best but the best of some subpopulation (example in Figure 6(b)). In general, it could increase the diversity. We have evaluated seven different types of neighborhood topologies:

- The star model (ST). As shown in Figure 6(a), all the particles (light gray) follow the best particle among all (light orange).
- Two subpopulations – i.e., 2 groups of neighbors having two gbests, one for each subpopulation (S2). As shown in Figure 6(b), the particles (light gray) follow the best particle of its neighborhood (light orange).
- Four subpopulations – i.e., four groups of neighbors having four gbests, one for each subpopulation (S4).
- Two subpopulations having one gbest, which is the average of the gbest of each subpopulation (AV2). As shown in Figure 6(c), the particles (light gray) follow the average of the best particle of each neighborhood (light green).
- Four subpopulations having one gbest, which is the average of the gbest of each subpopulation (AV4).

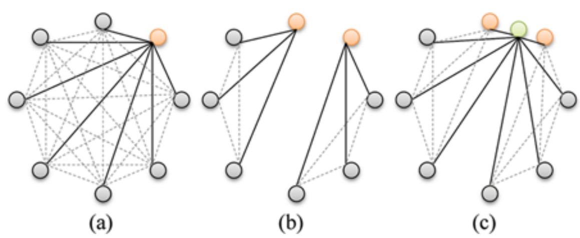


FIGURE 6 Examples of neighborhood topologies. (a) Star model – every particle may see any other particles (dashed line), although they follow (straight line) just the global best (light orange). (b) Two subpopulations – i.e., 2 groups of neighbors having two gbests, one for each subpopulation. Every particle may see any other particles (dashed line) in their subpopulation, although they follow (straight line) the global best (light orange) of each subpopulation. (c) Altered two subpopulations—there are two subpopulations and the gbest is the average of the gbest of each subpopulation. The particles (light gray) follow (straight line) the average of the best particle of each neighborhood (light green) (color figure available online).

- Two subpopulations in which each position of a particle follows one or another position of the gbest randomly (RPR).
- Two subpopulations in which the particle follows one or another gbest randomly – i.e., following all the particles (RPO).

We can see in Figure 7 the results of using different types of neighborhood topologies. We can see the poorest results in RPO, RPR, and AV4. The sets ST, S2, S4, and AV2 were evaluated statistically and showed a *p*-value from the Kruskal-Wallis rank sum test[5] equal to 0.355 – i.e., the hypothesis that the sets come from the same distribution is not rejected using 95% of confidence; hence we consider it as similar results. Because the statistical test leads us to consider it as similar results, we believe is better to use the least complex option, which is the Star Model (ST).

Nonuniform Inertia

In previous sections we have presented some explanations about the inertia influence. We have shown that maintaining the inertia uniform, equal to 0.3 have had better results than inertia equal to 0.5 and 0.7.

Works (Silveira et al. 2010; Deep, Arya, and Bansal 2011; Kentzoglanakis and Poole 2009) have evaluated nonuniform inertia showing that sometimes it may improve the PSO. Hence, this section aims to evaluate the impact of using nonuniform inertia in the localization problem. We have evaluated four different types of inertia behavior: one uniform (Figure 8(a)) and three nonuniforms, with linear decay (Figure 8(b)), exponential decay (Figure 8(c)) and sinusoid behavior (Figure 8(d)).

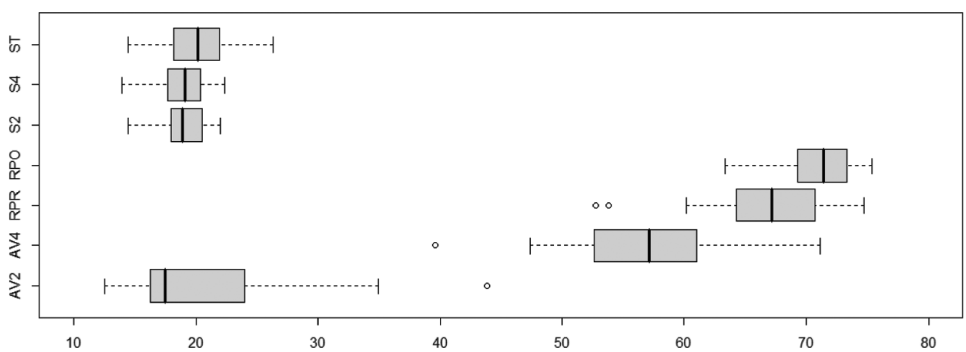


FIGURE 7 Results using different types of neighborhood topologies. We can see the poorest results in RPO, RPR, and AV4. The sets ST, S2, S4, and AV2 were evaluated statistically and showed a *p*-value from the Kruskal-Wallis rank sum test equal to 0.355—i.e., these four sets are considered as equivalent, using 95% of confidence.

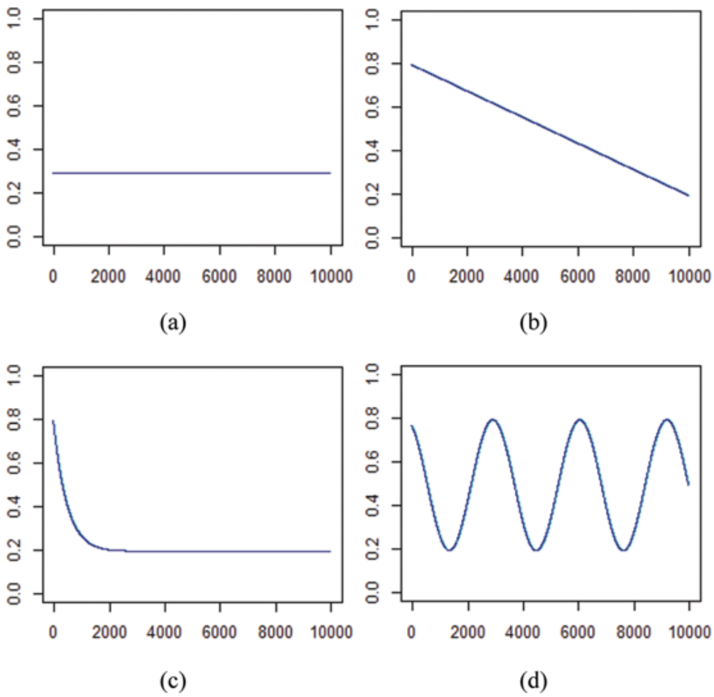


FIGURE 8 Example of inertia behavior: (a) uniform ($w = 0.3$), (b) linear decay, (c) Exponential, (d) Sinusoid. Values are normalized between 0.8 and 0.2 (color figure available online).

Figure 9 shows the result of the evolution considering the four evaluated types of inertia behavior. We can see that the use of uniform inertia has presented better results than nonuniform inertia. The result of the statistical test (Welch Two-Sample t-test) showed p-values lower than 0.000 for the sets $\{(\mathrm{ST}, \mathrm{EX}), (\mathrm{ST}, \mathrm{LN}), (\mathrm{ST}, \mathrm{SN})\}$; in other words, the hypothesis that the population means are equal is rejected, using 95% of confidence. Hence, for this problem, we can see that the use of uniform inertia showed significant better results than other types of inertia.

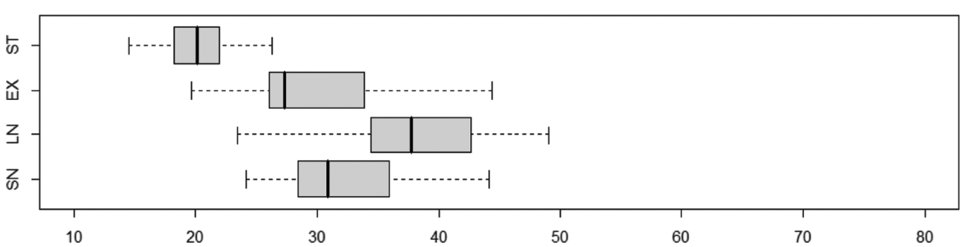


FIGURE 9 Result of the evolution considering the four evaluated types of inertia behavior. We can see that the use of uniform inertia has presented better results than nonuniform inertia.

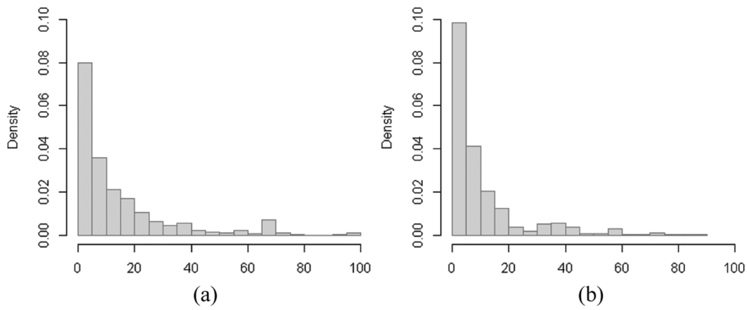


FIGURE 10 Histogram of the localization error using the best acquired ANN. (a) Using 100 k generations in the PSO. (b) Using 100k generations in the PSO. The *x* axes represent the error in centimeters.

More Generations

As a last step, we performed a new evaluation considering the best set found so far but running the evolutionary process for 100 k generations instead of 10 k. Results can be seen in Figure 10. The data shown in Figure 10(a) and 10(b) presents an average error and standard deviation, respectively, of (14.6, 18.4) and (10.2, 14.4); in other words, running the PSO for 100 k generations allowed us to decrease the average error by more ≈30%. We can see that Figure 10(a) has less results in the first class (0 to 5 cm) and a bigger tail than Figure 10(b). Statistical evaluation using the Mann-Whitney test (because it can be considered as a non-normal distribution) showed *p*-value equal to $1.517e^{-11}$ (the hypothesis that the two populations are the same is rejected using 95% of confidence). Figure 11 shows a section of the plane (Figure 1(a)) with expected and obtained values for four positions, using the best acquired ANN. For all positions in the evaluated plane, 86% of the errors are below 20 cm.

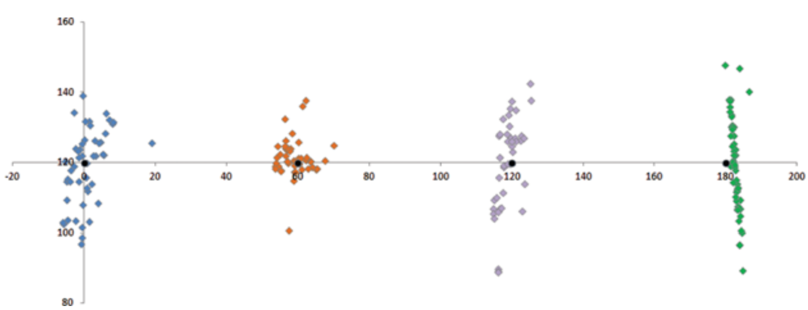


FIGURE 11 Section of the plane (Figure 1(a)) with expected and obtained values for four coordinates using the best evolved ANN. The black dots are the expected value; diamonds show the obtained values. Each color represents a different position. Axes *x* and *y* are in centimeters. For all positions in the full evaluated plane, 86% of the errors are below 20 cm (color figure available online).

CONCLUSION AND FUTURE WORK

In this article we have described an investigation addressing the evolution and the use of an ANN to assist in the problem of indoor localization by using data gathered from WNs. With the data obtained from the WN, we employed a median noise filter as shown in Pessin and collegues (2011). We evaluated several PSO and ANN settings. Results showed that the use of the transfer function in the ANN together with the PSO Full Model allowed us to decrease the average error from ≈45 cm to ≈15 cm. Also, we might see that the use of PSO neighborhood topology and the nonuniform inertia did not allow any significant improvement. Finally, the system could be improved using a larger number of neurons in the hidden layer and a larger number of generations, leading to an average error of ≈10 cm.

Nevertheless, it is important to notice that results presented in this paper cannot be directly compared with results from Pessin and coauthors (2011) because the studies have not employed the same data; they were collected in different environments and with different robots. Future work may include an investigation to improve the local search, because, even using a significantly vast number of generations, we still have a slightly decreasing error. Further, we are considering the other two approaches, that is, the comparison with other evolutionary techniques and a comparison with classical ANN learning algorithms to verify accuracy and efficiency.

NOTES

1. Source code and data files used to evolve the ANN are available in goo.gl/vfXN2.
2. Although we have used the humanoid robot NAO, the proposed methodology may be applied to any kind of device with wireless capabilities.
3. Used as iwlist < interface > scanning.
4. Sets N12 and N16 can be considered as normal distributions. We have used the Shapiro-Wilk normality test, which showed p-values equal to 0.634 and 0.928.
5. We have used Kruskal–Wallis rank sum test because the hypothesis of AV2 being a normal distribution was rejected by the Shapiro-Wilk normality test.

REFERENCES

Deep, K., M. Arya, and J. C. Bansal. 2011. A non-deterministic adaptive inertia weight in pso. In *Proceedings of the 13th annual conference on genetic and evolutionary computation, GECCO '11*, 1155–1162. New York, NY, USA: ACM.

Dorofki, M., A. H. Elshafie, O. Jaafar, O. A. Karim, and S. Mastura. 2012. Comparison of artificial neural network transfer functions abilities to simulate extreme runoff data. In *International conference on environment, energy and biotechnology*, IACSIT Press, Singapore, April 2012.

Eberhart, R. C., J. Kennedy, and Y. Shi. 2001. *Swarm intelligence*. San Francisco, CA: M. Kaufmann.

Elnahrawy, E., X. Li, and R. Martin. 2004. The limits of localization using signal strength: A comparative study. In *IEEE SECON, Sensor and Ad Hoc Communications and Networks, First Annual IEEE Communications Society Conference*, Santa Clara, CA, October 4–7, 2004, 406–414.

Engelbrecht, A. P. 2005. *Fundamentals of computational swarm intelligence*. Hoboken, NJ: Wiley.

Espinace, P., A. Soto, and M. Torres-Torriti. 2008. Real-time robot localization in indoor environments using structural information. In *Robotic Symposium, 2008. LARS '08. IEEE Latin American*, 79–84. Los Alamitos, CA, USA: IEEE Computer Society.

Fogel, D. 2006. Evolutionary computation: Toward a new philosophy of machine intelligence. *IEEE Press Series on Computational Intelligence*. Piscataway, NJ: Wiley.

Fu, S., Z. Hou, and G. Yang. 2009. An indoor navigation system for autonomous mobile robot using wsn. In *International Conference on Networking, Sensing and Control, ICNSC '09*, Okayama, Japan, March 26–29, 2009, 227–232.

Haykin, S. O. 2008. *Neural networks and learning machines* 3rd ed. Upper Saddle River, NJ: Prentice Hall.

Herrera, F., and J. Zhang. 2009. Optimal control of batch processes using particle swarm optimisation with stacked neural network models. *Computers & Chemical Engineering* 33 (10): 1593–1601.

Huang, G.-B. 2003. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks* 14 (2): 274–281.

Kennedy, J. 1997. The particle swarm: Social adaptation of knowledge. In *IEEE International Conference on Evolutionary Computation*, Indianapolis, IN, April 13–16, 1997, 303–308.

Kentzoglanakis, K., and M. Poole. 2009. Particle swarm optimization with an oscillating inertia weight. In *Proceedings of the 11th annual conference on genetic and evolutionary computation. GECCO '09*, 1749–1750. New York, NY, USA: ACM.

Ladd, A., K. Bekris, A. Rudys, D. Wallach, and L. Kavraki. 2004. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation* 20 (3): 555–559.

Malik, R. F., T. A. Rahman, S. Z. M. Hashim, and R. Ngah. 2007. New particle swarm optimizer with sigmoid increasing inertia weight. *International Journal of Computer Science and Security* 1:35–44.

Martinelli, A. 2002. The odometry error of a mobile robot with a synchronous drive system. *IEEE Transactions on Robotics and Automation* 18 (3): 399–405.

Mitchell, T. M. 1997. *Machine learning*. Boston, MA: McGraw-Hill.

Napier, A., G. Sibley, and P. Newman. 2010. Real-time bounded-error pose estimation for road vehicles using vision. In *IEEE conference on intelligent transportation systems*, 1141–1146. Funchal, September 19–22.

Ozcan, E., and C. Mohan. 1999. Particle swarm optimization: Surfing the waves. In *Proceedings of the 1999 Congress on evolutionary computation, CEC 99*, Vol. 3. Washington, DC: IEEE Service Center.

Pant, M., R. Thangaraj, and A. Abraham. 2009. Particle swarm optimization: Performance tuning and empirical analysis. In *Foundations of computational intelligence*, vol. 3. Vol. 203 of Studies in Computational Intelligence ed. A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht, 101–128. Berlin, Heidelberg: Springer.

Pessin, G., F. S. Osório, J. R. Souza, F. G. Costa, J. Ueyama, D. F. Wolf, T. Braun, P. A. Vargas. 2012. Evolving an indoor robotic localization system based on wireless networks. In *Proceedings of the 13th international conference on engineering applications of neural networks (EANN)*, London, UK, September 20–23, 2012, 61–70.

Pessin, G., F. S. Osório, J. Ueyama, J. R. Souza, D. F. Wolf, T. Braun, and P. A. Vargas. 2011. Evaluating the impact of the number of access points in mobile robots localization using artificial neural networks. In *Proceedings of the 5th international conference on communication system software and middleware.* 10:1–9.

Robles, J., M. Deicke, and R. Lehnert. 2010. 3d fingerprint-based localization for wireless sensor networks. In *International Workshop on Positioning navigation and communication* (WPNC). Dresden, Germany, March.

Silveira, T., H. de Oliveira, R. Salgado, L. da Silva, and G. Mateus. 2010. Cosine function applied to the inertia control in the particle swarm optimization. In *2010 IEEE congress on evolutionary computation (CEC)*, 1–8. Barcelona, Spain, July 18–23.

Teoh, E., K. Tan, and C. Xiang. 2006. Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. *IEEE Transactions on Neural Networks* 17 (6): 1623–1629.

Thrun, S., D. Fox, W. Burgard, and F. Dellaert. 2001. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence* 128 (1–2): 99–141.

Yao, X. 1999. Evolving artificial neural networks. In Proceedings of the IEEE 87 (9): 1423–1447.