Mobile Robot Indoor Localization Using Artificial Neural Networks and Wireless Networks

Gustavo Pessin[α], Fernando S. Osório[α], Jó Ueyama[α], Denis F. Wolf[α] and Torsten Braun[β]
[α]Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP)
São Carlos, SP, Brazil
[β]Institute of Computer Science and Applied Mathematics
University of Bern
Bern, Switzerland
Email: {pessin, fosorio, joueyama, denis}@icmc.usp.br, braun@iam.unibe.ch

***Abstract*—Accurate position information of an agent (i.e. robot, animal, or people) is a requirement to accomplish several tasks. Some sensors like GPS provide global position estimation but it is restricted to outdoor environments and has an inherent imprecision of a few meters. In indoor spaces, other sensors like lasers and cameras can be used for position estimation, 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 view, which makes the localization task harder. In the case of video cameras, the variation of light is also a serious issue. Nowadays Wireless Networks (WN) are widely available in indoor environments and allow efficient global localization demanding relatively low computing resources. Other advantages of this approach are scalability, robustness, and independence of specific features of the environment. However, the inherent instability in the wireless signal does not allow its direct use for very accurate position estimation. In this paper we evaluate the use of an Artificial Neural Network (ANN) to improve the estimation of the position of a mobile node in indoor environment using data provided by wireless networks. Our approach uses the ANN capabilities of learning and generalization to reduce the effect of the unstable data, increasing the accuracy of the agent's position. In order to validate our approach several ANN topologies have been evaluated in experimental tests performed with a mobile node in an indoor space.**

I. Introduction

Correctly estimating its own location is a prior assumption to accomplish several tasks in autonomous mobile robotic area. Also, knowledge about location can be used to track animals and people (i.e. to track the movement of the people while practicing sports). Sensors like GPS provide global position estimation but it is restricted to outdoor environments and has an inherent imprecision of few meters. The use of GPS is quite common in outdoors as a primary source of position, while more accurate estimation is obtained by fusion of other sensors, like lasers and cameras [1], [2].

In indoor spaces, other sensors like lasers and cameras can be used for pose estimation [3], [4], 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 view, which makes the localization task harder. In the case of video cameras, the variation of light is also a serious issue. Another possibility is an odometer, which provides useful information in some cases [5], [6] but it has an incremental error that usually invalidates their use in real systems.

Wireless Networks are widely available in indoor environments and allow efficient global localization demanding relatively low computing resources. Other advantages of this approach are scalability, robustness, and independence of specific features of the environment. However, the inherent instability in the wireless signal does not allow its direct use for accurate position estimation. One machine learning technique that could reduce the instability of the signals of the WN are Artificial Neural Networks, given its capability of learning from examples, and generalization and adaptation of the outputs. This is a method largely used in applications that require approximation, prediction or classification [7].

The main objective of this paper is the evaluation of ANNs to obtain the position of mobile nodes using measurements from wireless devices (802.11b/g). The measurements from the wireless network are Received Signal Strength Indication (RSSI) and Link Quality Indicator (LQI). These values are used as input of an ANN to learn the location without any another information or any requirement of mathematical approach. We evaluate several topologies of ANNs and also evaluate a simple technique to reduce the error in the location using average of multiples measurements from the wireless network.

This paper has the following structure: Section II introduces a short theoretical description and applications of artificial neural networks and wireless networks. Section III presents the methodology used to create and evaluate the experiments. Section IV describes the evaluation of all performed experiments. The last section presents the conclusion and the future perspectives of the presented work.

II. Theoretical Background

A. Artificial Neural Network

An Artificial Neural Network (ANN) is a collection of units (neurons) connected by weighted links (synapses). Input and output units receive and transfer signals from the environment to the environment. Internal units are called hidden because they do not have contact with the external environment [8]. The basic attributes of an ANN can be divided into topology and neurodynamics. The topology determines the structure of the network, i.e. the number of neurons and their interconnectivity. The neurodynamics defines the functional properties of the network, that is how it learns, recovers, combines and compares new information with knowledge already stored [9]. Mathematically, ANNs are universal approximators, that perform mappings in multivariable functions spaces [10].

The ability to learn and generalize[1] is one of the major advantages of ANNs, which gives it a power far beyond the simple direct mapping inputs and outputs.

Neural networks are widely used in applications that require approximation, prediction or classification. The work [11] presents an ANN able to perform the navigation of a robot in a simulated two-dimensional environment. The ANN controls the direction of the robot to areas with lower density of occupation by vegetation, the inputs of ANN are the vegetation densities observed and the output is the angle to which the agent should move. [12] presents an ANN to perform the navigation of a robot in a simulated three-dimensional environment. The ANN inputs are information collected from sensors (location, orientation, distance to obstacles) and the outputs are the speed and angle to be applied in linear and angular motors, respectively. Moreover, the work [13] presents an ANN to classify navigable and non-navigable regions in images, the ANN inputs are attributes of color, as average of color channels and entropy. Other studies using ANNs can be seen in [7], [14], [15].

In this paper, the development of the ANN was done with the Stuttgart Neural Network Simulator (SNNS) [16]. The SNNS is an environment to develop topologies and to train ANNs which has a large number of learning algorithms, such as backpropagation, quick propagation, resilient backpropagation, among others. The core system is developed in C and its use can be made completely through command line, but also has an interface developed in JAVA (JavaNNS). An application package from SNNS, the SNNS2C, allows the conversion of the ANN into a C code, which can be easily inserted into another application.

B. Wireless Networks

In most cases, signals from wireless networks propagate omnidirectionally. It could be directional depending on the type of antenna. The power signal decreases related to the distance from the server station. Using trilateration, and at least 3 server stations, we could use a simple calculation considering RSSI to obtain the location, similar to GPS. But, unlike GPS, the signal from wireless presents more instability and suffers from more interferences [17], [18], [19].

In [20] it has been shown that absolute performance in localization, using wireless networks, depends on the environment configuration. Hence, different approaches could work better in different environments, such as using different kinds of signals or filters. Evaluations in large indoor sites (like a building) introduce more difficulties in the localization due to attenuation and reflection of the signals on the walls and the different sources of interferences. Wireless localization addressing localization inside a building can be found in [21] and [22].

Another approach in localization is the use of a Wireless Sensor Network (WSN). The main difference in this approach is that in WSNs there are large numbers of small sensors which extract information from the environment. The information acquired by the sensors can be considered as a fingerprint. It is an interesting solution, but it depends on many resources which could make the system expensive. Existing work using WSN to obtain localization can be found in [23] and [22].

Our approach uses only 4 access points to provide wireless signals, which minimizes the number of resources. Also, the localization is done inside a small area, like a room, not in the entire building.

III. Methodology

As mentioned earlier, we have evaluated the use of an Artificial Neural Network (ANN) to obtain the position of a mobile node in indoor environment using data provided by wireless network (IEEE 802.11b/g). Our approach uses the ANN capabilities of learning and generalization to reduce the effect of the unstable data (due to signal strength oscillation), increasing accuracy of the agent's position estimation.

The indoor environment used to obtain data and to deploy the mobile node can be seen in the Fig. 1 and 2. The working area of the mobile node[2] is inside a room and is represented as a Cartesian plane. There are 4 access points (APs), one in each corner of the plane. The mobile node is located inside the plane with one wireless card used to scan the networks and signals provided by the APs. The data used to train the ANN was collected in 8 readings from each marked point – displacement of 90cm (Fig. 2). Given the plan with 270cm x 270cm, it means 16 points to read, resulting in 128 readings.

In order to validate our approach, several ANN topologies have been evaluated in experimental tests performed with a mobile node in this indoor environment. The inputs of the ANN are the signals received by the mobile node antenna from the 4 statically positioned APs. The values obtained from the wireless networks are the Received Signal Strength Indication (RSSI) and the Link Quality Indicator (LQI). These values are obtained with the GNU/Linux command *iwlist*. As we use the *iwlist* command, there is no need to establish a connection (or login) with the different specific networks. The scan of the networks without a connection provides enough information to this evaluation. Without a connection, the system becomes easier to use, more lightweight and flexible.

[1]*Generalize* can be considered as the production of acceptable outputs for inputs not presented during learning.

[2]The Fig. 2 shows a little robot inside the plane but to scan the WNs and to obtain the data used as ANN input we used a mobile computer. The GNU/Linux command *iwlist* used to scan the networks is not yet implemented in the robot.

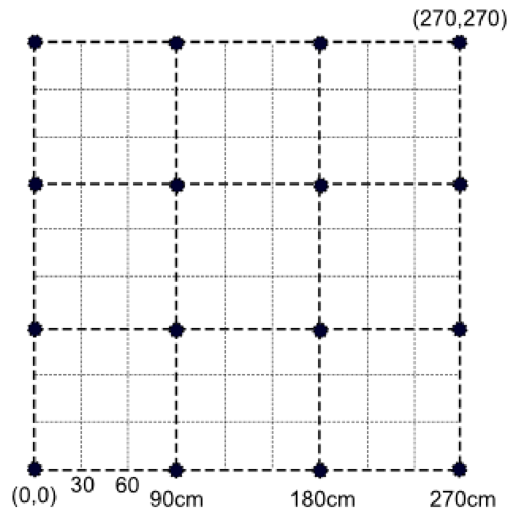

Fig. 1. Graphical representation of the working area. It represents an area with 270cm x 270cm.

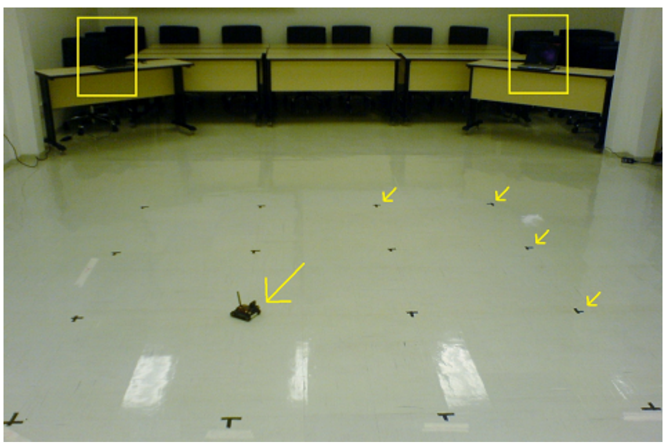

Fig. 2. Picture of the working area with the expected mobile node, similar to the representation presented in Figure 1. The yellow rectangles show two source of network signals (APs). Large arrow indicates the robot. Small arrows indicate the plan marks (each 90cm).

RSSI is a metric of the signal strength or signal power present in a received radio signal. Technically any device with wireless communication functionality provides it to the upper layers of the network stacks. The main advantage of using RSSI is its low cost. Given that every wireless device implements the possibility to deliver that value in its circuitry, there is no need of additional hardware development or adaptation. LQI is a metric that considers the RSSI and the environment noise [17], [18], [19].

We have considered three different ANN input data and input layer configurations. The first one considers only RSSI, the second considers only LQI while the third considers both RSSI and LQI. Also, we made evaluations with 8 different hidden layer configurations, considering 4, 8, 12, 16, 20, 24, 28 and 32 neurons. As we use 4 APs, the inputs of the ANN use one neuron for each network signal. The order is important, and hence, the AP 1 was associated to neuron 1, AP 2 with neuron 2 and so on. When the ANN has as an input only RSSI or only LQI, there are 4 input neurons. When the ANN has as an input RSSI and LQI, the input layer has 8 neurons.

The outputs of the network are two values, the coordinates (x,y) of the receiving antenna in the plane, a.k.a. the mobile node position. We train the ANN with the power signal of each source antenna expecting to get the position of the mobile node in the Cartesian plane. Therefore, after training the ANN, we could use it to obtain the localization and to track the displacement of the mobile node along a path.

The error is measured in centimeters, using the distance formula (distance between two points), as show Eq. 1.

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

IV. Experiments and Results

In the first step, we have evaluated the impact of using RSSI alone, LQI alone and both measurements together. These ANNs have also 8 different hidden layers, with 4, 8, 12, 16, 20, 24, 28 and 32 neurons. The result of training and validation can be seen in Tables I, II and III. The bold value in the tables indicates the best value (lower error) obtained. ANNs are susceptible to random values used in the initialization of the weights, so, we show in all tables the result of 5 different initializations and ANN training and validation.

We run the ANN training for 200.000 cycles, doing validation each 1.000 cycles. After 200.000 training cycles we use a script to find the cycle were the lower error occurred in the validation set (Optimum Generalization Point - OGP). In this way, Tables I, II and III show, beyond the MSE, the OGP of the respective ANN validation set.

We can see in Table II that the topology with 4 hidden neurons, using only RSSI as input, does not show learning capabilities. The error, both in training and in validation remains constant in this ANN for all random seeds used. For all ANN topologies with 8 and more hidden neurons the results appears to be more susceptible to the random seed used than to the network topology. The MSE ranges from $\approx 8,000$ to $\approx 10,000$ in most topologies. However, two samples with specific random seeds allowed to obtain MSEs smaller than 8,000. These ANNs were the ANN with 16 neurons in the hidden layers using RSSI, random seed 5 and the ANN with 16 neurons in the hidden layers using LQI, random seed 3.

As the tables show the Mean Square Error (MSE), and being this information not much human representative, we convert the result of the error of the trained ANN in centimeters. The error in cms can be seen in the Table IV. The results, in cms, does not show any major difference. We can see in Table IV that both the average error as the standard deviation differ by less than 10%. Therefore, we can not guarantee that there is a significant difference between using the one or the other topology.

As we consider this error somewhat large we try to reduce it using average of multiple readings of the wireless network signals as ANN input. Hence, the average of multiples readings of wireless networks were used as input of the ANN. We

TABLE I
RESULTS OF ANN TRAINING AND VALIDATION USING LQI.

ANN Topology		Mean Square Error Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
4x4x2	Cycle with OGP	12,000	8,000	11,000	21,000	9,000
	MSE Validation	12,765.10	9,931.66	9,761.12	10,087.30	9,961.20
	MSE Training	8,774.41	6,436.16	6,486.81	6,451.71	6,386.72
4x8x2	Cycle with OGP	26,000	6,000	10,000	13,000	16,000
	MSE Validation	8,052.01	10,177.60	10,455.70	9,335.31	10,146.70
	MSE Training	3,214.54	4,038.38	3,557.03	3,476.81	3,919.17
4x12x2	Cycle with OGP	8,000	14,000	30,000	29,000	10,000
	MSE Validation	10,173.60	9,223.82	9,031.10	8,192.70	9,510.75
	MSE Training	2,096.11	2,241.99	2,510.17	2,501.75	1,870.38
4x16x2	Cycle with OGP	11,000	36,000	45,000	11,000	4,000
	MSE Validation	10,267.00	9,152.31	**7,675.02**	10,390.70	9,733.44
	MSE Training	1,798.24	2,052.72	1,903.77	2,758.26	1,270.88
4x20x2	Cycle with OGP	38,000	133,000	12,000	20,000	42,000
	MSE Validation	8,503.22	8,892.63	9,564.22	8,690.54	9,771.28
	MSE Training	1,360.20	3,238.05	1,661.50	725.15	3,262.58
4x24x2	Cycle with OGP	15,000	6,000	37,000	150,000	24,000
	MSE Validation	8,257.35	10,654.60	9,012.88	8,603.57	9,758.70
	MSE Training	1,236.62	706.17	1,178.45	2,858.17	1,227.23
4x28x2	Cycle with OGP	50,000	7,000	13,000	20,000	17,000
	MSE Validation	8,623.52	10,407.40	8,417.83	8,570.47	9,849.23
	MSE Training	1,814.08	1,360.59	556.01	2,192.64	2,620.35
4x32x2	Cycle with OGP	2,000	26,000	26,000	14,000	8,000
	MSE Validation	10,174.30	8,986.58	10,273.70	10,684.90	10,210.20
	MSE Training	655.64	1,385.76	2,196.86	989.09	897.50

TABLE II
RESULTS OF ANN TRAINING AND VALIDATION USING RSSI.

ANN Topology		Mean Square Error Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
4x4x2	Cycle with OGP	1,000	1,000	1,000	1,000	1,000
	MSE Validation	19,521.00	19,521.00	19,521.00	19,521.00	19,521.00
	MSE Training	19,323.40	19,323.40	19,323.40	19,323.40	19,323.40
4x8x2	Cycle with OGP	2,000	3,000	2,000	4,000	1,000
	MSE Validation	10,332.10	8,983.84	10,462.70	10,425.60	10,154.10
	MSE Training	3,417.38	4,567.84	3,352.75	4,165.33	3,863.15
4x12x2	Cycle with OGP	5,000	3,000	5,000	8,000	3,000
	MSE Validation	9,088.69	10,545.10	10,888.70	9,306.96	10,173.80
	MSE Training	3,976.18	3,376.28	3,236.67	3,022.70	2,799.89
4x16x2	Cycle with OGP	7,000	3,000	16,000	5,000	28,000
	MSE Validation	9,878.54	9,175.65	10,032.70	10,179.70	**7,675.60**
	MSE Training	1,384.38	1,964.64	4,431.05	3,659.89	3,753.49
4x20x2	Cycle with OGP	10,000	20,000	3,000	1,000	5,000
	MSE Validation	10,194.10	9,524.93	10,252.30	11,649,8	9,617.74
	MSE Training	1,748.87	1,934.26	2,250.03	2,400.03	2,044.45
4x24x2	Cycle with OGP	1,000	14,000	1,000	4,000	17,000
	MSE Validation	10,851.90	9,271.33	10,829.10	9,497.56	8,467.30
	MSE Training	2,307.75	2,093.99	1,450.24	1,273.91	1,238.20
4x28x2	Cycle with OGP	7,000	36,000	3,000	3,000	7,000
	MSE Validation	8,736.25	9,591.05	9,871.50	9,250.73	8,497.69
	MSE Training	1,992.95	2,391.61	1,381.44	683.96	1,359.90
4x32x2	Cycle with OGP	2,000	1,000	6,000	18,000	2,000
	MSE Validation	9,092.92	9,813.68	10,327.40	9,229.80	10,130.40
	MSE Training	1,509.22	1,414.04	1,746.82	1,776.24	2,620.17

evaluate the average of 2, 4 and 6 readings of the wireless network signals (scans of wireless signals). In this evaluation, we use the best ANN topology chosen from the previous step. The best topology was the ANN with 16 hidden neurons and using as input LQI. We run also 5 times the training and validation of the ANN using different random seeds. Table V and Fig. 3 show the results, in centimeters, of the best ANNs.

We can see in Table V and in Fig. 3 that using as input in the ANN the average of the multiple readings from the wireless network, we could reduce the average error in the ANN learning from 73.04cm (without using average of scan) to 28.72cm (using average of 6 scans). It means a reduction of 60.68% in the average error. Also, it could reduce the standard deviation from 49.06cm (without using average of scan) to 14.76cm (using average of 6 scans). It means a reduction of 69.91% in the standard deviation.

TABLE III
RESULTS OF ANN TRAINING AND VALIDATION USING RSSI AND LQI.

		Mean Square Error				
ANN Topology		Seed 1	Seed 2	Seed 3	Seed 4	Seed 5
8x4x2	Cycle with OGP	3,000	6,000	14,000	9,000	3,000
	MSE Validation	10,062.10	9,928.30	10,120.80	10,385.60	10,675.30
	MSE Training	5,897.86	5,479.84	5,480.17	5,479.72	6,192.29
8x8x2	Cycle with OGP	5,000	2,000	3,000	1,000	2,000
	MSE Validation	10,317.20	8,920.89	10,415.20	9,979.49	9,109.60
	MSE Training	3,965.76	3,800.78	3,855.22	3,940.69	2,910.31
8x12x2	Cycle with OGP	2,000	4,000	1,000	19,000	1,000
	MSE Validation	11,594.30	10,043.60	11,054.20	9,091.86	10,664.60
	MSE Training	1,647.04	1,931.91	3,153.80	2,663.73	2,427.86
8x16x2	Cycle with OGP	7,000	1,000	5,000	4,000	2,000
	MSE Validation	8,604.26	10,516.80	9,172.27	**8,507.51**	10,414.50
	MSE Training	1,242.66	1,469.67	1,066.27	1,596.17	1,325.05
8x20x2	Cycle with OGP	5,000	1,000	5,000	3,000	5,000
	MSE Validation	9,130.13	10,599.90	8,951.33	9,131.57	10,207.30
	MSE Training	1,306.92	1,853.70	1,456.52	965.96	1,846.93
8x24x2	Cycle with OGP	1,000	1,000	1,000	1,000	1,000
	MSE Validation	10,104.70	9,673.64	9,737.22	10,607.80	11,620.80
	MSE Training	509.35	505.88	522.17	851.61	625.17
8x28x2	Cycle with OGP	1,000	2,000	2,000	4,000	1,000
	MSE Validation	10,446.40	9,094.02	10,753.00	9,947.86	9,656.12
	MSE Training	436.23	402.34	395.05	603.34	1,022.99
8x32x2	Cycle with OGP	5,000	1,000	2,000	1,000	2,000
	MSE Validation	10,567.50	10,618.50	8,558.21	8,961.94	8,956.89
	MSE Training	302.39	150.07	571.63	468.40	654.63

TABLE IV
RESULTS OF THE BEST ANNS USING DIFERENT INPUTS.

	ANN Inputs		
	LQI	RSSI	LQI and RSSI
Average error (cm)	73.04	75.75	80.94
Standard deviation (cm)	49.06	44.63	44.85
Bigger error (cm)	187.57	197.47	197.49
Lower error (cm)	14.34	16.96	8.14

TABLE V
RESULTS OF THE ANNS USING THE AVERAGE OF MULTIPLES SCANS.

	ANN Inputs			
	Unique scan	Average of 2 scans	Average of 4 scans	Average of 6 scans
Average error (cm)	73.04	60.10	48.82	28.72
Std. dev. (cm)	49.06	38.48	39.69	14.76
Bigger error (cm)	187.57	171.93	150.90	51.43
Lower error (cm)	14.34	3.06	3.11	5.88

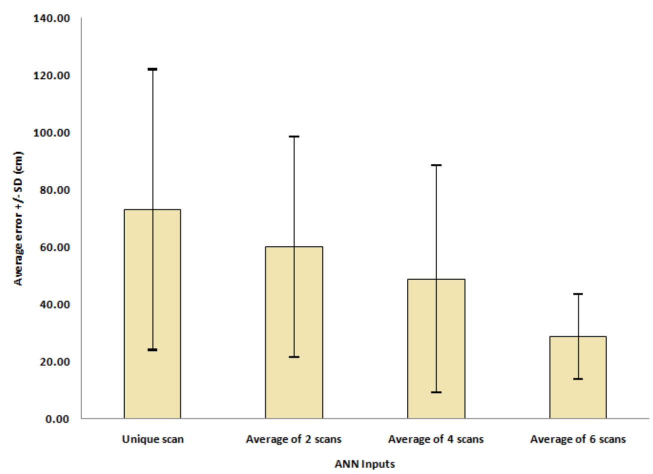

Fig. 3. Results, in centimeters, of the ANNs trained with average of multiples scans. The graph show average error +/- standard deviation.

We use the mobile node to traverse a path (and get the track) in the environment, considering wireless scanning each 90cm x 90cm displacement. The paths can be seen in Fig. 4. The results are not quite good yet, but Fig. 4(b) present results significantly better than Fig. 4(a). We can see from Fig. 4 that the tracked path improves with the use of averages.

V. CONCLUSIONS

Accurate position information of an agent (i.e. robot, animal, or people) is a requirement to accomplish several tasks. Some sensors like GPS provide global position estimation but it is restricted to outdoor environments and has an inherent imprecision of few meters. In indoor spaces, other sensors like lasers and cameras can be used for pose estimation, 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 view, which makes the localization task harder. In the case of video cameras, the variation of light is also a serious issue. Nowadays Wireless Networks are widely available in indoor environments and could allow efficient global localization demanding relatively low computing resources. However, the inherent instability in the wireless signal does not allow its direct use for accurate position estimation.

In this paper we have evaluated the use of an Artificial Neural Network to obtain the position of a mobile node in

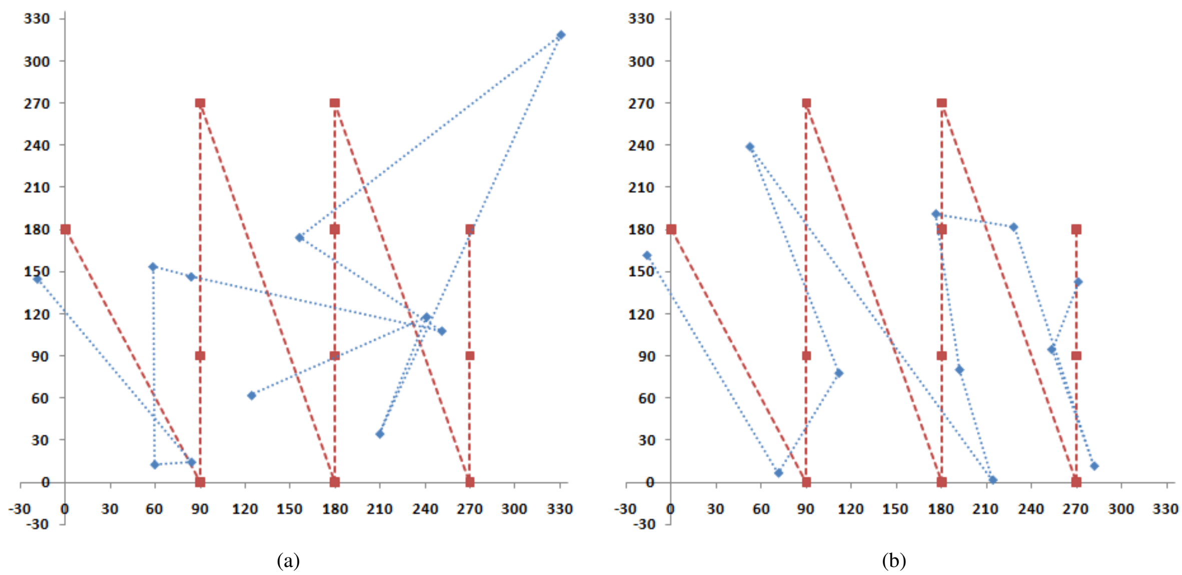

Fig. 4. Paths using the best trained ANNs. (a) Just one scan in the ANN input. (b) Average of 6 scan in the ANN input. Red line: original path. Blue line: ANN tracking.

indoor environment using data provided by wireless network. We evaluate several topologies of ANNs. To further reduce the error, we evaluated the use of average of multiples wireless scanning, which allowed reducing radius error average in 60.68%. Finally, we used the ANN to get the current position of a mobile node performing a path, in which we track the robot with an average error of 28.72cm.

The main future work planned is to seek improvements in the system to obtain, beyond the position, the orientation of the mobile node. In this way we plan use another wireless technologies, like Bluetooth and ZigBee and make evaluations with more than 4 Access Points.

Acknowledgment

The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology – Critical Embedded Systems – Brazil), processes 573963/2008-8 and 08/57870-9. Also acknowledge CAPES and FAPESP for their financial support of this research (doctoral grant). And acknowledge Diego F. Sciotti due to the help in setting up the test environment.

References

[1] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge: The Great Robot Race*. Springer, 2007.

[2] ——, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Springer, 2009.

[3] H. Lang, Y. Wang, and C. de Silva, "Mobile robot localization and object pose estimation using optical encoder, vision and laser sensors," in *IEEE Int. Conf. on Automation and Logistics*, 2008, pp. 617–622.

[4] A. Napier, G. Sibley, and P. Newman, "Real-time bounded-error pose estimation for road vehicles using vision," in *13th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC)*, 2010, pp. 1141–1146.

[5] A. Martinelli, "The odometry error of a mobile robot with a synchronous drive system," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 3, pp. 399 –405, Jun. 2002.

[6] K. S. Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *IEEE Int. Conf. on Robotics and Automation*, vol. 4, 1997, pp. 2783–2788.

[7] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[8] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, 2000.

[9] S. V. Kartalopoulos, *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*. Wiley, 1995.

[10] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[11] G. Pessin, F. S. Osório, S. R. Musse, V. Nonnenmacher, and S. S. Ferreira, "Utilizando agentes autônomos com aprendizado para a identificação e combate de incêndios em Áreas florestais," in *VII Simpósio de Informática do Planalto Médio (SIPM'07)*, 2007.

[12] G. Pessin, F. S. Osório, A. Y. Hata, and D. F. Wolf, "Intelligent control and evolutionary strategies applied to multirobotic systems," in *IEEE-ICIT 2010 Int. Conf. on Industrial Technology*, 2010, pp. 1427–1432.

[13] D. Sales, P. Shinzato, G. Pessin, F. Osório, and D. Wolf, "Vision-based autonomous navigation system using ann and fsm control," in *7th Latin American Robotics Symposium (LARS)*, 2010, pp. 1–6.

[14] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 30, pp. 451–462, 2000.

[15] A. P. Braga, A. P. L. F. Carvalho, and T. B. Ludermir, *Redes Neurais Artificiais: Teoria e aplicações*. Rio de Janeiro, RJ, Brasil : LTC, 2007.

[16] A. Zell, G. Mamier, M. Mache, R. Hübner, S. Herrmann, T. Soyez, M. Schmalzl, T. Sommer, A. Hatzigeorgiou, D. Posselt, T. Schreiner, B. Kett, G. Clemente, and J. Wieland, "Stuttgart neural network simulator - user manual, version 4.1 (report no. 6/95)," University of Stuttgart, Tech. Rep., 1995.

[17] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2002.

[18] J. Ross, *The Book of Wireless: A Painless Guide to Wi-Fi and Broadband Wireless*. No Starch Press, 2008.

[19] M. Jacobsson, I. Niemegeers, and S. H. de Groot, *Personal Networks: Wireless Networking for Personal Devices*. Wiley, 2010.

[20] E. Elnahrawy, X. Li, and R. Martin, "The limits of localization using signal strength: a comparative study," in *IEEE Com. Society Conf. on Sensor and Ad Hoc Communications and Networks*, 2004, pp. 406–414.

[21] P. Espinace, A. Soto, and M. Torres-Torriti, "Real-time robot localization in indoor environments using structural information," *Latin American Robotics Symposium and Intelligent Robotics Meeting*, pp. 79–84, 2008.

[22] M. Vieira, R. Govindan, and G. Sukhatme, "Scalable and practical pursuit-evasion with networked robots," *Intelligent Service Robotics*, vol. 2, pp. 247–263, 2009.

[23] Y. Sun, J. Xiao, and F. Cabrera-Mora, "Robot localization and energy-efficient wireless communications by multiple antennas," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 377–381.