Associative Search Network for RSSI-Based Target Localization in Unknown Environments

V. Loscrí[1], S. Guzzo Bonifacio[1], N. Mitton[1], and S. Fiorenza[2]

[1] Inria Lille - Nord Europe, France
[2] University of Calabria, Italy

Abstract. Received Signal Strength Indicator (RSSI) is commonly considered and is very popular for target localization applications, since it does not require extra-circuitry and is always available on current devices. Unfortunately, target localizations based on RSSI are affected with many issues, above all in indoor environments. In this paper, we focus on the pervasive localization of target objects in an unknown environment. In order to accomplish the localization task, we implement an Associative Search Network (ASN) on the robots and we deploy a real test-bed to evaluate the effectiveness of the ASN for target localization. The ASN is based on the computation of weights, to "dictate" the correct direction of movement, closer to the target. Results show that RSSI through an ASN is effective to localize a target, since there is an implicit mechanism of correction, deriving from the learning ASN approach.

Keywords: ASN, Target Localization, RSSI, robots, experimentation.

1 Introduction

In the recent years, research communities have focused and considered pervasive target localization [1] and cooperative target localization. Target localization can be also envisaged as a sub-problem of coverage of specific areas, where events of interest occur [17], and where the correct and timely localization is mandatory. Among all the parameters that a wireless device (e.g. a robot) can measure, the Received Signal Strength Indicator (RSSI) is one of the most popular considered for target localization [14] [15], above all in the context of Wireless Sensor Networks. Its popularity is due to different factors, such as it is always available between communicating devices, and it does not require extra-circuitry that would result in higher costs and energy consumption. Furthermore, the availability of the RSSI measure on all the devices, makes possible the implementation of a localization technique for heterogeneous nodes. This latter feature increases the potential scalability of this kind of approach, as envisaged in [16].

In this paper, we propose to exploit the RSSI parameter to localize a target in an unknown environment. The localization technique is based on an Associative Search Network (ASN) [2] and is performed in indoor environments. The network we implement on top of our robots is Hopfield-inspired [6] and we will show that it shares, with this type of system, the capability to converge towards stable

N. Mitton et al. (Eds.): AdHocNets 2015, LNICST 155, pp. 146–157, 2015.
DOI: 10.1007/978-3-319-25067-0_12

states. By providing our robots with learning capabilities, we make RSSI a viable solution for target localization. One of the main premises related with the system developed, is that the computation of the weights for the direction decisions, is performed without any external oracle. This feature comes down to provide our devices with the possibility to dynamically adapt the weights without the intervention of an external central unit. In practice, the resulting system will be totally distributed and evolutionary by dynamically adapting its behavior to the external conditions. As a proof of concept of the proposed approach, we developed the whole ASN on Arduino-based mobile robots. These robotic platforms have been built from scratch and are totally reprogrammable. The brain of the robot is a mini-pc.Even if the robots are also able to communicate with each others, in this context, we have only used the communication paradigm to assign the task (i.e. the identity of the target to be localized). The Arduino module is mainly used to "transfer" the movement commands to the four wheels and to implement all the components related to the movement. In summary, the main contributions of the paper are as follows:

- a new Received Signal Strength (RSSI)-based Associative Search Network (ASN) for target localization;
- an evaluation though ASN implementation on real hardware (Arduino robots).

The rest of the paper is organized as follows. Section 3 browses the literature for the three macro topics tackled in this paper : *a*) Target Localization, *b*) Associative Search Networks and *c*) RSSI use in localization processes. In Section 4, we state the Associative Search Network problem (ASN) for target localization. Section 5 details our contribution. Section 6 describes the scenario considered for evaluation purpose and the test-bed deployment. Section 7 gives the main features of the ASN implementation on robots. Finally, Section 8 concludes this work and explores future research paths.

2 The Associative Search Network Problem

In this section, we introduce the ASN Problem. We start by considering an ASN that can be defined as a system where there is no outside process that suggests the correct association between a pattern with a key. Instead, for each key, the associative system searches for the pattern that minimizes a reinforcement signal (i.e. a payoff). The system is able to store, by the mean of an associative memory, the results of reinforcement feedback coming from the environment. In the ASN, is not considered at all the presence of a "oracle", that has to provide the pattern to be stored. This feature makes this system similar to an Hopfield-network [6]. Another important feature is that it does not require an a priori knowledge about the best associations. ASN combines two learning methods that are usually considered separately: *(i)* a pattern recognition mechanism to respond to each key with the appropriate output pattern and *(ii)* a stochastic automaton method to maximize a reinforcement signal or payoff [18] [19]. If we consider that the ASN interacts with an environment E, at each time unit

t, E provides a context vector $X(t) = (x_1(t), x_2(t), ..., x_n(t))$, where $x_i(t)$ is a positive real number and n is the number of inputs. E will also provide a payoff or reinforcement learning $z(t)$. The ASN produces an output pattern $y(t) = (y_1(t), ...y_m(t))$, where m is the number of outputs, where each $y_i(t) \in \{0, 1\}$ and is received by E. In practice, the vector $X(t)$ is to provide information about the environment to the ASN. Different contexts may require different actions from the ASN. As an example, we can consider a mobile device that is required to reach a specific target. The context is represented by an estimation of the distance. After a movement, a different context will require a different action, that could correspond to a request of changing direction (this is a different action required). The reinforcement signal is intended as a kind of evaluation of how much an action is appropriate in a certain context. A more appropriate formulation of the problem could be: let us assume that $X(t)$ belongs to a finite set $X = (X^1, ..., X^k)$ of context vectors and let also assume that to each $X^\alpha \in X$ corresponds a payoff or reinforcement function Z^α. If E always evaluates an output vector in one time step and if $X(t) = X^\alpha$, then $Z(t + 1) = Z^\alpha(Y(t))$. This means that E provides a "training sequence" over X if it implements an infinite sequence of payoff functions and emits the corresponding sequence of context vectors $X^{i1}, X^{i2}, ..., X^{il}$. Each $X^{il} \in X$ and each element of X occurs infinitely often. The termination condition of the associative search problem is solved when, after some finite portion of a training sequence, the ASN responds to each $X^\alpha \in X$ with the output pattern $Y^\alpha = (y_1^\alpha, ...y_m^\alpha)$ which maximizes Z^α. As outlined in [2], the output vectors required from the system are only based on scalar feedback. Other mechanisms that are also able to solve similar problems, such as perceptrons based mechanisms [4], have some counterparts, e.g. they require a separate error feedback. The basic unit of an ASN is the adaptive element and in the simplest version, an ASN can be regarded as constituted by a single adaptive element. Let us indicate with $x_i, i = 1, ..., n$ the context input, z represents the payoff (or reinforcement signal) and y is the output. Every context input x_i is associated with a weight $w_i(t) \in R$. Let assume $W(t)$ is the vector of weights at time t and $s(t)$ is the weighted sum at time t of the contexts inputs. We obtain $s(t) = \sum_{i=1}^{n} w_i(t)x_i(t) = W(t)X(t)$ and the output $y(t)$ is

$$y(t) = sign(t) = \begin{cases} 1 & \text{if } s(t) + noise > 0, \\ 0 & \text{if } s(t) + noise = 0, \\ -1 & \text{if } s(t) + noise < 0. \end{cases} \tag{1}$$

where *noise* is a random variable with zero mean normal distribution. The element's output depends on the value s. If s is positive $y(t)$ will be more likely 1, otherwise it will be more likely 0. The update of the weights is governed by the following equation, at each time step:

$$\begin{aligned} w_i(t + 1) = w_i(t) + c[z(t) - z(t - 1)] \\ [y(t - 1) - y(t - 2)]x_i(t - 1), \\ i = 1, ..., n \end{aligned} \tag{2}$$

where c is a constant determining the learning rate. The more the value of n increases the more the accuracy increases too, but also the complexity will increase and, with it, the occupancy of the memory. In this equation, the response latency is considered equal to zero. It is worth noticing that, when the simplest ASN with only a single adaptive element is considered, the search of the optimal action is determined by two possible actions by the ASN. However, in a larger ASN (with more than 1 adaptive element), the adaptive elements exploit their capability of operating in an effective way, with random payoff response characteristics. From 1, we can observe that the change of the payoff signal z is used by the adaptive element for determining weight changes. Of course, if the payoff (or reinforcement signal) changes at every time step and it does not vary smoothly over time, an adaptive element that implements the rules 1 and 2, is not capable to solve an associative search problem.

3 Related Work

The issue addressed in this paper is characterized by different properties, the target localization, the ASN and RSSI-based localization. Target Localization represents the main goal that the robot has to fulfill, the ASN is the main technique used in this work to reach the goal and the Received Strength Signal Indicator is the parameter used to obtain information from the surrounding environment. To the best of our knowledge, combining these three components is new. We thus briefly survey the literature for these three topics independently.

Target Localization. A special category of target localization, named Anchor Based considers a subset of nodes, placed in fixed coordinates, called Anchor Nodes. Through the exchange of messages among those nodes and a target, it is possible to estimate the position of the target inside the monitored area. Such a localization can have different purposes, like [7] where measured RSSI is used as parameter in maximum likelihood estimation algorithm. A more complex use of this technique can be found in [8] where the RSSI is used to determine the position of a robot using a distributed algorithm and to direct it to follow a path. In [11], the authors consider the target localization problem based on range measurement by considering a single mobile robot or several cooperating robots. Similarly to our approach, the authors consider that the robots do not have access to global knowledge about the environment. They do not know their current position, do not share a common sense of direction, etc. The robot is asked to estimate the relative coordinates of the target. Anyway, the approach they consider is totally different, since it is based on a specific filter.

In [12], the authors consider a set of mobile robots able to localize a set of unknown static targets within a known obstacle map. The robots use measurement Probability Hypothesis Density, or PHD, filter to collect the information for localization purpose. The authors do not make reference to RSSI as viable parameter to localize the targets. Moreover, the main difference is that the authors in [12] assume they know the obstacle map. In our case we introduce an

explicit mechanism to detect and avoid the obstacles. The purpose is yet different and the aim is to use anchor nodes (if available), to help a robot in the localization process of an active target without a priori knowledge of the map, the geographic coordinates, etc. and without a oracle that manages the right weights associated with the weight matrix to compute the output.

Associative Search Network. ASN has been introduced by Barto and Sutton [2] as a learning process. It interacts with an environment (E) receiving from it a feedback link. Through the analysis of this feedback and a reinforcement payoff signal the ASN is able to learn the best actions to perform in order to maximize the payoff function and reach a goal. This technique has been further analyzed in [9], where the authors define a simulation environment to solve the practical problem called Hill Climbing. In the scenario depicted in this example a robot had to climb to the top of a hill where a tree is placed. The operations of the robot are assisted by the presence of some landmarks. Therein, the authors show how the presence of the landmarks and the use of ASN can improve the efficiency. Our work aims to replicate this scenario in a real environment, rather than a simulated one, to analyze the difference. We were mainly focused to identify the assumption that remains valid and the ones that is necessary to reduce when moving from a simulated scenario to a real one.

RSSI Use in Localization Processes. The Received Signal Strength Indicator (RSSI) is often used as a parameter in target localization algorithms because of its relationship with the Path Loss Model, also known as Friis transmission equation [3]. Thanks to this equation it is possible to calculate the power of a transmitted signal at a fixed distance d from the transmitter. It can be used as well to obtain the distance, from a transmission point, given the power of the received signal. The use of RSSI for distance estimation may present some drawbacks, especially for indoor localization, due to the presence of multi-path fading, shadowing and scattering which affects the transmitted signal as depicted in [10]. Heurtefeux and Valois outline that the great popularity of localization protocols based on RSSI (in Wireless Sensor Networks) is mainly due to the fact that no extra hardware is required and the theory formulates the RSSI in terms of distance function, but the disadvantages can make its use for Localization Target purpose unfeasible. In [13], the authors show the effectiveness of a navigation technique based on RSSI. Similarly to our approach, the orientation adjustment and the motion tracking are performed through the help of other nodes (sensors nodes in their case). The main difference is that they consider sensors distributed in a grid pattern. We also make reference to landmarks that are arranged in the middle of the side of a rectangle, but our approach is able to dynamically adjust and learn about the landmarks even if they are arranged in different positions and are different in number.

4 Target Localization Problem Statement

The main goal of this work is the target localization in an unknown environment, based on the Received Signal Strength Indicator (RSSI). We provide our robots

with an ASN and formalize the Target Localization as an Associative Search Network problem. By the definition of the appropriate weights, the robot will move in a weighted space, instead of a physical space. This means that, the robots will perform a specific action based on the ASN implemented on them. We got inspiration from the work of Barto and Sutton [2], where the authors show how a simple network could be used to model a learning approach based on reference points (landmarks). Specifically, the movement rules defined by Barto and Sutton, are based on distinct olfactory gradients (emitted by the reference points).

By considering a payoff function z, that is maximized when the robot reaches the objective and is decreasing while the distance (between the robot and the target) increases, it is possible to show that the robot is able to find the right path and reach the target. Of course, by providing the system only with the payoff function makes the movement of the robot less precise and the overall process longer. In our reference model, the network is constituted by 4 input units and 4 output units. The input unit i can assume the values $North$, $South$, $East$ and $West$, the context input $x_i(t)$ is the signal emitted by the correspondent reference point and y represents the output of the ASN system. Every input unit is completely connected to each output unit j, where $j = North$, $South$, $East$ and $West$. In this way, each input unit can "modulate" or adapt 4 connection weights $w_{ji}(t)$ in the connection matrix by following the equation 2. Each weight encodes a confidence degree in such a way that, when the robot is close to a reference/landmark point i, it should proceed towards the direction j, closer to the target.

The confidential degree $s_j(t)$ (in order to move in direction j) is computed as the sum of the products of the current weights and signals received by the reference points as

$$s_j(t) = w_{0j}(t) + \sum_i w_{ji}(t)x_i(t), \tag{3}$$

where $w_{0j}(t)$ is a polarization term. The weights w_{0j} are updated as follows:

$$w_{0j}(t+1) = f[w_{0j}(t) + c_0(z(t) - z(t-1))y(t-1)], \tag{4}$$

where

$$f(x) = \begin{cases} BOUND & \text{if } x > BOUND, \\ 0 & \text{if } x < 0, \\ x & \text{otherwise} \end{cases} \tag{5}$$

f bounds each w_{0j} to the interval $[0, BOUND]$. c, c_0 and $BOUND$ are positive real numbers. The rule as defined in 5 is necessary to allow the ASN to correctly work also in the absence of landmark information. More details about values and impact of c, c_0 and $BOUND$ can be found in [2].

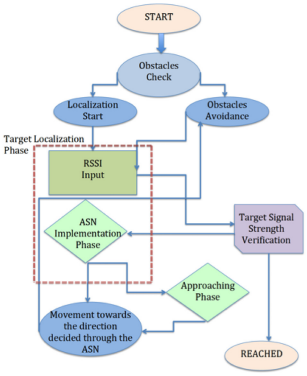
In our specific case, if our robot is close to the reference point $North$, the output unit $South$ will be activated and next the robot will head to $South$. Furthermore, if the robot is in the quadrant $South - West$, the output units

of *North* and *East* will be activated and the next step of the robot will be in the North-East direction. The robot has to learn the appropriate weights by implementing the rule as in Eq. 2. In this case, a connection weight w_{ji} will change if and only if a movement towards direction j, i.e. $(y_j(t-1) > 0)$, is executed and the robot is close to a reference point $i(x_i(t-1) > 0)$. By considering $z(t)$ as a measure of the closeness of the target, we can observe that w_{ji} increases when z increases, yielding that direction j makes the robot to move towards the right direction. In this case, a movement j is likely to occur again. On the other hand, whether w_{ji} decreases, the function z also decreases and the robot will move towards the wrong direction.

5 Our Target Localization Algorithm

This section details our Localization Algorithm, detailing the rules that drive the movement of the robot. The main goal for the robot is to "detect" the target (receive a signal strength with a sufficient level from the target) and to move to reach it without a priori knowledge about the environment.

In order to correctly move, the robot has to implement a behavioral logic, based on the input information it receives (e.g. the RSSI) and the elaboration of the inputs through the ASN.



(a) Steps of the behavior of a robot.

(b) Real scenario.

The Localization Algorithm is split in two phases: the 1) ASN implementation Phase and the 2) Approaching Phase, as shown in Fig. 5. During both phases, an underlying obstacle avoidance process, detailed in the next section, is running.

We assume the target to be reached is always turned-on and that the robot is moving in the room where the target is. Based on these assumptions, our algorithm terminates when, based on the inputs received the robot estimates it has reached the target. The robot checks for the list of tasks and implements the first one. It enters the ASN Implementation Phase, which consists in receiving the "signals" from different devices and analyzing them. If the robot does not

overhear its target, it moves forward for an arbitrary time Δt (e.g 1 or 2 sec.) by following a Random Way Path (RWP)[5]. It travels a prefixed distance, then it stops and checks for listening the devices. It repeats this process while it does not hear the target device.

Algorithm 1. Localization Algorithm

• **Local variables:** TargetFound = TargetReached = FALSE; List List of tasks;
RSSI-based target localization
while List $\neq \emptyset$ **do**
  $T \leftarrow$ POP(List) {Move to next task/target T in the list}
  while (TargetReached == FALSE) **do**
    if TargetFound **then**
      Collect x_i {Collect RSSI signal x_i from T and from landmarks L}
      DIR$\leftarrow$ASN-Localization(x_i); Move in DIR direction
      if TargetReached==TRUE STOP {task completed;} and Remove task from List
    else
      $\Delta t \leftarrow$ Random(). Move forward for Δt at speed s.
    end if
  end while
end while
ASN-Localization(x_i)
Compute s_j $\forall j$ {Solve Eq.2 and 3}
Return i such that $s_i = max_{\forall j} s_j$

6 Performance Evaluation

In this section, we describe the test-bed considered to assess the performances of the ASN-based target localization. First, we detail the entities considered to realize the proof-of-concept and then we describe the scenario.

6.1 The "Entities" Involved

The reference scenario we implemented is characterized with heterogeneous items. Specifically we have:
1) The target node : TP-LINK Router Wireless N300;
2) 4 landmarks - 1 NETGEAR Wireless Router MR314, 1 ALICE Gate2 Plus Wi-Fi and 2 notebooks HP 630 (hotspots);
3) Rovers (robots equipped with wheels to support mobility).

We placed the two hotspots in the points *West* and *East* of the area, the target in the centre of the area and at *North* and *South* we put the other routers. The reference scenario is represented by an area of $15m^2$ ($3 \times 5m$) as shown in Figure 1(b). The received power value has been obtained both from the anchor nodes (landmarks) and the target through the command *Iwlist*, that is available in the Linux platform as part of the *wireless-tools* library. Through this package, it is possible to have a set of commands to control the wireless devices based on the standard 802.11. The *Iwlist* command is used in combination with some parameters to better specify the data requested by a user. It details, for every detected network, a set of data related to the ESSID of the network, the

quality of the signal, the transmission channel frequency, and the Received Signal Strength Indication (RSSI).

To the follow we detail each step of the algorithm as implemented in our Rover and shown in Figure 5.

Obstacle Check and Obstacle Avoidance. Every robot runs an obstacle detection and avoidance based on ultrasound sensors. Once an obstacle is detected, the robot bypasses it. To do so, the robot scans the area on the right side with an angle α°. If the obstacle is still there, the robot scans the area with an angle $2 \times \alpha^\circ$ on the left side. If the obstacle is still there, the robot moves backward, turns α° right again and resumes its previous movement (either in searching or approaching phase).

Localization Start. If the robot does not detect any obstacle, it runs the *Localization Phase*, by acquiring the signal inputs. If the robot individuates the target ID among the signals received, it compares the RSSI value with a *threshold* value. If the signal RSSI results greater than a certain *threshold*, the robot estimates the target as reached, otherwise the robot enters the ASN-Localization algorithm.

ASN Implementation Phase This is the core of the algorithm. Based on the received input signals $x_1, x_2, ..., x_n$, the ASN will output a specific action that corresponds to a specific direction towards which the robot will move. The output signals $y_1, y_2, ..., y_m$ will constitute the new RSSI values, deriving from the new position of the robot. The payoff z represents the *closeness* of the robot from the target.

6.2 Results

In order to evaluate the ASN technique proposed, we realized a proof-of-concept based on a testbed as described in 6.1 and a variable number of robots (ranging from 1 to 3). We build this Arduino platform from scratch and we equipped it with a mini-pc, that constitues the *"brain"* of our robot. We performed three types of experiments, every result is the average over more than 30 runs. The parameters we evaluated are the time needed to reach the target (Delay) and how close the robot is positioned from the target when the algorithm exits (Dist). Numerical results are reported in Table 1.

Table 1. Results. Delays are in sec, distances in cm

Robot 1		Robot 2		Robot 3	
Delay	Dist.	Delay	Dist.	Delay	Dist.
61,6	26	–	–	–	–
65,7	25,5	80,5	31,3	–	–
69,6	26,5	82,2	33	93,4	39,7

7 Characterization of Our ASN-Based Target Localization Technique

In this section, we summarize the main features of our ASN system and we discuss some derived properties. Our system is characterized as follows:

- The learning process is no central unit;
- The weights are adjusted locally on current variables (signal, position);
- Weights do not depend on the difference between the desired output and the actual value of the system;
- The transition process runs until the network has reached an equilibrium. In our case, it is achieved when the RSSI-estimated distance is smaller than a threshold value, the robot will not move and then RSSI are unchanged.

Let w_{ij} be the weight value that connects the output of the i^{th} context input with the j^{th} output, $W = w_{ij}$ the weight matrix and $Y = [y_1, ..., y_n]^T$ the output vector. W is symmetric (i.e. $w_{ij} = w_{ji}$) and $w_{ii} = 0$. We consider a discrete Hopfield network used as an auto-associative memory [6] for searching purpose. Based on the premises considered, the evaluation of the stability property of our system can be performed by considering the computational energy function E, which is defined in n-dimensional output space Y as:

$$\Delta E = E(x(t+1)) - E(x(t)) = -\frac{1}{2} Y^T W Y \tag{6}$$

that is:

$$E = -\frac{1}{2}\sum_i \sum_j w_{ji} x_j(t+1) - \sum_i w_{0j}(t) x_i(t+1) + \frac{1}{2}\sum_i \sum_j w_{ji} x_j(t) + \sum_i w_{0j}(t-1) x_i(t) \tag{7}$$

where w_{0j} is as defined in Eq. 3. In the theory of stability, if the structure of the matrix is as those defined for our weights matrix W, and the schedule, where only a unit of the network is updated at a time, namely *asynchronous* update, it is possible to show that the system converges to one stable state in finite time. The stability is proved by showing that the energy function always decreases as the state of the processing are changed one by one. Let us consider that the neuron input (context input), that just changes state at step t is neuron p. Therefore, $x_p(t+1)$ is determined as:

$$x_p(t+1) = \begin{cases} 1 & \text{if } s(t) + noise > 0, \\ x_p(t) & \text{if } s(t) + noise = 0, \\ -1 & \text{if } s(t) + noise < 0. \end{cases} \tag{8}$$

It is worth recalling that X is the input pattern and Y is desired output pattern and in the case of auto-associative memory, we have $X = Y$, and the diagonal entries of the weights matrix W are set to 0, namely $w_{ii} = 0$, with $i = 1...n$. If all the states of the network are to be updated at once (as in our case), then the next state of the system will be represented as: $x(t+1) = y(W^T x(t))$. When the exemplars are orthonormal and we have: $y(x) = x$ and $YX^T x^r = \sum_i \delta_{ir} y^i = y^i$, with r=1...n and δ_{ir} is the *kronecker delta*, then we obtain: $y(W^T x^r) = y(x^r) = x^r$ that means that each stored

pattern (or memory element) is a stable state of the network. Whether $x_p(t+1)$ is determined as 8, for all the other inputs (or context inputs), we have $x_i(t+1) = x_i(t)$ for $i \neq p$. Furthermore, we have $w_{pp} = 0$, and:

$$\Delta E = -((x_p(t+1) - x_p(t))(\sum_j w_{jp} x_j(t)) + w_{0p}(t)) \tag{9}$$

namely,

$$\Delta E = -((x_p(t+1) - x_p(t)) s_p(t) \tag{10}$$

It is worth noticing that, if the value of x_p remains the same, then $x_p(t+1) = x_p(t)$ and the $\Delta E = 0$. If they are not the same, either it will be $x_p(t) = -1$ and $x_p(t+1) = 1$ due to the fact that $s_p(t) > 0$, or $x_p(t) = 1$ and $x_p(t+1) = -1$ due to the fact that $s_p(t) < 0$. Whatever the case is, if $x_p(t+1) \neq x_p(t)$ it is in a direction for which $\Delta E < 0$. Therefore, for this type of specific network (discrete Hopfield), we have $\Delta E < 0$. Since the energy function decreases at each state (some fixed amount) and it is bounded, it reaches a minimum value in a finite number of state changes. This can be translated as a convergence to one stable state of the network in finite time. The type of schedule considered here is named *asynchronous* update, since only one unit at each time is updated. Where all the units are updated at once, namely the *synchronous* update, the convergence is not guaranteed, since it may result in a cycle of length two. Of course, some of the stored patterns may not be a stable state. In fact, we are dealing experiments with RSSI in indoor environments, and we experimented cases of components of multi-path that add in some points, resulting in higher values of the target signal. We also faced with some spurious stable states, that is different from the stored patterns. Based on the details we have given in the implementat Section, whether the initial state is set to one of the exemplar (the ASN implementation output says that the robot already reached the target, based on the input signals received), the robot remains there (it does not move). On the other hand, if the initial state is set to some arbitrary input, then the network converges to one of the stored memory elements, depending on the basin of attraction in which $x(0)$ lies.

8 Conclusions

In this work we implemented an Associative Search Network on Arduino-based robots to perform target indoor localization tasks. As input signals, we considered the RSSI parameters for the inherent advantages that it has, such as no extra-hardware required and it is always available. These features can play a very important role when heterogeneous devices are considered and are asked to accomplish the target localization task. The realized ASN is Hopfield-network based, and this allowed us to characterize our system with some main and important features regarding stable states. Even if there are open issues related to the presence of spurious stable states (i.e. optimal states that are different form those stored), we showed that the system is effective and allows the robots to reach the target object. Moreover, the learning technique as implemented is not constrained neither with a specific number of landmarks nor with a specific position in the area, since the system evolves and learns by acquiring the external signals.

References

1. Vempaty, A., Han, Y.S., Varshney, P.K.: Target Localization in Wireless Sensor Networks using Error Correcting Codes. IEEE Trans. Inf. Theory 60(1), 697–712 (2014)

2. Barto, A.G., Sutton, R.S., Brouwer, P.S.: Associative Search Network: A Reinforcement Learning Associative Memory. Biol. Cybern. 40, 201–211 (1981)
3. Shaw, J.A.: Radiometry and the Friis transmission equation. Am. J. Phys. 33(81) (2013)
4. Minsky, M.L., Papert, S.: Perceptron: an introduction to computational geometry. MIT Press, Cambridge (1969)
5. Hyytiä, E., Virtamo, J.: Random waypoint model in n-dimensional space. Operations Research Letters (2005)
6. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Acad. Sci. 79, 2554–2558 (1982)
7. Priwgharm, R., Srivilas, K., Cherntanomwong, P.: Indoor localization system using RSSI measurement in wireless sensor network based on ZigBee standard. In: NCIT (2010)
8. Deshpande, N., Grant, E., Henderson, T.C.: Target localization in unknown environments using static wireless sensors and mobile robots. In: IEEE MFI, Salt Lake City (2010)
9. Bota, M., Guazzelli, A.: "The Associative Search Network: Landmark Learning and Hill Climbing. The Neural Simulation Language A System for Brain Modeling. MIT Press, Cambridge (2002)
10. Heurtefeux, K., Valois, F.: Is RSSI a good choice for localization in wireless sensor network? In: IEEE AINA, Fukuoka, Japan (2012)
11. Zhang, X., Li, S., Lin, Z., Wang, H.: Range based target localization using a single mobile robot or multiple cooperative mobile robots. In: Proc. ICCA, June 2013
12. Dames, P., Kumar, V.: Cooperative multi-target localization with noisy sensors. In: IEEE ICRA (2013)
13. Zhou, N., Zhao, X., Tan, M.: RSSI-based mobile robot navigation in grid-pattern wireless sensor network. In: CAC (2013)
14. Gang, L., Wei, Z., Shiyi, M., Shuqin, S.: Sensor selection for target tracking in a wireless sensor network of bearing-only sensor. Inter. J. of Digital Content Technology and its Applications 6-1, 41–48 (2012)
15. Hamdoun, S., Rachedi, A., Benslimane, A.: RSSI-based localization algorithms using spatial diversity in wireless sensor networks. In: Ad Hoc and Ubiquitous Computing (IJAHUC), January 2014
16. Zanca, G., Zorzi, F., Zanella, A., Zorzi, M.: Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks. In: REALWSN, Glasgow, UK (2008)
17. Loscrí, V.: Performance evaluation of novel distributed coverage techniques for swarms of flying robots. In: IEEE WCNC, Istanbul, Turkey (2014)
18. Tsetlin, M.L.: Automaton Theory and modeling of biological systems. Academic Press, New York (1973)
19. Narendra, K.S., Thatachar, M.A.L.: Learning Automaton - A Survey. IEEE Trans. Syst. Man Cybern. 4, 323–334 (1972)