Robust and Efficient Self-Adaptive Position Tracking in Wireless Embedded Systems

Ramil Agliamzanov*, Assia Belbachir† and Kasım Sinan Yıldırım‡
*Higher Institute for Information Technology and Information Systems, Kazan Federal University, Kazan, Russia
†IPSA, 7-9 Rue Maurice Grandcoing Ivry-sur-Seine, 94200 France
‡Department of Computer Engineering, Ege University, İzmir, Turkey

***Abstract*—Apart from static deployments, sensor nodes in Wireless Sensor Networks (WSNs) are unaware of their location information. In order to estimate their actual or relative positions with respect to other nodes, they are required to self-localize themselves by collecting information from their environment. However, due to the high dynamism and the noise introduced by the WSN environment, self-localization procedures are not straightforward and they may require quite sophisticated algorithmic techniques to satisfy precision requirements of the WSN applications. Among the self-localization procedures in the literature, the ones based upon the technique of trilateration are easy to implement and efficient in terms of resource requirements. On the other hand, their performance is fragile against environmental dynamics. Besides, even though multilateration based procedures are reported to be more robust, their practicability in WSNs seems questionable due to their high resource requirements.**

In this paper, our objective is to develop a practical self-localization procedure for WSNs that puts away the fragility against noisy ranging measurements in an efficient manner. To that end, we take a different approach to self-localization procedure and treat it as a search process during which sensor nodes find their relative positions without knowing the actual correct values. We present a novel trilateration-based self-localization procedure by exploiting a robust and efficient search technique, named Adaptive Value Tracking (AVT), that finds and tracks a dynamic searched value in a given search space through successive feedbacks. We evaluate this procedure on a real testbed setup and show that our approach to self-localization is efficient, robust to environmental dynamics and adaptive in the sense of reacting to position changes.

I. Introduction

Localization is one of the fundamental middleware services in Wireless Sensor Networks (WSNs). There are applications in WSNs, such as forest-fire detection, that demand the location of the sensed environmental phenomenon. However, apart from specific static deployments, sensor nodes are unaware of their location information. GPS receivers are currently not suitable for sensor nodes due to their high cost, power consumption and complex hardware structure. Therefore, sensor nodes are required to localize themselves (self-localization) by collecting information from their environment in order to estimate their actual and/or relative positions with respect to other nodes. The objective here is to minimize their estimation errors, which is not straightforward to achieve and requires coping with several aspects related to the high dynamism and the noise introduced by the WSN environment.

The self-localization procedures in the literature can be classified into two categories: *range-free* and *range-based*. The *range-free* procedures are built on distance estimates that are calculated by using communication range and connectivity information of the nodes [1]. Explicit point-to-point distance measurements that may depend on additional hardware capabilities are not required. Although the prominent property of these procedures is their simplicity and low-costness, their accuracy is quite low. On the other hand, the *range-based* procedures demand point-to-point distance measurements by using ranging techniques such as Time Difference of Arrival (TDoA) of the ultrasound signals [2]. They are known to exhibit high accuracy, but introduce extra overhead since they may require additional hardware capabilities.

Our focus in this paper is range-based self-localization of a single node u that is in the communication range of $n \geq 3$ anchor nodes whose positions (x_i, y_i) are known. We conceptualize this single-hop procedure as comprising two step process, as suggested in [3]: at the first step, node u communicates with the anchor nodes and obtains its relative distance measurements by using any ranging method and the position information of the anchor nodes. Mathematically, the collected information can be written as a linear matrix equation $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is a (n-1)x2 matrix whose entries are defined as $a_{i,1} = x_n - x_i$ and $a_{i,2} = y_n - y_i$, $\mathbf{x} = [x_u \quad y_u]^T$ is a vector that holds the unknown coordinates of the node u, and $\mathbf{b}$ is a vector whose entries are defined as

$$b_{ij} = (d_i^2 - d_n^2) - (x_i^2 - x_n^2) - (y_i^2 - y_n^2) \qquad (1)$$

where d_i denotes the measured distance between node u and the anchor node i. In the second step of the self-localization procedure, node u applies a localization algorithm and estimates its potential position. In the simplest case, the position estimate can be obtained by solving the linear matrix equation for $n = 3$, that turns into a *trilateration* process. However, if the distance measurements are noisy, trilateration is not sufficient to obtain an accurate position estimate. One solution to this problem is to incorporate the distance measurements from more than three anchor nodes if possible and perform maximum likelihood *multilateration* [4], [5]. In other words, with the vector $\mathbf{x}$ that minimizes the mean square error, i.e. minimizing the expression $\|\mathbf{Ax} - \mathbf{b}\|_2^2$ one gets an improved estimated position compared to trilateration. However, such techniques have serious drawbacks. First, they increase the memory overhead since relative distance measurements and position information of the all anchors should be stored.

Second, they require computationally expensive matrix operations such as matrix transpose and multiplication. Thus, the practicability of these techniques in WSNs seems questionable.

In common application scenarios, trilateration based self-localization procedures are preferable not only because of their ease of implementability and efficiency but also it may not be possible to obtain information from more than three anchor nodes simultaneously. However, as aforesaid, their performance is fragile against environmental dynamics, which is our point of interest in this paper. The objective of this study is to develop a practical self-localization procedure for WSNs that puts away the fragility against noisy ranging measurements in an efficient manner. To that end, we take a different approach to self-localization procedure: for the first time, we treat the whole self-localization steps, i.e. the ranging step and localization algorithm that estimates the potential position, as a *search process* in which the relative distances to the anchor nodes and the position of the node are trying to be found without knowing their correct value*s*. With this perspective, we develop a novel trilateration-based self-localization procedure by exploiting a robust and efficient search technique, named Adaptive Value Tracking (AVT), that finds and tracks a *dynamic* searched value in a given search space through successive feedbacks. The real-world experiments in this study proved that our approach to self-localization is efficient, robust to environmental dynamics and adaptive in the sense of reacting to position changes.

The remainder of this paper is organized as follows. In Section II, we briefly introduce the method of adaptive value tracking. Then, in Section III we propose an efficient and robust self-adaptive position tracking procedure to self-localize any mobile node by collecting noisy information from its environment. In the next section, we evaluate the performance of the proposed approach considering the data collected from the experiments that were run on a real hardware platform. Finally, we conclude in Section V and propose future work.

II. Robust and Efficient Dynamic Search with Adaptive Value Trackers

In this section, we summarize the technique of Adaptive Value Tracking that provides the robustness and adaptivity of our self-localization system. In this technique, an Adaptive Value Tracker (AVT) finds and tracks a *dynamic* searched value, i.e. a searched value that may change in the time due to the dynamics of the system, in a given search space as fast as possible [6]. Up until now, this technique has been successfully used in several scientific and industrial projects [7], [8], [9], [10], [11], [12], [13]. Moreover, it has recently been incorporated in order to achieve self-organizing and fully distributed time synchronization in Wireless Sensor Networks [14], [6], [15]. In the light of these studies, we handle the problem of localization as a dynamic search process and develop a self-localization system based upon AVTs in the following sections.

From the perspective of software engineering, an AVT is a software component that implements a compact and efficient search algorithm in order to search and track a dynamic value $v^\star$. The prerequirement of the search algorithm is the information about the search space

$$[v_{min}, v_{max}] \quad \subset \quad \mathbb{R} \tag{2}$$

where v_{min} is the lower boundary and v_{max} is the upper boundary for the searched value $v^\star$. At any time t, the *owner component* can interact with the AVT component in order to obtain the current proposed value v_t and inform the search the direction that *probably* lead the proposed value v_t to converge to the searched value $v^\star$. More specifically, the owner component should determine if the searched value $v^\star$ is smaller than, equal to or greater than the current proposed value v_t, without knowing the actual value $v^\star$. This decision is not trivial and is made by taking into account the goal of the *owner component*. After this determination, the owner component informs the AVT component about increasing, decreasing or preserving the proposed value via sending a feedback. Hence, the tracking of a dynamic value is established via the successive feedbacks coming from the *owner component*. After receiving the feedback, AVT component may increase or decrease its proposed value by Δ_t according to the received feedback. Here,

$$\Delta_t \quad \in \quad [\Delta_{min}, \Delta_{max}] \tag{3}$$

is called the *adjustment step* at time t where Δ_{min} is the lower boundary and Δ_{max} is the upper boundary for Δ_t. Δ_{min} represents the minimum adjustment step that avt can use and it is also called *precision* since avt can only guarantee that it will approximate the $v^\star$ value within a margin of $\pm\Delta_{min}$: $\Delta_{min} \geq |v_t - v^\star|$. On the other hand, Δ_{max} represents the maximum adjustment step that avt can use, i.e. it is the *maximum evolution speed* of v_t. We refer the reader to [6] for the detailed explanation and analysis of the technique of AVT.

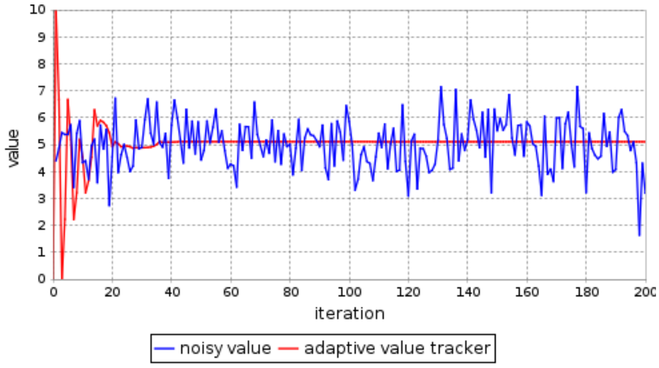

Fig. 1. A simulation result related to the behavior of AVT under noisy information. During simulations we set $v^\star = 5$ and introduced a noise to this value which is distributed as a zero-mean Gaussian random variable with variance 1. For AVT parameters, we used $[v_{min}, v_{max}] = [0, 10]$ and $[\Delta_{min}, \Delta_{max}] = [0.0001, 1]$. At each iteration, the error between the noisy value and the proposed value v_t of the AVT is calculated. Considering the error, a feedback about increasing, decreasing or good is sent to the AVT. As can be observed, the AVT has converged after finite amount of time and it is quite robust to the noisy values as compared to the actual noisy information. After the convergence, the values proposed by the AVT were quite stable.

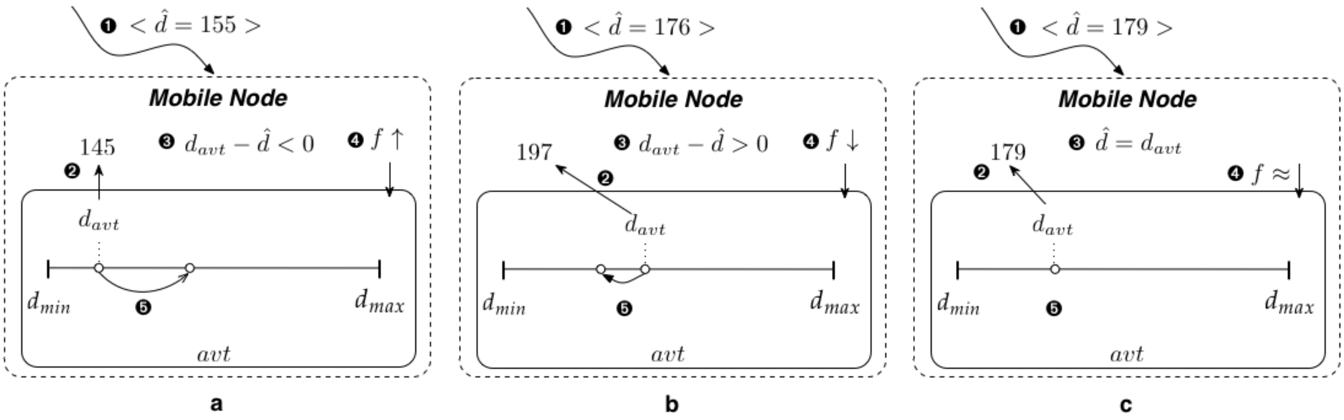

Fig. 2. The successive reactions of the mobile node (a, b and c respectively) upon obtaining successive distance measurements from its anchor nodes. When a new measurement $\hat{d}$ is obtained, e.g. via using ultrasound ranging (a.1), the distance value d_{avt} of the mobile node is retrieved from its AVT (a.2). Then by comparing the distance values $\hat{d}$ and d_{avt}, the mobile node observes that the error is smaller than the zero (a.3) and thus sends an increase feedback $f \uparrow$ to its adaptive value tracker avt (a.4). Upon receiving $f \uparrow$, avt increases its value (a.5). When a second distance measurement is obtained, the same steps are executed. However this time since the error is greater than zero (resp. the error is within tolerance bounds), a decrease feedback $f \downarrow$ (resp. a good feedback $f \approx$) is sent to d_{avt} and d_{avt} decreases its value (resp. d_{avt} does not change its value).

An important issue in practice, as discussed in [14], is the *robustness* and *adaptivity* of the AVTs. In other words, a good self-localization procedure should be minimally affected by the noise introduced from the WSN environment. Since AVTs are the core components of our self-localization system, their behaviors under environmental dynamics effect the whole system behavior. In this sense, we performed simulations in order to observe the behavior of AVTs under noisy information. During this simulations, we introduced a Gaussian noise for a fixed search value. At each iteration, the error between the noisy value and the proposed value from AVT is calculated. Considering this error, a feedback is sent to the AVT. From Figure (1) that presents our preliminary results, we claim that AVTs improve robustness against WSN dynamics (see also Figure 4 in [14]). However, it should also be noted that successively receiving erroneous feedbacks will increase the adjustment step exponentially and will make AVT further from the correct rate value. That's why the goal of owners have to be well-defined and correct.

III. An Efficient and Robust Self-Adaptive Localization Procedure

In this section, we take a different approach to self-localization and treat it as a *search process* that finds the actual position of a mobile node in a dynamic WSN environment where the measurements and collected information are noisy. We propose an efficient and robust self-adaptive position tracking approach whose objective is to self-localize any mobile node communicating three anchor nodes. The proposed approach handles single-hop self-localization as a two step *search* procedure. At the first step, the relative distances with respect to the anchors are estimated by employing an adaptive search process during which each relative distance is searched and found by using an AVT *without knowing its correct value*. In the second step of the algorithm, the relative distances coming from the first step and the (x,y) coordinates obtained from the anchors are used to estimate the actual position of the mobile node. Similar to the first step, this estimation is performed by employing an adaptive search process during which the actual (x,y) coordinates of the mobile node are searched and found by using two AVTs *without knowing their correct value*s.

A. A Robust Ranging Algorithm

As we stated previously, the first step of the self-localization procedure is ranging. In this step, the node communicates with the anchor nodes and estimates its relative distances by using a particular ranging method. However, due to network dynamics, the quality of its estimations are affected by the measurement noise. In order to get more robust estimates, we propose to handle the ranging as search process during which the relative distance d^* is searched inside the search space $[d_{min}, d_{max}]$ using an AVT, where d_{min} and d_{max} are the minimum and maximum distances that can be measured physically by the ranging method.

Algorithm 1 The main steps of the algorithm that estimates the range d between a mobile node and a specific anchor node.

1: Obtain a new range estimate $\hat{d}$ using any ranging method
2: $error = avt.getValue() - \hat{d}$
3: **if** $error < 0$ **then** $avt.adjust(f \uparrow)$
4: **else if** $error > 0$ **then** $avt.adjust(f \downarrow)$
5: **else** $avt.adjust(f \approx)$

The ranging steps between a mobile node and a particular anchor node are summarized in Algorithm (1). If the distance value $\hat{d}$, which is estimated by using the available hardware e.g., ultrasonic transmitters and receivers, is higher than the range value proposed by the avt used for tracking the relative distance of the mobile node, an increase feedback $f \uparrow$ for increasing the distance, if it is smaller then a decrease feedback $f \downarrow$ for decreasing the distance, otherwise a good feedback $f \approx$ for indicating that the current range value is good is sent. Employing this algorithm, the value proposed by the avt of the mobile node will converge to the actual distance value in finite amount of time. Figure (2) presents a sample execution of the Algorithm (1).

B. Adaptive Position Tracking Algorithm

The second step of the self-localization procedure is the localization algorithm that estimates the relative position of the mobile node. This estimation is based on trilateration, hence position and ranging data from three anchors are required. Thus, we assume that the mobile node is within the range of three anchor nodes and it applied Algorithm (1) to obtain its relative distance estimates to the anchor nodes. Therefore, the mobile node requires three AVTs in order to track these relative distances. Moreover, we also assume that the position information of all three anchor nodes are obtained via communication.

Algorithm 2 The main steps of the algorithm that estimates the (x, y) position of the mobile node.

Calculate $(\hat{x}, \hat{y})$ by trilateration
$x_{err} = avt_x.getValue - \hat{x}$
$y_{err} = avt_y.getValue - \hat{y}$
if $x_{err} > 0$ **then** $avt_x.adjust(f \uparrow)$
else if $x_{err} < 0$ **then** $avt_x.adjust(f \downarrow)$
else $avt_x.adjust(f \approx)$
if $y_{err} > 0$ **then** $avt_y.adjust(f \uparrow)$
else if $y_{err} < 0$ **then** $avt_y.adjust(f \downarrow)$
else $avt_y.adjust(f \approx)$

The localization steps of the mobile node is summarized in Algorithm (2). Having estimated its relative distances and obtained the positions of the anchor nodes, a trilateration is sufficient to estimate the (x, y) coordinate of the mobile node. However, the estimation error of the ranging step effects the quality of this estimation. In order to have a robust and stable estimation, we propose to estimate these coordinates as a search process during which the actual coordinate (x^*, y^*) is searched within the search spaces $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ respectively by using two AVTs. At any time, the avt_x and avt_y can propose the (x, y) coordinates. The error between the estimated $\hat{x}$ and $\hat{y}$ values calculated by the trilateration and the values proposed by avt_x and avt_y are calculated to inform the AVTs about the current feedback. With these steps, the value proposed by avt_x and avt_y of the mobile node will converge to the actual position in finite amount of time.

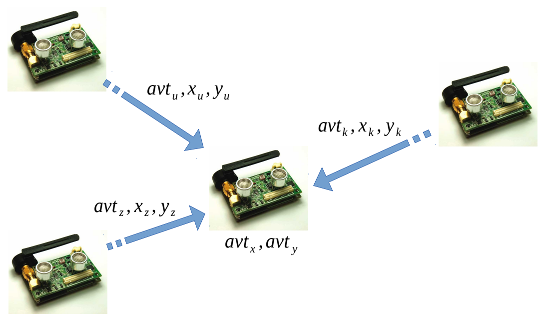

Fig. 3. Self-localization scenario between anchor nodes and mobile node.

As we mentioned before, we assumed that the mobile node applied Algorithm (1) to track its relative distance to the anchor nodes. However, it is worth to underline that Algorithm (2) can also be applied to the raw distance measurements, i.e. the distance measurements that are not tracked with AVTs. This could also be very relevant since the requirement of three distance tracking AVTs is eliminated, which reduces CPU and memory overhead but also decreasing robustness.

Figure (3) visualizes and summarizes the whole self-localization procedure with Algorithm (1) and (2). The mobile node keeps track of its relative distances to the anchor nodes u, k and z by storing three adaptive value trackers avt_u, avt_k and avt_z, respectively. By employing any ranging method, these trackers are informed by sending feedbacks to force them to converge to the actual relative distance values. Meanwhile, the mobile node also collects the location information (x_u, y_u), (x_k, y_k) and (x_z, y_z), respectively. Using collected coordinate values and the proposed distance values from the corresponding AVTs, a trilateration is sufficient to estimate the $(\hat{x}, \hat{y})$ coordinates of the mobile node. Using these estimates, mobile node informs avt_x that keeps track of its x coordinate and avt_y that keeps track of its y coordinate by sending feedbacks.

IV. Experimental Work

In order to evaluate the actual performance of the proposed approach, we implemented it in TinyOS [16] and performed real-world experiments in our testbed. The hardware platform used for the implementation was Sensenode sensor platform from Genetlab [17]. Sensenode is made up of two components, Node-RF 1.3b processing and communication module and Sense-L 1.3 sensing module, respectively. The MCU used in this platform is 16-bit low-power MSP430 microcontroller which has 48 kB program flash, 10 kB RAM and 1024 kB external flash memory. The transceiver is Chipcon CC2420 radio chip which provides a 250 kb/s data rate at 2.4 GHz frequency and compliant with IEEE 802.15.4. Sensing module is equipped with Kobitone Ultrasonic Transducers 255-400ST12-ROX and 255-400SR12-ROX, generating and receiving ultrasound at 40 kHZ frequency with a maximum range of 10 meters.

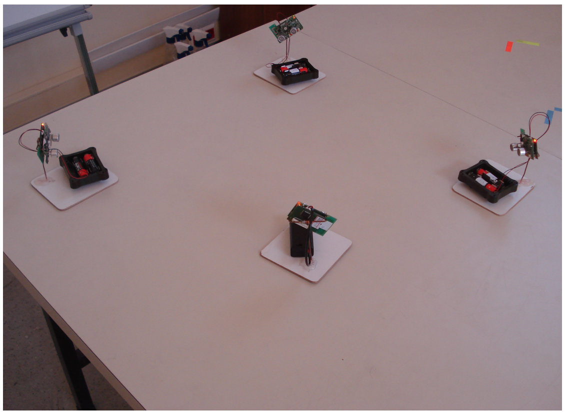

Fig. 4. The testbed setup during the experiments. Three anchor nodes were placed at fixed positions (0,0), (50,0) and (0,50), respectively. One mobile node that is unaware of its location information was within the line-of-sight of these anchors.

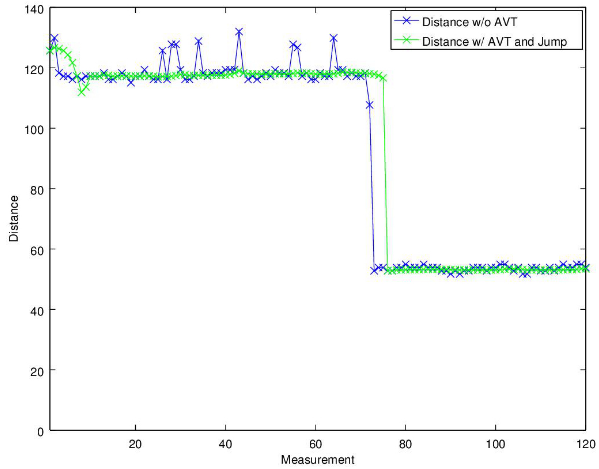

Fig. 5. The relative distance estimations of a mobile node with respect to an anchor node. Line with green color represents estimations with Algorithm (1) and line with blue color represents estimations based on raw measurements. The initial distance between the mobile node and the anchor node was 100 cm. Then, the mobile node moved closer to the anchor node to make the relative distance 50 cm.

We performed our experiments on a testbed of five Sensenode sensor nodes that were placed on a 100 cm x 100 cm area, as presented in Figure (4). In this testbed, three of the nodes were functioning as anchor nodes that knew their actual positions. These nodes were programmed and placed at fixed reference positions with (x,y) coordinates (0,0), (50,0) and (0,50), respectively. Within line-of-sight with of these three anchor nodes, one mobile node that is unaware of its location information was communicating with the anchor nodes periodically in order to perform ranging and localization. Apart from these nodes, a base station node was listening the network and transferring the range and position estimates of the mobile node to the serial port of our PC. A java application listening the serial port logged the experimental data that were analyzed at the end of the experiments.

During the experiments, the mobile sensor node used three AVTs to keep track of its relative distance to the anchor nodes and two AVTs to keep track of its x and y coordinates. Since the deployment area is a 100 cm x 100 cm square, we defined the upper bound and lower bounds of the searched distance measurements for the AVTs that were used to keep track of the relative distances as $v^\star \in [v_{min}; v_{max}] = [0, 100\sqrt{2}]$. On the other hand, these parameters were $v^\star \in [v_{min}; v_{max}] = [0, 100]$ for the AVTs that were used to keep track of the x and y coordinates. We adjusted the upper and lower bounds for the adjustment step Δ_t as $\Delta_t \in [\Delta_{min}, \Delta_{max}] = [0.1, 50]$ since a precision of 0.1 cm is sufficient to obtain robust distance and position estimates. It should be noted that due to the low quality of our ultrasound transducers, our range measurements during the experiments exhibited errors in the order of 10 cm. Therefore, our search precision was sufficient.

A. Evaluation of the Range Estimation

In order to evaluate the ranging performance of the proposed approach, we considered a simple scenario between the anchor node that was placed at (0,0) and the mobile node which was initially placed at (100,100). We performed an experiment during which we collected range estimates from the mobile node for 5 minutes (one measurement at each 5 second), then we moved the mobile node to (50,50) and collected range estimates for 5 minutes again.

In our implementation, the ranging measurements were based on Time Difference of Arrival (TDoA) of ultrasonic signals, similar to the approach presented in [18]. This approach demands common notion of time among the members of the network. For this purpose, we used AVTS protocol [6] for time synchronization. After time synchronization is established, the ranging measurements were obtained using straightforward steps. In summary, the mobile node was transmitting ranging requests to the anchor node with a frequency of 5 seconds. Whenever a request is received from a mobile node, the anchor node transmits a RF signal that carries the transmission time t_1 and then it transmits an ultrasonic wave immediately. The mobile node receives t_1 from the RF message and saves the receipt time t_2 of the ultrasonic wave. Since the propagation speed of the ultrasonic signal v_{sound} is known, the estimated distance $\hat{d}$ can be calculated as $\hat{d} = (t_2 - t_1)v_{sound}$. However, these measurements are noisy due to environmental dynamics and the quality of the ultrasonic hardware. For instance, the speed of the sound v_s has large sensitivity to the temperature variations [18]. Moreover, the strength of the ultrasonic signal decreases with the distance between the sender and the receiver. Thus, the receiver side may detect the signal with a non-deterministic delay introducing large measurement variations.

Figure (5) presents the range estimations performed by the mobile node with and without Algorithm (1). It can be noticed that, even though the real distance between mobile and the anchor node does not change, raw ranging measurements are quite unstable and they suffer from environmental dynamics. On the other hand, with Algorithm (1), the range measurements are quite stable and they are robust against

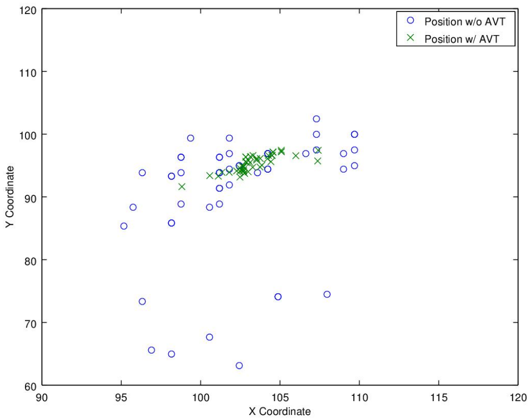

Fig. 6. The position estimations without (blue circles) and with (green crosses) Algorithm (2). It can be observed that the proposed coordinate estimates are more stable and correct with Algorithm (2).

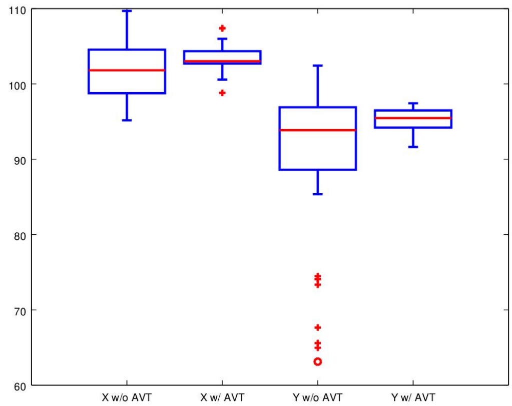

Fig. 7. The summary of position estimation accuracy. The error boundaries of the X coordinates (left) and those for the Y coordinates (right) are presented with trilateration-only and our self-adaptive procedure, respectively.

noisy measurements. Observe that, when the position of the mobile node was changed, Algorithm (1) converged in a quite reasonable time and suggested robust values compared to raw measurements. Moreover, when the mobile node is more closer to the anchor node, the error of the distance estimations became smaller, as we stated previously. It can be concluded from these experimental data that Algorithm (1) is robust to environmental dynamics and adaptive in the sense of reacting to range changes. Note that we already presented the resource efficiency of the proposed approach through extensive measurements on a real hardware platform in [6]. Hence, with a very little computation and memory overhead, one gets consistent ranging measurements that will be used by the second step of the self-localization procedure, which we evaluate in the next subsection.

B. Evaluation of the Position Estimation

In order to evaluate the localization performance, we involved all of the anchor nodes and considered a similar scenario to the previous experiments. During the experiments, the mobile node that was initially placed at (100,100) was collecting coordinates from the anchor nodes and obtaining range estimates simultaneously for approximately 5 minutes. Then the mobile node was moved to (50,50) and the former steps are performed for 5 minutes.

Figure 6 presents the position estimations when the mobile node was at (100,100). It can be observed from this figure that, the trilateration only solution suffers from poor performance due to noisy range estimations. The proposed position estimates are quite unstable and they exhibit large errors. On the other hand, Algorithm (2) proposed robust and quite stable position estimates compared to the former approach. Figure 7 presents the summary of the position estimation accuracy during the experiments. It can be concluded that handling localization problem as an adaptive search and employing the method of adaptive value tracking not only improved the ranging estimations but also position estimations. In addition, all these significant improvements came at a price of very little computation and memory overhead.

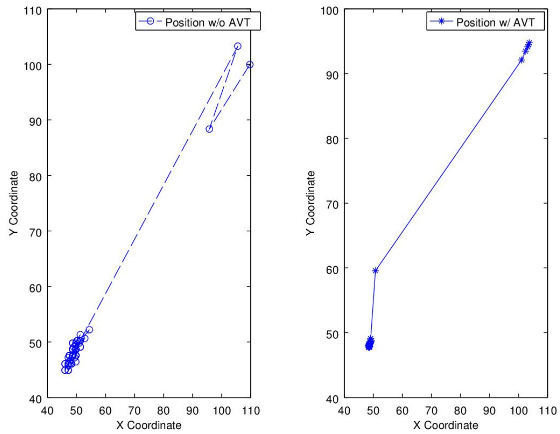

Fig. 8. Adaptivity of the position estimation with Algorithm (2) in terms of mobility.

Figure 8 presents the adaptivity of Algorithm (2) in terms of mobility when the mobile node was moved from (100,100) to (50,50). With trilateration only and raw range measurements, the mobile node reacted to the position change immediately, but its position estimates was unstable. However, mobile node with Algorithm (2) did not react to the position change immediately at first. This is due to the fact that AVT is required to receive successive feedbacks of the same direction in order to increase its delta value and converge to the desired position. Once the convergence is established, the proposed position estimates are quite stable. It can be concluded that Algorithm (2) is adaptive in terms of mobility and reacts to the position changes in a quite reasonable amount of time.

C. Summary of the Experimental Work

In the light of our experiments, we had the following observations and conclusions:

- The error of range measurements with our ranging procedure was between 2-5 cm while that with pure trilateration was 10-15 cm.
- The error of position estimates with our self-localization procedure was between 2-5 cm while that with pure trilateration was 10-15 cm.
- Our self-localization system was robust against noisy ultrasonic range measurements and its estimates were stable as compared to pure trilateration.
- Our self-localization procedure is lightweight in terms of processing and memory requirements that fits perfectly to the requirements of WSN nodes.
- Our system adapts itself quite fast thanks to the feedbacks sent to the AVTs.

V. Conclusion

In this study, self-localization has been considered as a dynamic search procedure for the first time in the literature. Following this consideration, we developed a practical trilateration based self-localization procedure for WSNs that is robust against noisy ranging measurements, adaptive to network dynamics and efficient in terms of resource requirements. The efficiency, adaptivity and robustness of this procedure have been gained by treating the whole self-localization as a search process and exploiting a robust and efficient search technique named Adaptive Value Tracking. We presented an evaluation of this approach on a small testbed and presented experimental results. Based on this evaluation, we observed that the estimation error of our self-localization system is quite low as compared to the pure trilateration with drastically lower CPU overhead and considerably smaller code size.

An interesting future study may include the multi-hop extension of our self-localization procedure and its evaluation on a real testbed. In addition to the study of multi-hop, the approach presented in this paper can be used for the localization of drones. Drones are Unmanned Aerial Vehicles (UAVs) that autonomously follow a pre-programmed mission. However, making a fully autonomous drone able to hover accurately near an object of interest, to explore its environment with or without GPS, to navigate in a cluttered environment without map and to avoid obstacles in a dynamic environment is still a great challenge for drone manufacturers and scientists working on aerial robots. To this end, localization is a big concern. A critical situation occurs when a drone temporarily loses its GPS position information, which might lead it to crash. This can happen, for instance, when flying close to buildings where GPS signal is lost and/or when navigation through some specific areas (such as forests and tunnels), where GPS coverage is vulnerable. In such situations, it is desirable that the drone can rely on fall-back systems and regain stable flight as soon as possible. Most of the developed fall-back systems uses heavyweight mechanisms to compensate the GPS failure, such as infrared and visual cameras [19] or laser range scanners [20]. Considering that drones have memory, energy and timing constraints, a desirable localization protocol should rather use lightweight mechanisms. Based on this observation, we can enhance the self-localization technique for drones where they only exchange their instant location information. This technique is both faster and consumes less energy, as shown in this paper.

VI. Acknowledgments

The authors would to thank very much Önder Gürcan for his valuable comments that significantly improved this study.

References

[1] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM, 2003, pp. 81–95.

[2] G. Oberholzer, P. Sommer, and R. Wattenhofer, “The spiderbat ultrasound positioning system,” in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 403–404.

[3] Á. Lédeczi and M. Maróti, “Wireless sensor node localization,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 85–99, 2012.

[4] M. DeGroot and M. Schervish, *Probability and Statistics, third ed.,*. Addison-Wesley, 2002.

[5] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, 2010.

[6] K. Yildirim and O. Gürcan, “Efficient time synchronization in a wireless sensor network by adaptive value tracking,” *Wireless Communications, IEEE Transactions on*, vol. 13, no. 7, pp. 3650–3664, July 2014.

[7] N. Brax, E. Andonoff, M. Gleizes, and P. Glize, “Self-adapted aided decision-making: Application to maritime surveillance,” in *5th International Conference on Agents and Artificial Intelligence*, 2013.

[8] ——, “Mas4at : un sma auto-adaptatif pour le déclenchement d'alertes dans le cadre de la surveillance maritime.” *Revue d'Intelligence Artificielle (forthcoming)*, 2013.

[9] L. Pons, C. Bernon, and P. Glize, “Scenario Control for (Serious) Games using Self-organizing Multi-Agent Systems (regular paper),” in *International Conference on Complex Systems (ICCS), Agadir, Morocco, 05/11/2012-06/11/2012*. http://ieeexplore.ieee.org/: IEEExplore digital library, 2013, p. (electronic medium).

[10] O. Gürcan, C. Bernon, K. S. Türker, J.-P. Mano, P. Glize, and O. Dikenelli, “Simulating human single motor units using self-organizing agents,” in *Self-Adaptive and Self-Organizing Systems (SASO), 2012 IEEE Sixth International Conference on*, sept. 2012, pp. 11 –20.

[11] Ö. Gürcan, K. S. Türker, J.-P. Mano, C. Bernon, O. Dikenelli, and P. Glize, “Mimicking human neuronal pathways in silico: an emergent model on the effective connectivity,” *Journal of Computational Neuroscience*, pp. 1–23, 2013, 2013/07/04/online. [Online]. Available: http://dx.doi.org/10.1007/s10827-013-0467-3

[12] E. Kaddoum and J.-P. Georgé, “Collective Self-Tuning for Complex Product Design (short paper),” in *IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Lyon - France*. CPS (Conference Publishing Services), september 2012, p. (electronic medium).

[13] S. Lemouzy, V. Camps, and P. Glize, “Real time learning of behaviour features for personalised interest assessment,” in *Advances in Practical Applications of Agents and Multiagent Systems*, ser. Advances in Soft Computing, Y. Demazeau, F. Dignum, J. Corchado, and J. Pérez, Eds. Springer Berlin / Heidelberg, 2010, vol. 70, pp. 5–14.

[14] Ö. Gürcan and K. S. Yildirim, “Self-organizing time synchronization of wireless sensor networks with adaptive value trackers,” in *Self-Adaptive and Self-Organizing Systems (SASO), 2013 IEEE Sixth Int. Conf. on*, sept. 2013, pp. 91 –100.

[15] K. S. Yildirim and Ö. Gürcan, “Adaptive synchronization of robotic sensor networks,” in *First International Workshop on Robotic Sensor Networks part of Cyber-Physical Systems Week, Berlin, Germany*, April 2014.

[16] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, "Tinyos: An operating system for sensor networks," in *in Ambient Intelligence*. Springer Verlag, 2004.

[17] Genetlab, "Sensenode technical specifications document v1.3b," 2008.

[18] N. B. Priyantha, "The cricket indoor location system," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[19] L. Merino, F. C. and. J.R. Martinez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple uavs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, pp. 165–184, 2006.

[20] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in gps-denied environments using onboard sensing," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2012)*, St Paul, MN, 2012.