
Article:
Lagkas, T. orcid.org/0000-0002-0749-9794, Eleftherakis, G. orcid.org/0000-0003-4857-4006, Dimopoulos, K. et al. (1 more author) (2020) Signal strength based scheme for following mobile IoT devices in dynamic environments. Pervasive and Mobile Computing, 65. 101165. ISSN 1574-1192

https://doi.org/10.1016/j.pmcj.2020.101165

Signal Strength based Scheme for Following Mobile IoT Devices in Dynamic Environments

Thomas Lagkas[a,b], George Eleftherakis[a], Konstantinos Dimopoulos[a], Jie Zhang[c]

[a] *Computer Science Department, The University of Sheffield International Faculty - CITY College, Thessaloniki, Greece*
[b] *Department of Computer Science, International Hellenic University, Kavala Campus, Greece*
[c] *Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, UK*

Abstract

The increased maturity level of technological achievements towards the realization of the Internet of Things (IoT) vision allowed sophisticated solutions to emerge, offering reliable monitoring in highly dynamic environments that lack well-defined and well-designed infrastructures. In this paper, we use a bio-inspired IoT architecture, which allows flexible creation and discovery of sensor-based services offering self-organization and self-optimization properties to the dynamic network, in order to make the required monitoring information available. The main contribution of the paper is the introduction of a new algorithm for following mobile monitored targets/individuals in the context of an IoT system, especially a dynamic one as the aforementioned. The devised technique, called Hot-Cold, is able to ensure proximity maintenance by the tracking robotic device solely based on the strength of the RF signal broadcasted by the target to communicate its sensors' data. Complete geometrical, numerical, simulation, and convergence analyses of the proposed technique are thoroughly presented, along with a detailed simulation-based evaluation that reveals the higher following accuracy of Hot-Cold compared to the popular con-

Email addresses: t.Lagkas@sheffield.ac.uk (Thomas Lagkas), g.eleftherakis@sheffield.ac.uk (George Eleftherakis), k.dimopoulous@sheffield.ac.uk (Konstantinos Dimopoulos), jie.zhang@sheffield.ac.uk (Jie Zhang)

T. Lagkas, G. Eleftherakis, K. Dimopoulos, and J. Zhang, "Signal strength based scheme for following mobile IoT devices in dynamic environments," *Pervasive and Mobile Computing, Elsevier*, 2020.

https://doi.org/10.1016/j.pmcj.2020.101165

cept of trilateration-based tracking. Finally, a prototype of the full architecture was implemented not only to demonstrate the applicability of the presented approach for monitoring in dynamic environments, but also the operability of the introduced tracking technique.

Keywords: GPS-denied environments, infrastructureless localization, IoT architectures, mobile tracking.

---

1. Introduction

It is a fact that the world frequently witnesses emergent situations which are related with major problems in infrastructures due to natural causes or the result of intentional or accidental human actions. Apart from their tremendous impact they all shared another characteristic: they reduced communication among humans. Modern means of communication were significantly degraded which rendered any rescue operations significantly harder. One of the main challenges in emergency management is to perform efficient monitoring and coordination. This aspect heavily relies on the communication between involved actors and the availability of information by monitoring victims' vital signs and also other environmental measurements. Traditional approaches in emergency scenarios lean on some sort of centralized or well and in advance engineered solution. However, as recent emergencies demonstrated, there is a need for communication mechanisms which do not rely on centralized systems and infrastructure. Such alternative solutions can be used as a fall-back mechanism in the case that primary systems fail.

At the same time, a remarkable maturity in recent technological advancements have led to the Internet of Things (IoT) as the most promising achievement towards smart solutions in a variety of applications. In this context, there is an inevitable need for scalable IoT architectures that offer advanced positioning, localization and context awareness based services for sophisticated applications enabling smart solutions (e.g., e-health, smart cities, smart emergency management, etc.). Such solutions need to be available even in extremely

dynamic and GPS-denied environments, allowing the deployment of flexible and autonomous sensor networks, especially in situations where well-defined infrastructures do not exist or are not preferable.

In this sense and in order to enable effective management and communication, there is a need for monitoring security and rescue forces personnel, victims, and other actors offering useful information in environments with no-well designed or crippled infrastructures. A common problem in such scenarios is also tracking down continuously a target that transmits useful information aiming to follow it maintaining proximity. At the same time, another critical point is the need of using resource limited components in such an attempt and preserve energy.

This work has been developed as part of an IoT architecture introduced in [1] that is capable of providing autonomous sensor-based distributed services. This architecture is based on bio-inspired principles found in natural systems to achieve the required robust behavior. In more detail, it is based on networked autonomous software agents which are serving information collected by interconnected IoT devices in weakly structured environments. The agents' behaviour, such as service discovery and self-organization, is adapted to the patterns of service consumers' (i.e. users) requests. The considered system is described in Section 3. However, it is noted that the scheme presented in this paper is architecturally agnostic, which means that it can be applied either to the aforementioned IoT architecture, or to any other IoT system, or as a standalone application.

The main contribution of this paper is the introduction of a new scheme that allows a robotic device to efficiently follow a monitored target solely based on the RF signal the latter broadcasts for communicating its sensors' data. The respective algorithm is called "Hot-Cold" and is thoroughly described, analyzed, and evaluated. A distinctive characteristic of the proposed solution which enhances robustness is the ability to maintain proximity without the necessity of identifying target's position. This tracking scheme is realized as a significant component of an original IoT system, which enables agile services creation and discovery

for the provision of flexible access to real-time monitoring information in dynamic environments. System feasibility and effectiveness is ensured through the implementation and testing of a complete prototype. The conducted evaluation shows the efficiency of the proposed technique in following a radio-emitting IoT device solely based on the strength of its communication signal, outperforming the well-known trilateration-based tracking in realistic shadowing conditions.

The rest of the paper is structured as follows. The next section presents background approaches in the field of location based services in IoT, emphasizing on related work on target tracking. Section III presents the adopted IoT system architecture, describing all main components. The proposed Hot-Cold algorithm is detailed and analyzed in Section IV. The following section documents the evaluation of the tracking scheme, discusses simulation results, and presents the system prototype. Finally, Section VI concludes the paper and provides insights for future work.

2. Background and Related Work

In this section we review the main concepts of this work and we focus on target tracking approaches.

2.1. Internet of Things and Location-based Services

Internet of Things (IoT) is a global collection of physical and virtual devices, and all related infrastructure, exchanging information and providing services to each other and to the people who use it [2]. Many services provided are Location Based Services (LBS). Depending on the prospective one can see it, LBS involves the use of location information of the target or the provider of the service [3]. Location Service (LS) is the process of defining the location of an asset, and is a crucial part of any LBS, and is usually done by GPS or other sensor embedded in the device. These sensors can provide a very accurate location, but are costly in power consumption. This can become an issue when non-accurate location information will suffice, but there are power consumption limitations.

With each device added to the IoT, global processing power is increased linearly, but communication channels between devices is following a much higher progression. Whatever the case, the communication technology is some form of wireless communication [2]. This means that information transmission between the involved devices takes place by default anyway, hence, the transmission itself can be exploited to provide an inexpensive means of LS.

2.2. Localization Techniques Overview

Localization is a topic that has predominated mobile robotics for a very long time. According to [4], localization is the problem of defining the spacial information (location, velocity and orientation) of a mobile robot in space.

By far the simplest technique is that of odometry. This involves measuring the movement of the robot (using rotational sensors on the wheels, or inertial sensors) to determine the change in position on regular intervals. However, as this approach is open-loop, it requires validation regularly as errors in measurements soon accumulate.

Another technique is based on using beacons. Beacon systems can be used to actively or passively determine the location of the robot, through triangulation or trilateration. The active or passive component is determined by whether the transmitting beacons are located on fixed known locations with the receiver on the robot or vice versa. Triangulation uses the estimated angles between the robot and the beacons to calculate accurately the location of the robot. Trilateration is using the estimated distances from the robot to the beacons to achieve the same goal. In both cases, complex mathematical equations have to be solved, a process that can become computationally heavy especially if repeated often.

Other, more advanced localization techniques require visual recognition of artificial landmarks, or visual tracking of the robot itself through cameras on the ceiling. Both techniques require image processing, thus vision equipment and visual line of sight, and therefore are outside the scope of this paper. For a full review of these techniques and more, see [5].

Finally, WiFi Positioning Systems (WPS) use geo-references radio maps of areas, to provide positioning information [6]. The accuracy of this approach, however, has a heavy cost as it is limited within a predefined area, and the process of building in advance the radio map is required [7].

2.3. Tracking Strategies Overview

Tracking involves knowing the location of a target, and navigating towards it. This could involve the avoidance of obstacles or path making. Control laws for tracking a target can be formulated with estimations of distance from the target [8, 9, 10, 11, 12, 13]. In [8], a tracking strategy is presented, where the distance from the target is estimated over time, by using the strength of a received signal. Consecutive distance estimations are then used to define a control law that defines the motion of the tracking agent to follow the target object. Another strategy that uses distance measurements between target and tracker can be based on the trilateration method, we discussed at the previous subsection. According to [9], by using this strategy it is possible for the tracker to locate a moving target in a bounded time, given that the target's speed is up to half the speed of the tracker. A similar approach is followed in [10], but there the problem is solved using orientation and distance information from the target, while in [12] and [13], the problem is solved without knowledge of the orientation of the target, but only the estimated distance from it and its derivative.

2.4. Target Tracking Techniques for Dynamic Networks

Recently, tracking of mobile devices in dynamic networks has attracted a lot of interest, due to the promising applications in various use cases. Such dynamic networks include indoor tracking scenarios as well as outdoor tracking in the context of a wireless sensor network deployment. A number of related algorithms and solutions have been introduced, exploiting the properties of the emitted electromagnetic signal.

A well-known tracking technique for such environments of dynamic signal variations is based on Linear Least Squares (LLS). LLS is frequently adopted in localization scenarios for the estimation of parameters which are initially unknown, by adjusting the observed parameters. The specific technique is characterized as linear or non-linear depending on the type of the derived system of equations. In more detail, in a 2-dimension setup, the target location can be estimated employing at least three lateral or angular measurements typically taken by reference points or nodes of known coordinates within the network. Authors in [14] have applied linear LLS to enhance trilateration. In such a case, the location of the target is derived via the following formula:

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \, \boldsymbol{b} \tag{1}$$

where $\hat{\boldsymbol{x}}$ is the vector of the target coordinates' estimated corrections, $\boldsymbol{A}$ is the *design matrix*, and $\boldsymbol{b}$ is the vector with the *residual observations*. However, in a highly dynamic environment where the target and the tracking device have high relative speed, it becomes very challenging for LLS to make frequent and accurate estimations, since too frequent observations tend to be highly correlated resulting in a singular $\boldsymbol{A}^T \boldsymbol{A}$.

Another approach for tracking in dynamic environments, where deterministic modelling is very difficult, is the Particle Filter Localization (PFL) technique [15]. According to the respective algorithm, a number of random samples (particles) are initially generated. The target's state, as well as the particles' state, is defined by a set of parameters, such as position and velocity. In an iterative manner, observations are periodically collected and the particles' states are accordingly updated in an effort to estimate target's actual motion. Each particle is associated with an adjustable normally distributed weight, which indicates the probability to match the target's actual state, resulting in particles which converge to that target's state. In summary, PFL is a probabilistic algorithm with promising performance when a high number of particles are considered, which comes with the cost of increased computational requirements.

The use of Kalman Filtering (KF) for target tracking within sensor networks

has risen as an attractive technique adopted by a number of related algorithms with different variations, such as Extended Kalman Filtering (EKF) and Distributed Kalman Filtering (DKF). A promising approach is introduced in [16], where a message-passing version of the Kalman-Consensus Filtering (KCF) is proposed to facilitate distributed tracking of a maneuvering target in a network of sensors with limited range. The authors introduce a hierarchical architecture to collect and distribute the estimates of the micro Kalman filters in a Peer-to-Peer (P2P) sensor network. The microfilters update the states based on the received feedback and fuse their outputs as messages to other peers. The resulted P2P/Hierarchical architecture is shown to achieve high tracking performance, however, no sensors' mobility is considered.

A very interesting approach in dynamic target tracking, which is closely related with the scenarios considered in our work, is flocking control in mobile sensor networks. In such dynamic networks, nodes are typically mobile robotic devices equipped with various sensors. An adaptive flocking control algorithm is introduced in [17], where a group of mobile sensors cooperate and adjust connectivity and topology formation to the current network environment. Moreover, a multiple dynamic target tracking algorithm, called Seed Growing Graph Partition (SGGP), is proposed to address the merging/splitting problem. Both presented algorithms rely on graph network modelling and forces which either attract or repel the nodes. The conducted experimental tests verify the effectiveness of the algorithms, however, the focus is on the group adaptation rather than the explicit target tracking process.

Lastly, a promising target tracking solution for short range dynamic networks is based on Ultra-Wide Band (UWB) signals [18][19][20]. In principle, the adopted tracking methods do not fundamentally differ from the ones employed in typical RF-based approaches, however, some special properties of UWB make it an attractive and promising solution. UWB communications are composed of very short pulses (shorter than 1ns) with a low duty cycle from 1 to 1000. The modulated signal is spread over multiple frequency bands and transmitted. Apart from communication applications, UWB is also considered for localization

applications. The position estimation is typically performed through reference nodes of known positions through well-known techniques, such as Received Signal Strength (RSS), Angle of Arrival (AoA), Time of Arrival (ToA), and Time Difference of Arrival (TDoA). It has been shown that UWB can achieve high localization accuracy, mainly due to the decomposition of the multipath signal components in the channel's high bandwidth. However, the increased accuracy can be actually achieved only through time-based positioning methods, thanks to the signal's high time resolution, rather than RSS. The obvious drawback is that time-based techniques typically require good synchronization. In addition, UWB is usually of limited range, due to its high bandwidth, making it suitable mainly for short distance tracking.

It should be noted that all these target tracking approaches are based on distance measurements or estimation of the target position. Either relying on signal strength or angle or timing, estimation errors are inevitable, due to signal variations induced by shadowing, multipath fading or interference, or even due to the tracker's dead reckoning errors. In contrast, the proposed method does not involve distance measurements, taking advantage of the fact that estimating the exact target position is not required for the tracking process. The considered scenario does not aim at localizing the target, but staying in close range for the main reason of maintaining connectivity. This is achieved by exploiting the communication signal transmitted anyway by the target and considering its strength indicators in a differential manner. Such an approach does not require special communication equipment (such as directional antennas) nor transceivers' synchronization nor multiple reference points of known coordinates.

2.5. Localization Strategies based on Swarm Controlling

A promising approach for target localization that has recently risen is the use of robotic swarms. The control of drone swarms particularly has lately attracted significant interest, mainly due to the provided practicality and flexibility. In what follows in this subsection, we discuss a number of representative swarm controlling strategies for localization purposes.

A network of drones that carry RF signal strength monitoring equipment is presented in [21]. The collected measurements are utilized to estimate distances based on a propagation model. The location of the radio transmitter is identified via trilateration of the resulted values. The authors in [22] have introduced a scheme which combines linear and non-linear programming to analyse the drones' movement constraints and perform both optimal and optimization control. Authors in [23] proposed a technique for planning motion, according to which a number of drones are grouped together to perform target tracking in a collaborative manner that involves optimization of the monitoring process and the communication with a remotely located base station. The main goal is to ensure highly reliable connections to the base state, while at the same maximizing the collected sensor data. The provided evaluation results show that the process of optimizing transmissions is crucial for the integration of data and the precision of target positioning. In [24], another promising approach is provided, according to which a swarm of drones perform collaborative localization of a radio transmitter using a control technique that is based on model prediction. The devices utilize the RSSI measurements of their radio receivers to identify the next optimal route adopting Receding Horizon Control (RHC). An Extended Kalman Filter is applied to generate the predicted parameters of the drones' motion, which are then tuned via using the D-optimality criterion. The authors in [25] adopt a similar strategy, which is also based on RHC to enable localization of multiple entities. The motion of the group of agents is determined based on the ergodic theory, where the information density distribution is adjusted by bearing-only measurements to track moving targets in real time.

3. Overall IoT System Description

The devised IoT system [1] for monitoring mobile targets/individuals called eXtreme Sensor Network (XSN), comprises the "Global Network", and a number of "Regional Networks" that communicate through the former. An illustration of the main components of the system architecture is provided in Figure 1.

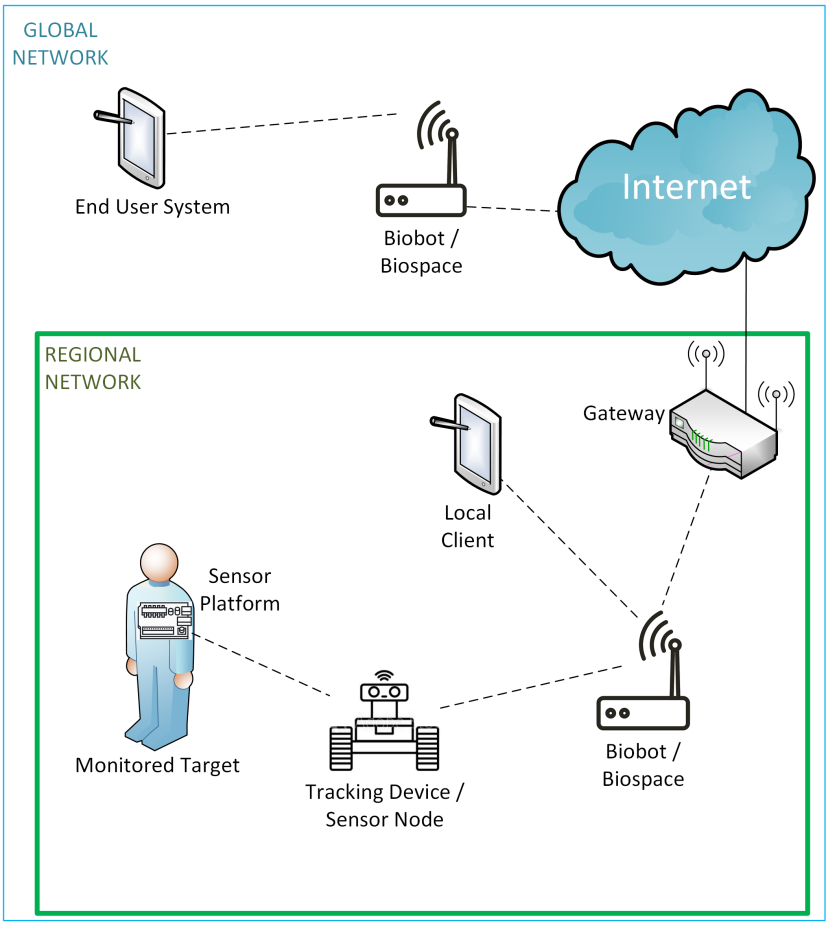


Figure 1: Architecture of the devised IoT System

3.1. Global Network

The agent-based distributed system is the part of the IoT architecture that realizes the global network, which allows efficient discovery of sensor data services and seamless access from remote locations over an unstructured distributed network. The main problem that is solved in that manner is the localization of the monitored target that carries the sensor platform and the reception of its data regardless of the exact regional network it is located in. The implemented distributed system in the form of IoT middleware is based on the EDBO (Emergent Distributed Bio-Organization) architecture [26] and was developed using JADEX [27]. The components that fulfill the aforementioned requirements and realize distributed access and remote service discovery in the context of the considered architecture are the following:

- *Biospace*: This is the platform that constitutes the basis for the creation of agents, which are able to serve sensor data.
- *Biobots*: These are the agents that access sensor nodes via RESTful requests, in order to provide information collected by sensor platforms. They are able to communicate in a distributed manner with each other for service discovery purposes.
- *End User Systems*: These are applications that access sensor services by communicating with the discovered Biobots. In the resulted IoT system, some Biobots are created in computing devices located in the same regional network with the sensor nodes, so that they have direct access to sensor data. From that point, information can be relayed over the global network. The end user systems are installed to user devices (such as tablets or smartphones) that remotely access sensor data or to remote servers that collect and process sensor information.

3.2. Regional Network

Each local deployment which allows direct access to a followed mobile target's sensor information constitutes a Regional Network. Below, the system

components which are included in such a deployment are described:

- *Monitored Target*: This might be a person (for instance victim, patient or elderly transmitting vital signs) or any mobile target that has sensors attached (a Sensor Platform) to collect monitoring data.

- *Sensor Platform*: This is an embedded system with integrated sensors carried by the monitored target. It is also equipped with an energy efficient short-range wireless network interface which enables broadcasting sensor information.

- *Tracking Device*: This an autonomous mobile robotic device which follows the monitored target with the purpose of maintaining proximity in order to gather and relay data generated by sensors or provide any type of assistance.

- *Sensor Node*: This is an embedded system with suitable wireless network interfaces to receive the signal broadcasted by the sensor platforms and then properly forward it, in the form of an entity which realizes mobile ad hoc network (MANET) routing. It ensures connectivity among multiple sensor nodes in the same regional network, but also connectivity with the Biobots of the Global Network. It may be fixed or carried by the tracking device and has sufficient processing capabilities to allow the creation of network services which enable client access to monitoring information. In the general sense and in the context of the regional network, it plays the role of the data sink.

- *Local Client*: This is an optional device handled by an end user located in the regional network to access sensor data. Since this entity lies in the region of the local network, it does not need to access the distributed service provided in the global network. It may directly query the Sensor Node to receive sensor data.

- *Gateway*: This is just a typical network device that plays the role of the gateway router for the regional network.

4. The "Hot-Cold" Target Following Scheme

The main role of the robotic device in the system architecture is carrying a sensor node that is kept in range of the sensor platform. In that manner, the monitored individual can move freely. Of course, keeping the robot close can lead to many additional promising applications, such as delivering items to a person (e.g. medicine), providing assistive services (e.g. making emergency calls) or even keeping company (numerous studies have shown that robots could help elderly people as companion pets [28]). The primary goal is maintaining communication range; for that reason we have implemented an RF-based following scheme that solely uses the strength of the signal broadcasted by the sensor platform to estimate its location and move within range.

The RF-based following scheme uses the RSSI (Received Signal Strength Indicator) value. This is an indication of the signal power received by the sensor node and transmitted by the sensor platform. Our aim is the introduction of a simple and robust technique exploiting the RF signal which is anyway broadcasted by sensing devices for communication reasons. It should be highlighted that our main goal is maintaining communication range, not accurately locating the sensor platform. Hence, the developed RF-based following scheme aims at ensuring exactly that.

The concept behind this scheme is clear. As long as RSSI is not decreased, the robot keeps moving forward until a maximum RSSI threshold is reached, indicating that the robot is too close to the monitored person (we call this status "halt"). If RSSI decreases, then the robotic device rotates and moves towards a different direction, in order to avoid moving out of range. Due to the high unreliability of the wireless link, it is not safe to make final decisions each time there is a new RSSI reading, since it could just be a random deviation from the value that actually corresponds to distance. This is the first issue we cope with. The decisions related with RSSI change are based on statistic metrics of consecutive measurements, according to the following steps:

1. The first step is the calculation of the mean RSSI value out of a number of

samples stored in the Samples Window (SW). The window size is denoted by SWS (Samples Window Size).

2. Next, after computing two mean values, we are considering the difference between the first and the second value. A positive difference (i.e. signal power increases) corresponds to the indication "Hot", whereas a negative difference (i.e. signal power decreases) corresponds to the indication "Cold".

3. Lastly, a decision is made. If the indication is not "Cold", the robot moves forward, otherwise it rotates. The rotation intends to move the robot closer to the target.

The "Hot-Cold" algorithm is presented in Algorithm 1.

Algorithm 1 The Hot-Cold Algorithm

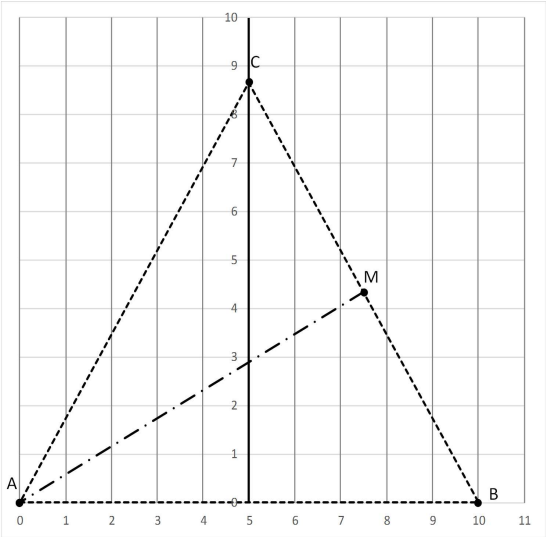
```
1: while Tracking-Following do
2:   SamplesAverage ← ∅
3:   j ← 1
4:   while j ≤ 2 do
5:     RSSI ← ∅
6:     i ← 1
7:     while i ≤ SWS do
8:       IsHalt ← false
9:       RSSI[i] ← GetRSSI
10:      SamplesAverage[j] += RSSI[i]
11:      if RSSI[i] ≤ HaltThreshold then
12:        if i < SWS then
13:          Move by robot step size
14:        end if
15:      else
16:        IsHalt ← true
17:      end if
18:      i++
19:    end while
20:    SamplesAverage[j] /= SWS
21:    j++
22:  end while
23:  if IsHalt = false then
24:    if SamplesAverage[1] > SamplesAverage[2] then
25:      Rotate
26:    end if
27:    Move by robot step size
28:  end if
29: end while
```

A crucial aspect of the introduced algorithm is the rotation angle. It is important to keep the algorithm simple and error tolerant. The whole scheme needs to exhibit advanced immunity to signal power variations, so that it is adequately robust to drive the robot close to the target. The exact localization is not significant; maintaining proximity is the highest priority. For these reasons, the main goal is to efficiently follow the target, while the rotation angle is fixed. We conclude on the optimal value of the rotation angle through a 3-stage analysis presented in the corresponding sub-sections below. The fourth subsection presents a convergence analysis which proves that by employing the introduced Hot-Cold algorithm, the robot reaches the followed target in finite number of steps.

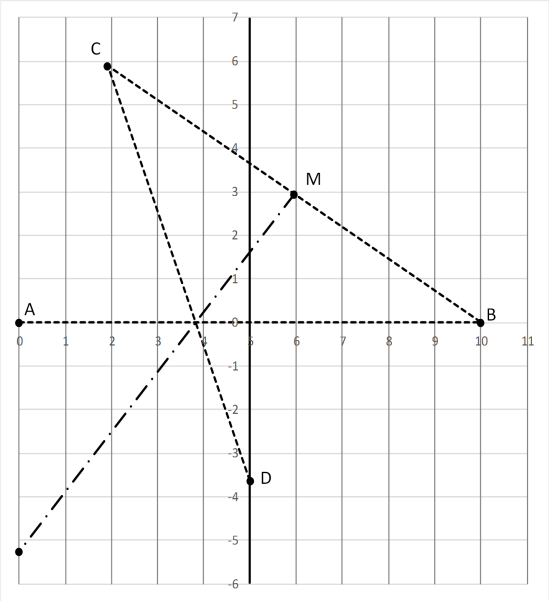
4.1. Geometrical Analysis of Rotation Angle

The first stage of this analysis focuses on the geometric properties of the proposed target following scheme. The objective here is to estimate a range of rotation angles which rapidly move the robotic device closer to the area where the target is most probably located.

We consider the scenario where the robot moves from point A to point B by one step equal to 10 distance units, as presented in Figure 2a. The starting robot position is A and a step later it is located at position B. Assuming that at that point the Hot-Cold algorithm deduces that the robot has moved away from the target, a rotation should be performed. In this example, the rotations are considered to be counter-clockwise, without loss of generality. The fact that the target is closer to A than B indicates that it is located in the area left from the vertical line bisector of the segment AB (note that the depicted y-axis lies on this bisector). In order to reach this area, the rotation angle needs to be higher than 120 degrees and lower than 270 degrees. The figure illustrates rotation by 120 degrees, which positions the robot after its second step at point C. Assuming again that at point C the robot is further from the target than it was at point B, a new rotation will take place. The target should be now positioned on the lower half of the area divided by the vertical line bisector



(a) Rotation angle: 120 degrees



(b) Rotation angle: 144 degrees

Figure 2: Three consecutive steps with rotations at angle of a) 120 degrees and b) 144 degrees

AM, where M is the midpoint of the line sector AB. It is noted that point A

lies on this bisector, when the robot rotates by 120 degrees counter-clockwise, as illustrated in Figure 2a. Taking also into account the observation of the first step which dictated that the target is on the left side of the y-axis, it is proven that the target is located in an area that is accessible by performing fixed angle rotation between 120 to 144 degrees. Specifically, this area, where the target is positioned, lies under the line defined by the bisector AM and left from the line defined by the y-axis. When rotating by 120 degrees, the robot will be positioned after the third step back to point A, which is at the borderline of the target area.

The graphical depiction of the the 3-step movement when rotating by 144 degrees is provided in Figure 2b. It can be seen that after three steps, the robot reaches position D, which is on the right border of the target area that lies under the line bisector intersecting point M and left from the line defined by y-axis. Please note that in the case of 120-degrees rotation illustrated in Figure 2a, point D overlaps with point A. In conclusion, the geometrical analysis reveals that the fixed rotation angle has to be higher than 120 and lower than 144 degrees for the robot to reach the target area in the minimum number of steps, when adopting Hot-Cold target following. Obviously, rotation angles out of this range could eventually drive the robot in the target area, however, on average more steps would be required, whereas the objective is to reach the target as fast as possible.

4.2. Numerical Analysis of Rotation Angle

The second stage of the analysis in the effort to identify the optimal rotation angle for efficient following via the Hot-Cold algorithm focuses on calculating the number of fixed rotations required to reach a target point at any angle. Specifically, the objective is to identify the rotation angle ϕ, which minimizes the average number of rotations required ($\overline{\omega}$) to reach any target angle θ_i within a deviation $\pm\varepsilon$.

The mathematical expression that relates the aforementioned variables is

shown in Eq.(2).

$$\theta_i - \varepsilon \leq (\omega_i \times \phi) mod 360 \leq \theta_i + \varepsilon \tag{2}$$

Solving Eq.(2) for ω leads to Eq.(3).

$$\begin{gathered} \theta_i - \varepsilon \leq \omega_i \phi - 360\kappa_i \leq \theta_i + \varepsilon \Rightarrow \\ \theta_i - \varepsilon + 360\kappa_i \leq \omega_i \phi \leq \theta_i + \varepsilon + 360\kappa_i \Rightarrow \\ \frac{\theta_i - \varepsilon + 360\kappa_i}{\phi} \leq \omega_i \leq \frac{\theta_i + \varepsilon + 360\kappa_i}{\phi} \end{gathered} \tag{3}$$

where κ_i is the lowest non-negative integer that makes ω_i integer. The optimization problem is formulated as follows:

$$\phi = \operatorname{argmin} \overline{\omega} \ , \ \forall i \in [1, n] \ \ , \text{ where } n \in \mathbb{Z}^+ \tag{4}$$

s.t.

$$\phi, \theta_i, \varepsilon \in [0, 360) \tag{5}$$

$$\phi \in \mathbb{Z} \tag{6}$$

$$\omega_i \in \mathbb{Z}^* \tag{7}$$

$$\kappa_i \in [0, \phi] \wedge \kappa_i \in \mathbb{Z} \tag{8}$$

In order to identify ω_i, we solve Eq.(3) in a numerical approach for all integer values of θ_i and ϕ, and different values of ε. Given that $\kappa_i \in [0, \phi]$, consecutive integer values of κ_i are tested in each iteration until the first solution of Eq.(3) is found. The optimal ϕ is the one which yields the lowest:

$$\overline{\omega} = \frac{\sum_{i=1}^{n} \omega_i}{n} \tag{9}$$

The procedure that provides numerical solution to the described problem through iterative trials was developed and executed in MATLAB. In more detail, we tested all rotation angles from 121 up to 143 degrees according to the findings of the geometrical analysis (120 and 144 degrees angles are borderline cases,

hence, non-optimal). It is noted that only angles of integer degrees are considered; further subdividing angles makes no difference, since the potential benefits would be minimal and in a real scenario a robotic following device could not make that accurate turns anyway. For each rotation angle (ϕ) we compute the number of rotations required (ω_i) to reach any target angle (θ_i) from 0 to 359 degrees within a deviation ($\pm\varepsilon$) from 0 to 30 degrees.

Figure 3 provides a heatmap of $\overline{\omega}$ values that have been derived by averaging over all 360 values of θ_i. As expected, high ε ensures low average number of rotations required (lighter regions). Cells with "X" represent cases where it was impossible to reach some target angles (θ_i) within the respective deviation (ε). The optimal rotation angle should require low number of rotations to reach a large number of target angles within small deviation. In order to conclude on the value of this angle according to the specific criteria, we have further averaged the $\overline{\omega}$ values over each rotation angle (ϕ) and plotted them along with the percentage of valid trials in Figure 4. It can be deduced that the lowest mean number of rotations (16.78) with no invalid trials is achieved when ϕ equals 139 degrees.

4.3. Exhaustive-Simulation Analysis of Rotation Angle

The third stage of the analysis for identifying the optimal rotation angle that would efficiently drive the robotic device close to the target involves exhaustive simulations in MATLAB. Specifically, the objective of this final part of the analysis is to compute the number of required steps taken by the robotic device to approach a fixed target when adopting the Hot-Cold algorithm principles. In each configuration, the target is placed ρ distance units away from the robot's starting position and at a direction of β degrees. The followed approach is actually exhaustive; simulations are executed for all integer values of β ranging from 0 to 359 degrees and ρ ranging from 10 to 100 distance units. Each simulation is terminated when the robotic device approaches the target within τ distance units; all integer values from 1 to 10 are tested. It is noted that a distance unit is set equal to the length of one step. Based on the Hot-Cold

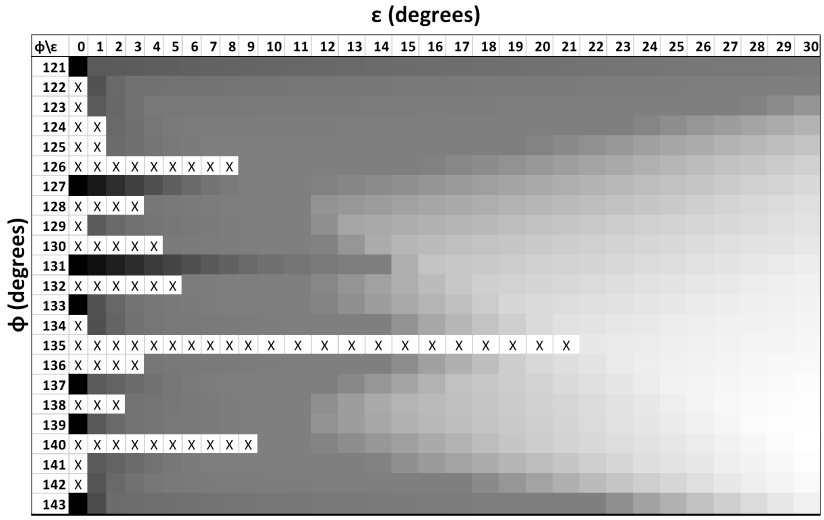


Figure 3: Heatmap of average number of rotations required for different rotation angles (ϕ) and target angle deviations ($\pm\varepsilon$) in degrees — Legend: Lightest is 3, Darkest is 179.5, 'X' is invalid

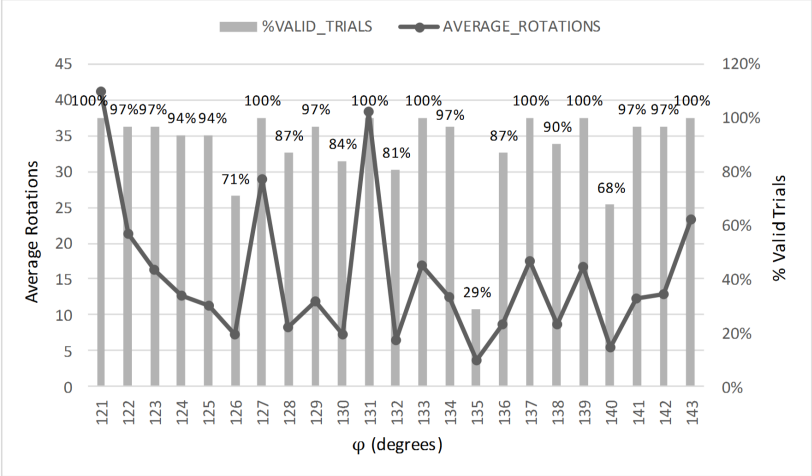


Figure 4: Overall average number of rotations required and percentage of valid trials against different rotation angles (ϕ)

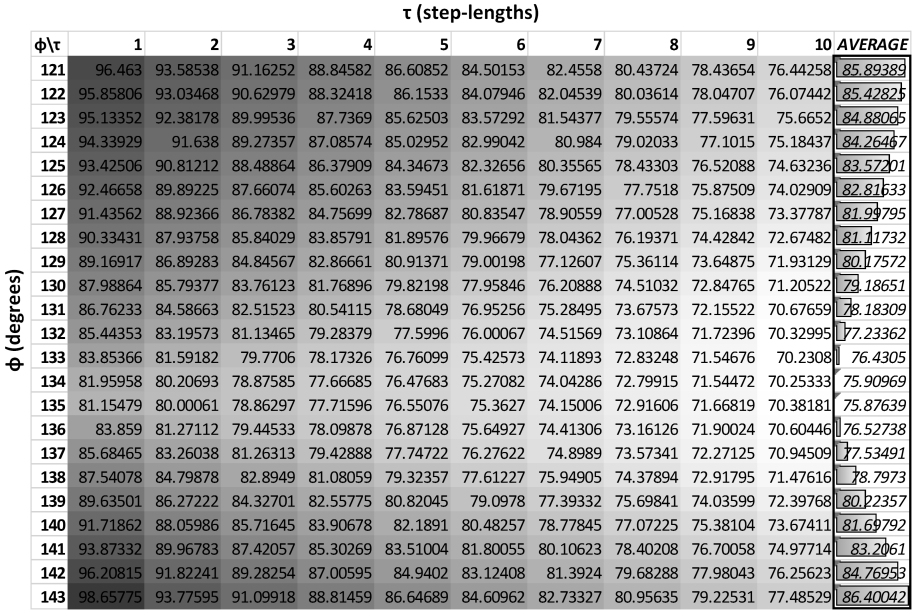
concept, the robot rotates at a fixed angle every time its new position is at a higher distance from the target than its previous position. Simulations are performed for all integer rotation angles (ϕ) from 121 to 143 degrees, with the total number of executed simulations just for this last stage of the analysis reaching 7,534,800.

The findings are depicted in the heatmap of Figure 5. For each combination of ϕ and τ, we calculate the mean number of steps considering all ρ and β values; the results are shown in the cells of the heatmap. As expected, the more relaxed the termination condition is (high τ values), the fewer steps are required (lighter regions). Regarding the optimal rotation angle, a clear pattern is revealed, especially when looking at the overall averages presented in the last column of the figure. It is evident that the closer a rotation angle is to 135 degrees, the fewer steps are required to reach the target. The minimum number of 75.87639 averaged steps is achieved for ϕ exactly equal to 135 degrees.

Conclusively, the conducted 3-stage analysis shows that the optimal rotation angle for a robotic device adopting the introduced Hot-Cold algorithm to approach a target as fast as possible is in the range of 135 to 139 degrees. Considering the inevitable declination from the set rotation angle of a robotic vehicle, configuring it to the median value of 137 degrees is a safe choice.

4.4. Convergence Analysis

Following the 3-stage analysis for the determination of the optimal rotation angle, in this subsection a convergence analysis is presented, which was conducted to prove whether the Hot-Cold algorithm theoretically ensures target approaching within a finite number of steps. We break down this analysis in two parts, Hot mode and Cold mode, demonstrating that in both modes the introduced algorithm manages to converge robot's position close to target's position.



τ (step-lengths)

ϕ (degrees) \ τ	1	2	3	4	5	6	7	8	9	10	AVERAGE
121	96.463	93.58538	91.16252	88.84582	86.60852	84.50153	82.4558	80.43724	78.43654	76.44258	85.89389
122	95.85806	93.03468	90.62979	88.32418	86.1533	84.07946	82.04539	80.03614	78.04707	76.07442	85.42825
123	95.13352	92.38178	89.99536	87.7369	85.62503	83.57292	81.54377	79.55574	77.59631	75.6652	84.88065
124	94.33929	91.638	89.27357	87.08574	85.02952	82.99042	80.984	79.02033	77.1015	75.18437	84.26467
125	93.42506	90.81212	88.48864	86.37909	84.34673	82.32656	80.35565	78.43303	76.52088	74.63236	83.57201
126	92.46658	89.89225	87.66074	85.60263	83.59451	81.61871	79.67195	77.7518	75.87509	74.02909	82.81633
127	91.43562	88.92366	86.78382	84.75699	82.78687	80.83547	78.90559	77.00528	75.16838	73.37787	81.99795
128	90.33431	87.93758	85.84029	83.85791	81.89576	79.96679	78.04362	76.19371	74.42842	72.67482	81.11732
129	89.16917	86.89283	84.84567	82.86661	80.91371	79.00198	77.12607	75.36114	73.64875	71.93129	80.17572
130	87.98864	85.79377	83.76123	81.76896	79.82198	77.95846	76.20888	74.51032	72.84765	71.20522	79.18651
131	86.76233	84.58663	82.51523	80.54115	78.68049	76.95256	75.28495	73.67573	72.15522	70.67659	78.18309
132	85.44353	83.19573	81.13465	79.28379	77.5996	76.00067	74.51569	73.10864	71.72396	70.32995	77.23362
133	83.85366	81.59182	79.7706	78.17326	76.76099	75.42573	74.11893	72.83248	71.54676	70.2308	76.4305
134	81.95958	80.20693	78.87585	77.66685	76.47683	75.27082	74.04286	72.79915	71.54472	70.25333	75.90969
135	81.15479	80.00061	78.86297	77.71596	76.55076	75.3627	74.15006	72.91606	71.66819	70.38181	75.87639
136	83.859	81.27112	79.44533	78.09878	76.87128	75.64927	74.41306	73.16126	71.90024	70.60446	76.52738
137	85.68465	83.26038	81.26313	79.42888	77.74722	76.27622	74.8989	73.57341	72.27125	70.94509	77.53491
138	87.54078	84.79878	82.8949	81.08059	79.32357	77.61227	75.94905	74.37894	72.91795	71.47616	78.7973
139	89.63501	86.27222	84.32701	82.55775	80.82045	79.0978	77.39332	75.69841	74.03599	72.39768	80.22357
140	91.71862	88.05986	85.71645	83.90678	82.1891	80.48257	78.77845	77.07225	75.38104	73.67411	81.69792
141	93.87332	89.96783	87.42057	85.30269	83.51004	81.80055	80.10623	78.40208	76.70058	74.97714	83.2061
142	96.20815	91.82241	89.28254	87.00595	84.9402	83.12408	81.3924	79.68288	77.98043	76.25623	84.76953
143	98.65775	93.77595	91.09918	88.81459	86.64689	84.60962	82.73327	80.95635	79.22531	77.48529	86.40042

Figure 5: Mean number of steps required against rotation angle (ϕ) in degrees and halt distance from target (τ) in step-lengths

Lemma 1

A robotic device which performs target following using the Hot-Cold algorithm always approaches the target while in Hot mode, given that their horizontal distance is greater than half of the robot's step, considering no signal fading, SWS equal to 1, and random walk as the target's mobility model.

Proof 1

Regarding the target's movement, since it employs random walk, in every step it changes its distance (d) from the robot in a uniform manner, with a mean value of 0.

Focusing on the robot's movement, being in Hot mode means that the current distance d is lower or equal to the corresponding distance during the previous step. It is noted that the two distances are directly comparable using the respective RSSI values, since for this theoretical analysis we assume that signal strength is only affected by propagation attenuation, not fading of any kind.

Figure 6 illustrates a general tracking scenario, where the robot performs four steps ($AB = BC = CD = DE = s$) and is located in five consecutive points (starting point A, ending point E). Covering the general model, we consider two alternative locations (at opposite sides symmetrical to the horizontal axis) for the target: P and P'. The application of the Pythagorean theorem yields the following equations for robot's first step:

$$AP^2 = PO^2 + AO^2 \tag{10}$$

$$BP^2 = PO^2 + BO^2 = PO^2 + (AO - AB)^2 \tag{11}$$

$$BP \leq AP \Rightarrow BP^2 \leq AP^2 \Rightarrow$$
$$PO^2 + (AO - AB)^2 \leq PO^2 + AO^2 \Rightarrow$$
$$AO \geq s/2 \tag{12}$$

Eq.(12) proves that the robotic device stays in Hot mode and moves forward approaching the target (current distance not greater than the previous one) as

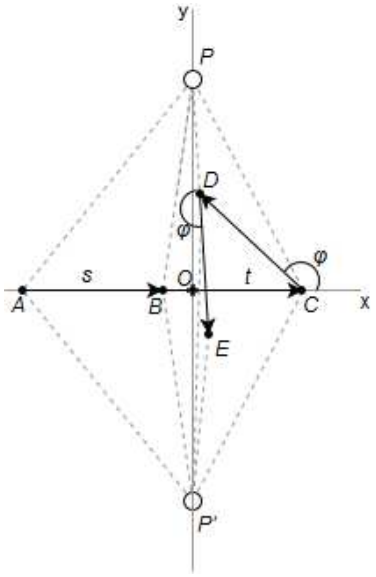


Figure 6: General tracking scenario, representing robot's 4 consecutive steps, transiting from Hot mode (AB, BC) to Cold mode (CD, DE)

long as their horizontal distance (AO) just before the current step is not shorter than the robot's half step size ($s/2$). At the point this condition ceases to hold (point B), the robot increases distance d with its immediate next step (point C), so transits to Cold mode. It is noted that due to symmetry the same also holds when the target is positioned at P'.

Lemma 2

A robotic device which performs target following using the Hot-Cold algorithm approaches in Cold mode the target right after the first rotation, when their signed vertical distance is higher than $0.7304s - 1.0649t$ (t is their horizontal absolute distance, s is the robot step size and $t > \frac{s}{2}$) or otherwise right after the second rotation, considering no signal fading, SWS equal to 1, and random walk as the target's mobility model.

Proof 2

Using as reference Figure 6, we now focus on the case that the robot transits to Cold mode when it moves to point C. The condition for this transition is $CP > BP$ (or equivalently $CP' > BP'$), which yields $CO > BO$. Given that $BC = s$, the latter condition holds when $t = OC > s/2$. According to the principles of the Hot-Cold algorithm, we consider that the robot rotates at point C by $\phi = 137°$ counterclockwise, without loss of generality (in case of clockwise rotation, points P and P' can be just considered exchanged). Right after the rotation and robot's movement by one step-size (s), its new location is D and there can be two cases regarding its distance from the target: i) it has been decreased (e.g. $DP < CP$) or ii) it has not been decreased (e.g. $DP' > CP'$). Hence, this analysis initially focuses on identifying the relation between the target's y-coordinate (denoted by r_1) and the robot's ability to approach right after its first rotation (point D). Specifically, we estimate the r_1 threshold which ensures that the considered distance after rotating becomes smaller. This part of the problem is formulated and solved as follows, where $t > \frac{s}{2}$:

$$CP > DP \Rightarrow CP^2 > DP^2 \Rightarrow$$

$$r_1^2 + t^2 > (t + s\cos(\phi))^2 + (s\sin(\phi) - r_1)^2 \Rightarrow$$

$$0 > s^2(\cos^2(\phi) + \sin^2(\phi)) + 2ts\cos(\phi) - 2r_1 s\sin(\phi) \Rightarrow$$

$$2r_1\sin(\phi) > s + 2t\cos(\phi) \xrightarrow{0 \leq \phi \leq \pi}$$

$$r_1 > \tfrac{1}{2}s\csc(\phi) + t\cot(\phi) \xrightarrow{\phi = 137°}$$

$$r_1 > 0.7304s - 1.0649t \tag{13}$$

Next, this analysis focuses on the condition for approaching the target right after the second rotation, which requires that at the first rotation the robot increased its distance, hence, it remained in Cold mode. In Figure 6, this is the case when the target is located at point P'. Following an approach similar to the above, it holds (where $t > \frac{s}{2}$):

$$DP > EP \Rightarrow DP^2 > EP^2 \Rightarrow$$

$$(t + s\cos(\phi))^2 + (s\sin(\phi) - r_2)^2 >$$

$$(t + s\cos(\phi) + s\cos(2\phi))^2 + (s\sin(\phi) + s\sin(2\phi) - r_2)^2 \Rightarrow$$

$$s + 2t\cos(\phi) - 2r_2\sin(\phi) >$$

$$s(\cos(\phi) + \cos(2\phi))^2 + 2t(\cos(\phi) + \cos(2\phi)) +$$

$$s(\sin(\phi) + \sin(2\phi))^2 + 2r_2(\sin(\phi) + \sin(2\phi)) \Rightarrow$$

$$2r_2\sin(2\phi) > 4s\cos^2(\tfrac{\phi}{2}) + 2t\cos(2\phi) - s \xrightarrow[\frac{3\pi}{2} \leq \phi \leq 2\pi]{\frac{\pi}{2} \leq \phi \leq \pi}$$

$$r_2 < \frac{4s\cos^2(\frac{\phi}{2}) + 2t\cos(2\phi) - s}{2\sin(2\phi)} \xrightarrow{\phi = 137^\circ}$$

$$r_2 < 0.2294s - 0.0629t \tag{14}$$

From Eq.(13) and Eq.(14) it derives that the r_1 threshold is always lower than the r_2 threshold, given that $t > \frac{s}{2}$, which is always true. This means that if the robot moves away form the target after the first rotation, it will definitely approach it after the second rotation, transiting from Cold to Hot mode. Furthermore, following the same analytical method, it is shown that in case of two required rotations, the robot-target distance (illustrated by EP' in Figure 6) is eventually smaller than the original distance before any rotations (CP'). Specifically, for the general case of considering target's position as P, the problem is formulated as follows, where r_3 is the target's y-coordinate and $t > \frac{s}{2}$:

$$CP > EP \Rightarrow CP^2 > EP^2 \Rightarrow$$

$$r_3^2 + t^2 >$$

$$(t + s\cos(\phi) + s\cos(2\phi))^2 + (s\sin(\phi) + s\sin(2\phi) - r_3)^2 \Rightarrow$$

$$0 > s(\cos(\phi) + \cos(2\phi))^2 + 2t(\cos(\phi) + \cos(2\phi)) +$$

$$s(\sin(\phi) + \sin(2\phi))^2 - 2r_3(\sin(\phi) + \sin(2\phi)) \Rightarrow$$

$$r_3(\sin(\phi) + \sin(2\phi) >$$

$$2s\cos^2(-\tfrac{\phi}{2}) + t(\cos(\phi) + \cos(2\phi)) \xrightarrow{\phi = 137^\circ}$$

$$r_3 < 2.1251t - 0.8646s \tag{15}$$

505

From Eq.(13) and Eq.(15) it derives that the r_1 threshold is always lower than the r_3 threshold, given that $t > \frac{s}{2}$, which is always true. Thus, it is proven that starting from the position where the robot enters the Cold mode (point C in Figure 6), it will always approach the target either right after the first 137° rotation in case their signed vertical distance is higher than $0.7304s - 1.0649t$ or right after its second rotation, otherwise.

5. System Evaluation

The evaluation of the devised system is based on a dual approach. Initially, we focus on the introduced Hot-Cold target following algorithm, which is thoroughly evaluated in a simulation-based manner. Then, a prototype is implemented, which is tested in controlled laboratory conditions.

5.1. Simulation-based Evaluation of Target Following Scheme

In order to thoroughly evaluate the main focus of this work, which is the introduced Hot-Cold RF-based target following scheme, a simulator [29] was developed in the Processing Integrated Development Environment [30]. There are two main objectives of the conducted simulations: i) identify the optimal SWS (Samples Window Size) values and ii) evaluate Hot-Cold performance by comparing it against a reference target following scheme. The set values of the main simulation parameters are shown in Table 1. The direction change in the simulator takes place in two simulation cycles, that is 1 sec duration. At this point, it is clarified that the purpose of the simulations is to conduct comparison-based evaluation, which is successfully achieved by setting the exact same parameter values to the different simulation settings which are compared against each other. It is also noted that radio propagation modeling is based on the log-distance path loss model with log-normal shadowing, since it is widely accepted and generic enough to simulate various environments [31]. Path loss at the reference distance is estimated according to Friis formula [32], resulting

in the following equation for estimating signal path loss:

$$PL = 10n \log d + 20 \log f + 20 \log \tfrac{4\pi}{c} + X_g \tag{16}$$

where d is the transmitter-receiver distance, f is the central frequency, c is the speed of light, X_g is a Gaussian random variable with mean $\mu = 0$ and standard deviation σ modelling slow fading due to mobility/shadowing. It should be noted that the correlation between the received signal strength and the distance is verified in multiple studies including the experiments and regression analyses performed in the context of WINNER I [33] and WINNER II [34] projects on wireless channel modeling. However, it is undeniable that this correlation is degraded by the presence of any form of noise (such as fading and interference). Driven by this fact, the proposed technique avoids the direct computation of distances based on RSSI, rather it utilizes indications of RSSI changes to roughly deduce whether the robotic device approaches the target or not.

Before proceeding with the performance evaluation, we first compare the efficiency of the adopted technique of averaging RSSI values within the Samples Window (SW) against the dominant Extended Kalman Filtering (EKF) technique. We have chosen EKF as a reference noise cancellation algorithm, on the grounds that it is probably the most widely accepted scheme for nonlinear state estimation in navigation systems, such as the Global Positioning System [35]. In our case, EKF was applied on the observed RSSI values that vary as the target moves and the robot follows it, which obviously constitutes a nonlinear model. The only observable metric is the received RSSI, while the estimated state is the true RSSI (i.e. relieved from noise), with both values being scalar. The EKF observation model as well as the process noise covariance were set to 1. On the other hand, the observation noise covariance, which optimally reflects the standard deviation of the Gaussian noise experienced by the receiver, is set to varying values given that the exact shadowing effect is unknown. Figure 7 presents the percent of correct tracking estimates (i.e. Hot or Cold) made when we replace in the introduced algorithm the SW averaging with EKF for different values of observation noise covariance. The dashed line depicts the

Table 1: Simulation Parameters

Simulation Parameter	**Value**
Simulation duration	1000 sec
Executions per simulation	5
Simulation space	100×100 m^2
Robot speed	7.2 km/h (default)
Target speed	3.6 km/h
Target mobility pattern	Random waypoints
Halt distance	3 m
Broadcast interval / Simulation cycle period	0.5 sec
Target TX power	0 dBm
Robot RX sensitivity	-94 dBm
Target TX antenna gain	0 dBi
Robot RX antenna gain	2 dBi
Path loss exponent	2.8
Signal frequency	2.4 GHz

respective results when we average the values within the SW, as described in Algorithm 1. It becomes evident that in the general case, the two techniques perform similarly in the context of the Hot-Cold algorithm. Of course, it should be clarified that EKF and its variations can be very promising, since they are highly adjustable and can be tuned for specific mobility and signal propagation models. It is undeniable that EKF exhibits its great potential when fusing the readings from multiple sensors. However, since in the considered environment only RSSI readings are assumed to be available and there can be no safe inference about the target's mobility model, the SW averaging technique is adopted.

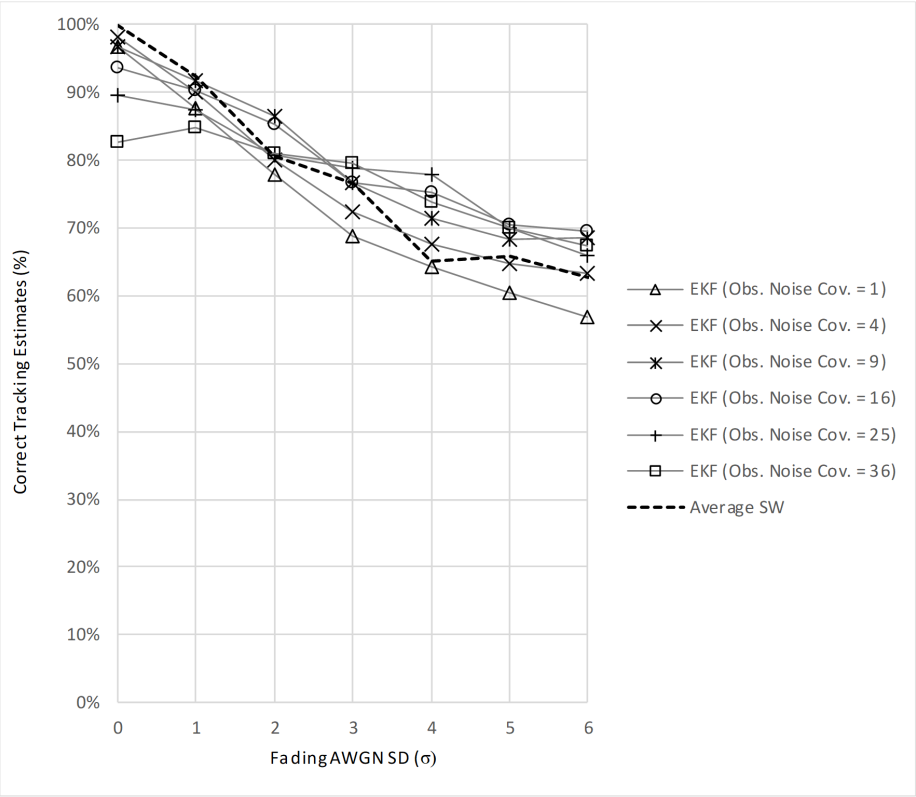


Figure 7: Correct tracking estimates of the adopted SW averaging technique compared with EKF for different values of observation noise covariance.

Nevertheless, it is clear that Hot-Cold is modular enough to allow replacements of its RSSI estimating component with any suitable filtering technique, such as EKF.

The key performance indicators that are used for evaluation purposes are the following: i) Average Distance: The distance between the robot and the target averaged over the simulation duration. Lower values indicate better performance. ii) Cycles in Range: Simulation cycles during which the robot stays in the communication range of the target. Higher values indicate better performance. iii) Cycles in Halt: Simulation cycles during which the robot freezes, due to short (halt) distance from the target. Higher values indicate better per-

formance.

In our effort to identify the optimal SWS values for the Hot-Cold algorithm, we have initially run simulations for SWS ranging from 1 to 10 and for standard deviation of noise due to fading (σ) ranging from 0 to 6. For each SWS value, the minimum Average Distance is identified, as well as the Average Distance corresponding to different σ values. In Figure 8, we plot the difference of each Average Distance value from the minimum value, along with the mean and standard deviation. It can be seen that on average the algorithm achieves smallest differences from the minimum distances for SWS values in the range of 3 to 7. In the place of "Average Distance", Figure 9 depicts "Cycles in Range" and Figure 10 depicts "Cycles in Halt", while considering the difference from the maximum value. The former shows that the robot stays more time in range for SWS values lying in the range of 4 to 7. Similarly, Figure 10 reveals that the robot reaches halt distance more times with SWS values ranging from 3 to 6.

The first part of the simulation-based evaluation of the introduced Hot-Cold algorithm has shown that on average highest following efficiency is achieved for SWS values in the range of 3 to 7. Hence, these are the SWS values that we are using to compare Hot-Cold versus a reference target following algorithm. The simulation results about the "Average Distance" metric are presented in Figure 11. The chart plots "Average Distance" as a function of the standard deviation (σ) of the Additive White Gaussian Noise (AWGN) used for modeling signal fading for different SWS values of the Hot-Cold algorithm as well as the Trilateration target following algorithm and a control case. The latter refers to the case that the robot is completely static, staying in its original position throughout the whole duration of the simulation, which results in average distance of 30 meters. As expected, lower noise yields shorter following distance. Moreover, the results reveal that for increased noise with σ equal to 5 or higher, RSSI-based target following algorithms become too inaccurate and they perform even worse than the control case. It should be noted though that this behavior is also due to the fact that while the target is constrained within

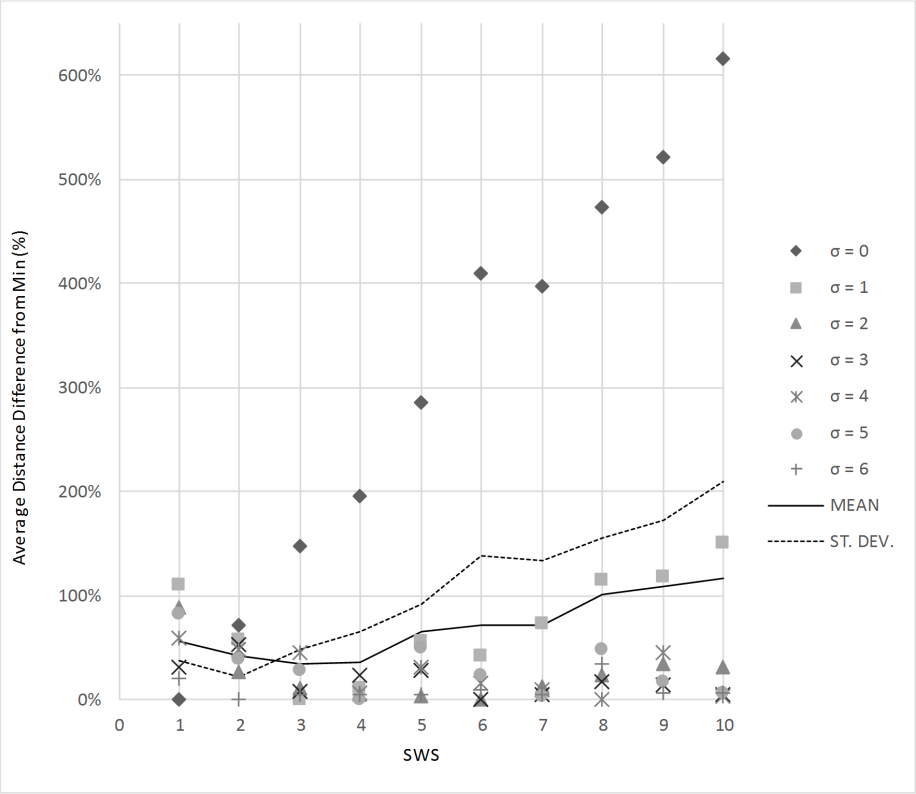


Figure 8: Difference from the minimum average distance between robot and target versus SWS values of the Hot-Cold algorithm, for different standard deviation values of noise due to fading (σ).

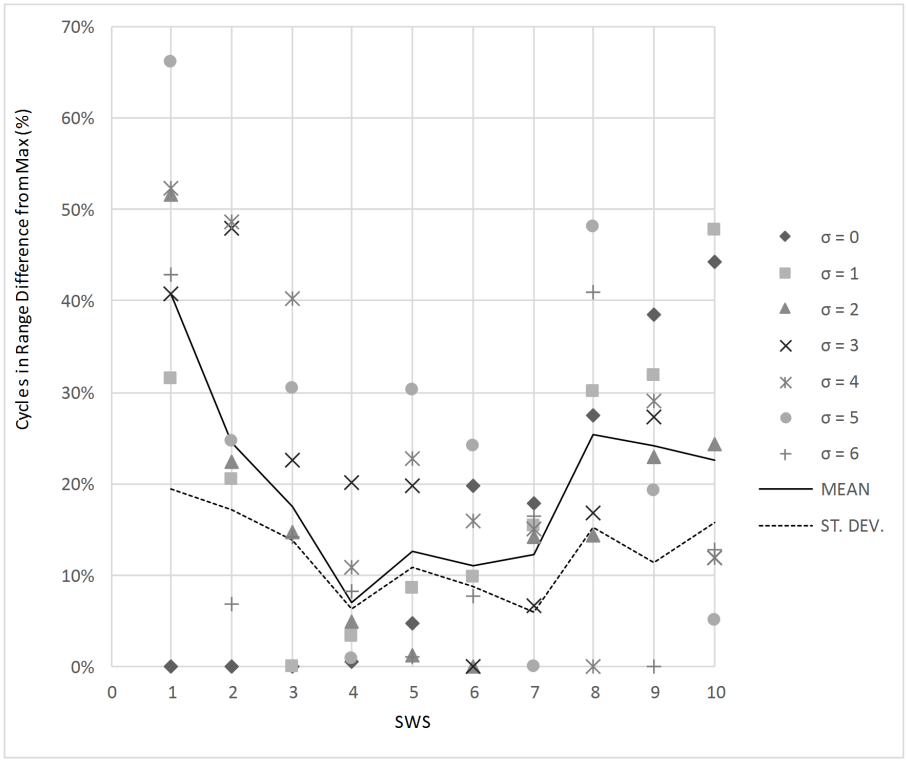


Figure 9: Difference from the maximum cycles the robot stays in communication range with the target versus SWS values of the Hot-Cold algorithm, for different standard deviation values of noise due to fading (σ).

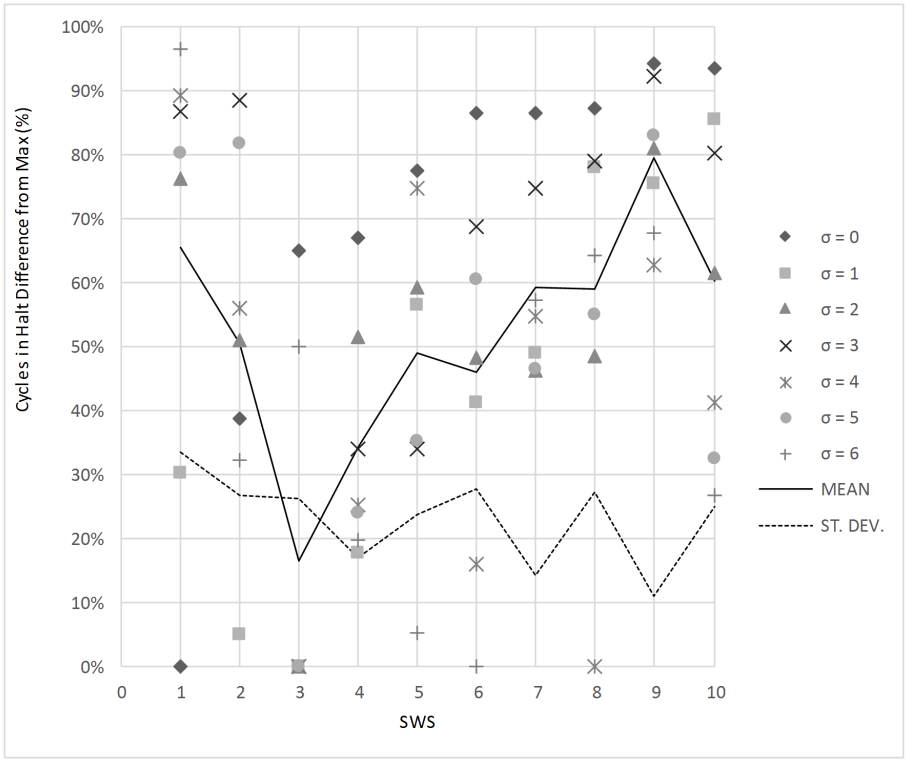


Figure 10: Difference from the maximum cycles the robot stays in halt distance from the target versus SWS values of the Hot-Cold algorithm, for different standard deviation values of noise due to fading (σ).

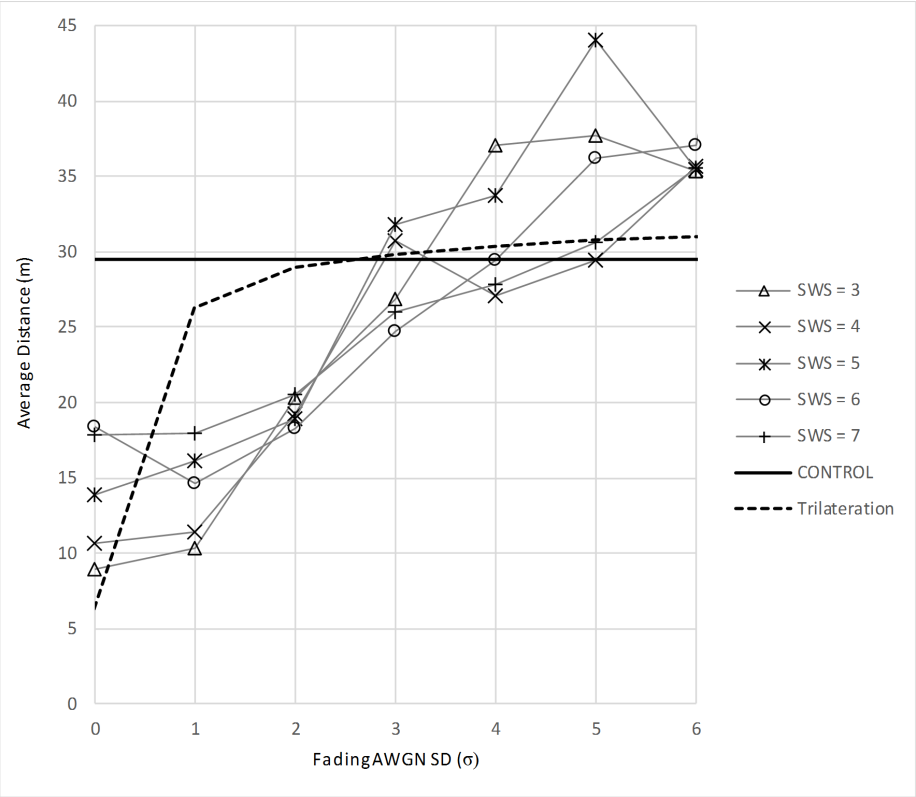


Figure 11: Average distance between robot and target versus the standard deviation of noise due to fading (σ), for different SWS values of the Hot-Cold algorithm, the Trilateration algorithm, and the control case.

the simulation space limits, the robot is free to move even beyond those limits, which causes large distances in case of too inaccurate following. However, for lower σ values, Hot-Cold outperforms Trilateration, except from the unrealistic case of noise-free signal ($\sigma = 0$).

600 Similar conclusions can be drawn when comparing "Cycles in Range". Figure 12 shows that the robot can stay longer in communication range when Hot-Cold is used for noise levels lower than six standard deviations. For higher levels of noise, staying still (control case) would actually perform better than trying to follow. In the unrealistic case of noise absence, both Hot-Cold and Trilateration

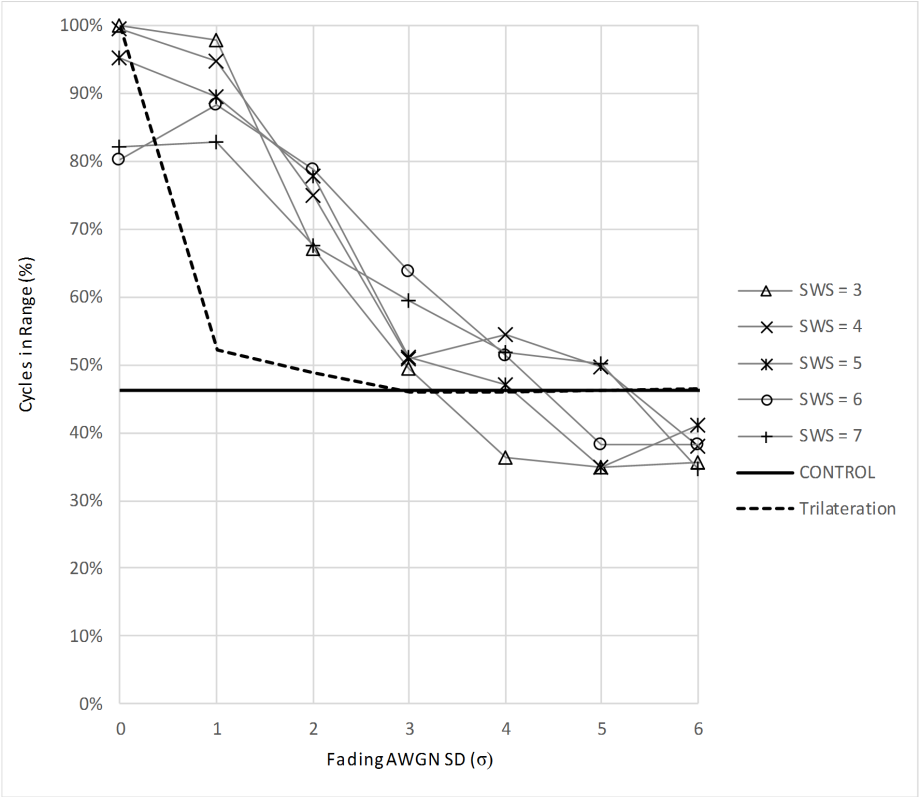


Figure 12: Simulation cycles during which the robot stays within the target's communication range versus the standard deviation of noise due to fading (σ), for different SWS values of the Hot-Cold algorithm, the Trilateration algorithm, and the control case.

can achieve 100% simulation cycles in range. Reaching halt distance from target (i.e. within 3 meters for our simulations) is even more challenging for the target following process. It can be seen in Figure 13 that the Trilateration curve overlaps with the control curve at almost 0% for fading due to shadowing with σ higher than 1. In fact, excluding the unrealistic case of σ equal to 0, Hot-Cold manages to drive the robot to halt distance clearly more frequently than Trilateration.

As last part of the simulation-based evaluation of the target following scheme, we investigate the impact of the robot-target relative speed on the tracking ef-

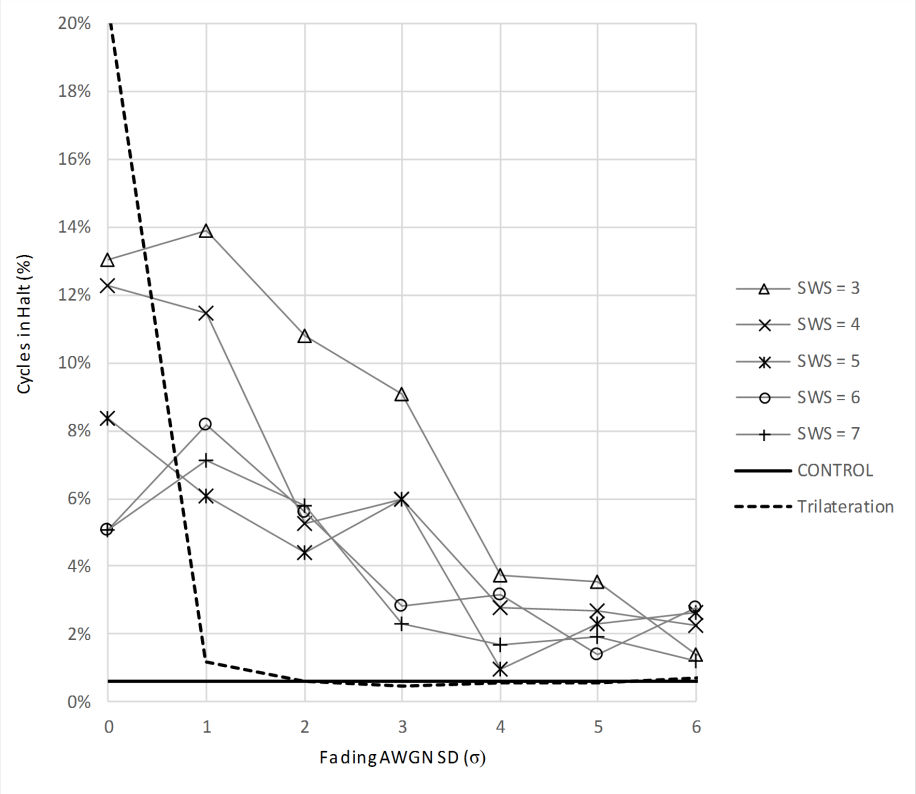


Figure 13: Simulation cycles during which the robot stays within halt distance from the target versus the standard deviation of noise due to fading (σ), for different SWS values of the Hot-Cold algorithm, the Trilateration algorithm, and the control case.

ficiency. Figure 14 presents the average distance between the target and the robot for different values of robot speed, keeping target's speed fixed at 3.6 km/h and signal fading σ equal to 2. The minimum considered robot speed is 3.6 km/h, since following a target which moves slower than it does not really make sense. The respective control and trilateration values are also plotted as references. It is evident that there is a relation between the robot speed and the SWS value. This is attributed to the distance covered by the robot relatively to the target and the spatial frequency of the RSSI readings. In more detail, since RSSI readings occur at fixed time periods of 0.5 sec, traveling at higher speeds leads to longer distances covered between the readings. In that manner, Hot-Cold decisions become more accurate, thus, tracking becomes more efficient (lower average robot-target distance). However, if the robot moves too fast, then it travels too far before making a rotation decision, which degrades tracking efficiency. This behavior is intensified by the impact of the SWS value, since smaller sample windows lead to more frequent but less accurate decisions. The results reveal that for each value of robot speed there is an optimal SWS which minimizes the average distance. For instance, in the context of the considered values, when the robot moves at 18 km/h, the optimal size of the samples window is 3, assuming that the target speed is 3.6 km/h and the signal fading σ is 2.

The results of the conducted simulations for the evaluation of the proposed Hot-Cold algorithm have provided insights for parameters' optimization and conclusions on performance through a comparative study. The Trilateration algorithm is employed as reference and is shown to perform excellent in ideal condition, when there is actually no fading due to shadowing. In all other cases, it fails to drive the robot close to the target as efficiently as Hot-Cold does. The reason is that Trilateration is based on accurate calculation of the robot-target distance according to the received signal strength. This approach provides perfect results in the unrealistic case of a noise free channel, but degrades fast as fading increases. On the other hand, the introduced algorithm provides target following capabilities based on relative signal differences after storing values in

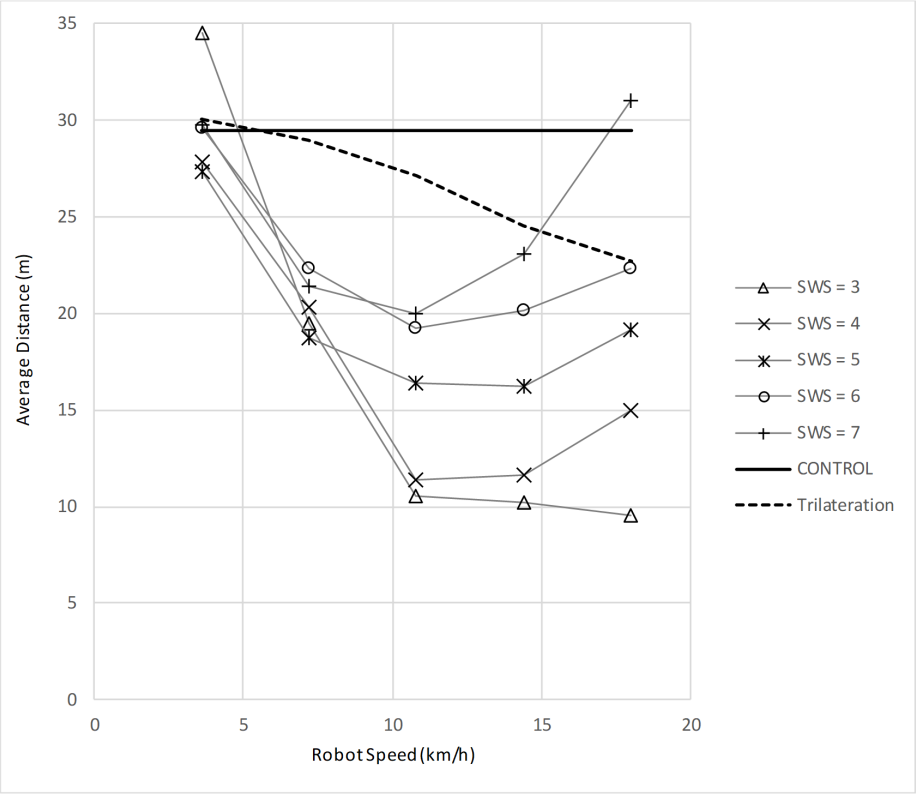


Figure 14: Average distance between robot and target versus the robot speed, for different SWS values of the Hot-Cold algorithm, the Trilateration algorithm, and the control case (standard deviation of noise due to fading (σ) is set to 2).

the Samples Window. As a result, Hot-Cold is more tolerant to random fading effects, managing to effectively turn the robot towards the moving target. Of course, when signal gets too unstable with great deviations (σ higher than 5), RSSI-based target following becomes too unreliable and quite infeasible. Most efficient following is shown to be possible for SWS values between 3 and 7.

5.2. Experimental Testbed

In order to perform real-world experiments on the introduced scheme, we have developed a testbed which enables following an individual and providing remote real-time access to her e-health data in the form of a service.

5.2.1. Robotic Device Software

In this subsection, we focus on the robotic device software, which includes the Hot-Cold algorithm. For this testbed, the robot software [36] was developed in the Eclipse Integrated Development Environment using Java and the LeJOS framework [37]. According to the system architecture, the program is executed in a Raspberry Pi that constitutes the sensor node and is mounted on the robotic device. It controls the robot over a USB connection with the robot processing unit.

Moreover, for practical reasons we have implemented in the robot an obstacle avoidance mechanism, which relies on two ultrasonic sensors positioned at the two corners (left, right) of the robot's lower front side that can detect obstacles at a distance up to 255 cm at an angular range of approximately $\pm 90°$. The concept of the respective developed algorithm is twofold: a) avoid moving towards the same obstacle repetitively, b) avoid following a direction which is almost opposite to the one already followed. For instance, if the robot meets a wall at an angle, it should always avoid it retaining the same direction and not turning back. The readings of the ultrasonic sensors are constantly checked. The corresponding steps are:

- If the readings of both sensors get lower than 25 cm, then the robot travels backwards by 10 cm and then rotates by 45°.

- Else if the reading of just the right sensor gets lower than 25 cm, then the robot travels backwards by 10 cm and then rotates by $+10^{\circ}$ (i.e. left direction).

- Else if the reading of just the left sensor gets lower than 25 cm, then the robot travels backwards by 10 cm and then rotates by -10° (i.e. right direction).

It is noticed that the specific obstacle avoidance scheme allows the robot to navigate both in wider as well as in narrower spaces. It never requires backwards movement for more than 10 cm, while it can drive the robot through a corridor as narrow as 70 cm (so that it can move in straight line between the walls without triggering the sensors). Of course, it should be clarified that the obstacle avoidance algorithm is fully configurable and replaceable, since it is not part of the main focus of this work. In fact, it is considered as part of the lower layer motion control, whereas the proposed Hot-Cold tracking algorithm operates on top of it indicating the general direction towards the target.

5.2.2. Testbed Setup

With the completion of the testbed, three experimental scenarios were set up, each one under two different radio propagation conditions. The tested case was monitoring in real-time the vital sings of a followed individual, both locally and remotely within the context of an IoT architecture. The role and properties of each entity of the testbed are described below:

- *Prototyped Sensor Platform*: The sensor platform was implemented and tested using an Arduino Uno board in combination with an e-health kit including a number of e-health sensors for estimating airflow, temperature, skin conductance, skin resistance, heart rate, SPO2, electrocardiogram, and body position. We equipped the board with a wireless network module featuring IEEE 802.15.4 [38] for broadcasting sensor data.

- *Prototyped Tracking Device*: The device following the monitored individual is a mobile robotic device, equipped with an ultrasound sensor for

obstacle avoidance.

- *Prototyped Sensor Node*: A Raspberry Pi 3 realizes the sensor node. It is equipped with an IEEE 802.15.4 module (through a conversion bridge) to communicate with the sensor platform. Moreover, it uses an IEEE 802.11 USB dongle to provide access to sensor data through relaying. Each sensor node performs MANET routing using the "Better Approach To Mobile Adhoc Networking" protocol [39] and is attached on the robotic device.

- *Prototyped Biospace/Biobot*: The EDBO Biospace was developed via JADEX as a Java program running in the relay node. The Biospace creates a Biobot providing the corresponding sensor data service, which is made discoverable to the whole Global Network of the IoT architecture through the provided service registry.

- *Prototyped End User System*: The end user system was realized as a Java-based client software running in a tablet, which is able to discover the created Biobots and access their sensor data service.

The testbed was deployed in a closed-space sports University facility of dimensions 35m x 40m (totaling 1400 m^2), where the Cartesian axes origin (0, 0) is placed at the top left corner. A robotic device equipped with a sensor node moves at 10 km/h and performs Hot-Cold tracking (with parameters $SWS = 4$ and $\phi = 137°$) of a person carrying a sensor platform, who walks at 5 km/h when moving. The edge-to-edge front wheel distance (i.e. the track width) of the robotic device is 17.5 cm and the duration of a direction change (stop-rotate-start) is on average 2.5 sec. The sensor data are forwarded over an IEEE 802.11 MANET through a Raspberry Pi relay node to a laptop that constitutes the end user system. Four iterations of 60 sec are performed for each experimental scenario, which are set up as follows:

- *1st Scenario*: This is considered as control scenario, where the target remains static at position (5, 5). The robot's starting position is (30, 35) facing away from the target (direction at 50°).

- *2nd Scenario*: The target moves in straight line starting at point (5, 5) and in 28 sec reaches the destination point (50, 35). The robot's starting position is (30, 5) with initial direction 0°.

- *3rd Scenario*: The target moves in zigzag starting at point (5, 5), visiting after 10 sec the first waypoint (5, 11.5), 20 sec later it reaches the second waypoint (30, 25), and 20.5 sec later it reaches the destination (5, 35). The robot's starting position is (30, 5) with initial direction 0°.

Two experimental evaluations took place for each one of the three scenarios, corresponding to different radio propagation conditions. In detail, the first experiment was performed under normal noise conditions, with no interference intentionally created in the testbed environment. The second experiment performed for each scenario took place under elevated noise conditions using two sources of interference: a pair of Raspberry Pi devices positioned at coordinates (0,0) and (35,40), respectively. These two devices were equipped with a 5 dBi omni-directional antenna and were configured to exchange data over an IEEE 802.11n link operating at channel 6, which has a central frequency of 2437 MHz and bandwidth of 20 MHz, and transmit at 13 dBm TX power. It is noted that the specific communication directly interferes with the IEEE 802.15.4 link connecting the target to the robotic device, which is set at frequency 2435 MHz that corresponds to channel 17 of the specific standard.

5.2.3. Testing Results

The experimental results of the robot-target distance for the 1st scenario are presented in Figure 15, along with an illustration of the scenario setup. In Figure 15a, it can be seen that in normal noise conditions, after correcting its direction, the robot manages to reach the static target within the experiment duration. Specifically, in three iterations the target is reached in about 20 sec, while in one iteration the target is reached in about 37 sec. Figure 15b shows that under elevated noise conditions, the robot has to correct its direction several times, due to some erroneous estimates of the signal strength variations caused

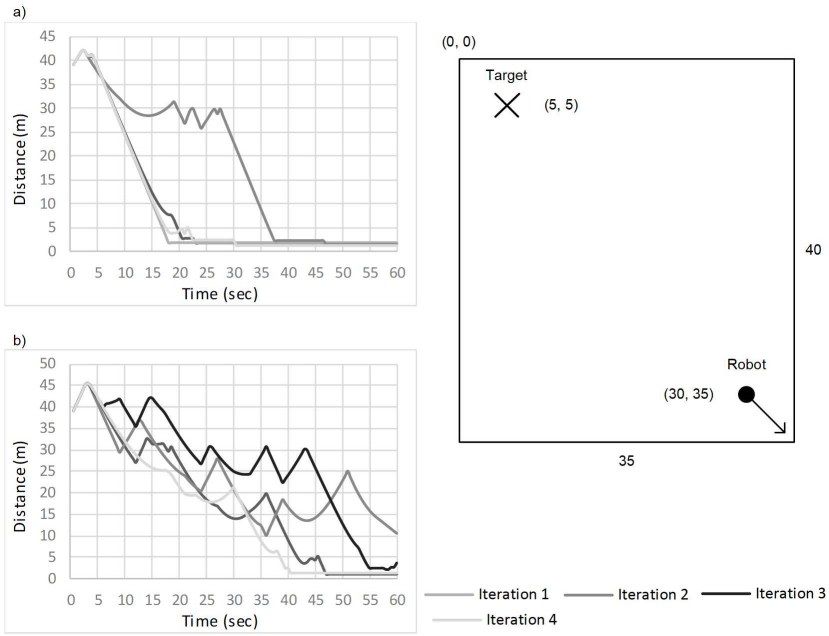


Figure 15: Distance between target and robot adopting Hot-Cold tracking versus time in the 1st scenario, under a) normal noise and b) elevated noise conditions

by the induced interference, however, it still manages to closely approach the target within the duration of the experiment.

In the 2nd scenario, the charts in Figure 16 also present target's distance from the robot's starting point (dashed line). It is evident that under normal noise conditions (Figure 16a) the robot initially corrects its direction and reaches the target about 2 sec after the latter arrives at destination (at time 28 sec), except from one iteration that required sixteen more seconds. Under elevated noise conditions, it can be seen that the robot has to realign its trajectory several times, but eventually manages to stay in the target's proximity, as show in Figure 16b.

Probably the most challenging scenario is the third one, with the corresponding results for normal noise conditions presented in Figure 17a. Until the first waypoint, the target moves almost opposite from the robot's initial direction, causing temporal distance increment. After that, the robot approaches, reach-

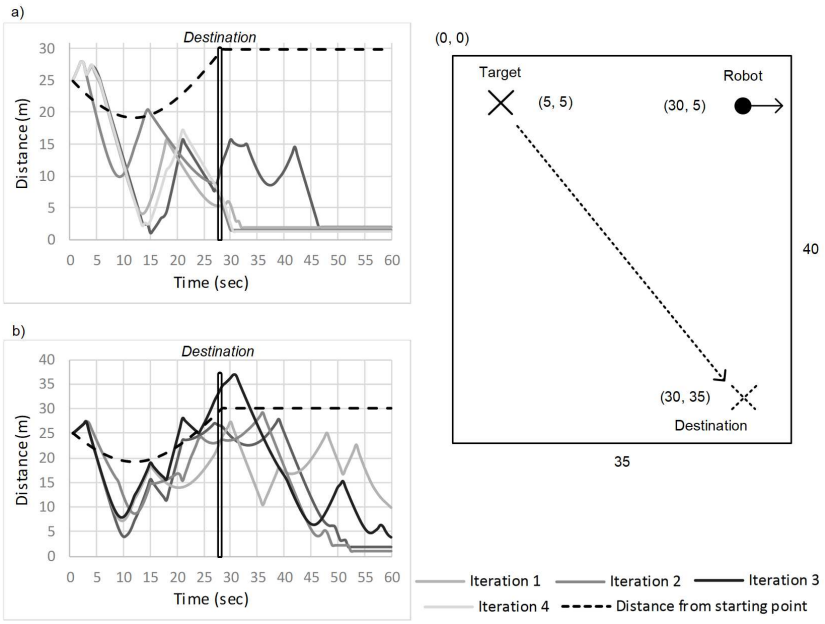


Figure 16: Distance between target and robot adopting Hot-Cold tracking versus time in the 2nd scenario, under a) normal noise and b) elevated noise conditions

ing minimum distance right after the target's arrival at the second waypoint. Then, the robot manages in all iterations to maintain proximity, however, the 9.5 sec that the target remains static at the destination point is not sufficient time for the robot to stay stably close to the target. A similar behaviour is also observed under elevated noise conditions, as presented in Figure 17b. It is evident that in the presence of excessive interference, it is quite difficult for the robot to stay close to the target when the latter reaches the second waypoint, however, it eventually manages to approach closely by the end of the experiment.

Conclusively, in all three scenarios, the Hot-Cold algorithm has succeeded in its goal of adjusting robot's trajectory to keep approaching the target, affected of course by the mobility pattern. Artificially increasing the radio interference led to degradation of the tracking performance (longer intervals before reaching the target), however, the robot managed to stay in target's close proximity.

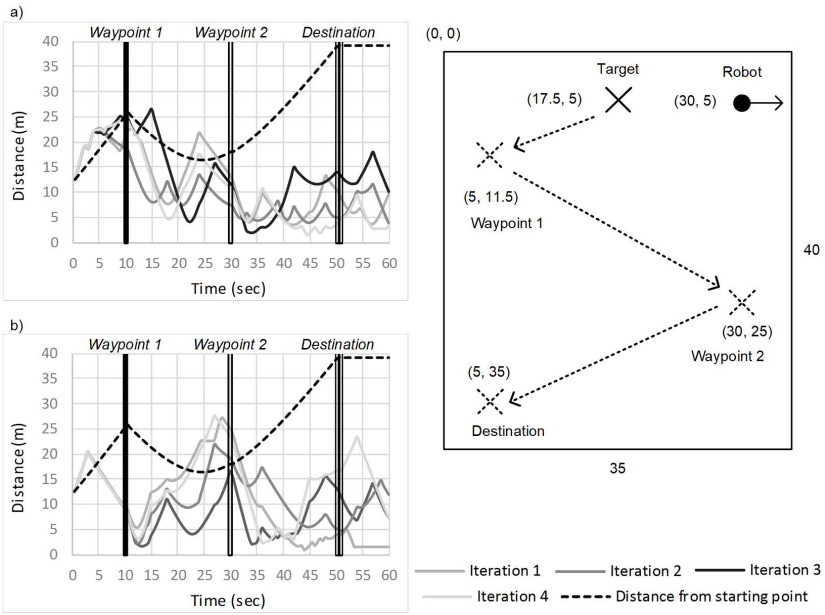


Figure 17: Distance between target and robot adopting Hot-Cold tracking versus time in the 3rd scenario, under a) normal noise and b) elevated noise conditions

During the experiments, end-to-end connectivity to the sensor platform was maintained, with all sensor data collected every 0.5 sec successfully relayed to the end user system.

In general, the tracking environment may affect the behavior of the Hot-Cold scheme in various ways. The structure of the considered scene has a direct impact on the RF signal propagation as well as on the target's mobility pattern and the robot's following abilities. Specifically, the presence of multiple obstacles (such as in an urban environment) can create high shadowing and multipath fading effects which lead to less reliable RSSI estimates, hence, to less accurate rotation decisions. On the other hand, a relatively open scene ensures weaker fading, thus, more efficient tracking decisions based on the received signal strength fluctuations. Moreover, heavily obstructed paths make it difficult to maintain a consistent tracking course and put most of the pressure on the adopted obstacle avoidance technique, which however is not an internal

component of the introduced Hot-Cold algorithm, but a complementary one. Generally, as expected, all elements that may affect signal propagation (such as objects' reflection and refraction factors) and/or mobility have an impact on the efficiency of the introduced technique. The presented evaluation results have revealed that in environments where the tracking robot can effectively move via obstacle avoidance and the standard deviation of the experienced Gaussian fading is lower than 5, the proposed scheme can ensure effective tracking performance.

6. Conclusions and Future Directions

In this paper, we have primarily introduced a new algorithm for following mobile monitored targets/individuals in the context of an IoT system. The devised technique, called Hot-Cold, is able to ensure proximity maintenance by the tracking robotic device solely based on the strength of the RF signal broadcasted by the target to communicate its sensors' data. Possible applications of such a tracking technique are quite promising and include the sustainment of communication links for monitoring purposes in dynamic environments with limited or unavailable network infrastructure. The monitoring information is made available over a bio-inspired IoT architecture, which allows flexible creation and discovery of sensor-based services.

For the identification of the optimal rotation angle employed by the tracking robot, a complete analysis was conducted in four steps: geometrical analysis, numerical analysis, exhaustive-simulation analysis, and convergence analysis. The analytical results reveal that performance optimization is achieved for Hot-Cold at a rotation angle of $\sim$137 degrees. An in-depth evaluation of the proposed technique was performed through simulations and in comparison with the well-known concept of trilateration-based tracking. The simulation results have identified the optimal configuration for Hot-Cold key parameters and have shown that it achieves superior performance for realistic levels of signal fading due to shadowing. The evaluation part is completed with the presentation

of a testbed, which demonstrates the proposed IoT system concept. All key components are thoroughly described, focusing on the target following aspect. The conducted experiments show the operability of the overall approach and especially focus on the effectiveness of the tracking technique.

Future work involves the optimization of the tracking technique for generalized target following scenarios in the context of IoT. For instance, we intend to investigate combinations of different numbers of tracking devices following one or more monitored targets in a cooperative manner. Moreover, possible applications can be extended from mobile tracking robots to Unmanned Autonomous Vehicles (UAVs - drones). In general, the related potential extensions in terms of candidate applications and functionality enhancements are numerous and very promising for the future of IoT and they definitely worth further exploration.

References

[1] G. Eleftherakis, D. Pappas, T. Lagkas, K. Rousis, O. Paunovski, Architecting the IoT Paradigm: A Middleware for Autonomous Distributed Sensor Networks, International Journal of Distributed Sensor Networks 11 (12) (2015) 139735. doi:10.1155/2015/139735.
URL http://journals.sagepub.com/doi/10.1155/2015/139735

[2] P. Ray, A survey on Internet of Things architectures, Journal of King Saud University - Computer and Information Sciencesdoi:10.1016/j.jksuci.2016.10.003.
URL http://linkinghub.elsevier.com/retrieve/pii/S1319157816300799

[3] A. Küpper, Location-based services: fundamentals and operation, John Wiley, Chichester, England ; Hoboken, NJ, 2005, oCLC: ocm60742115.

[4] R. Siegwart, I. R. Nourbakhsh, Introduction to Autonomous Mobile Robots, Bradford Company, Scituate, MA, USA, 2004.

[5] J. Hightower, G. Borriello, A survey and taxonomy of location systems for ubiquitous computing, IEEE computer 34 (8) (2001) 57–66.
URL http://www.csd.uoc.gr/~hy439/lectures11/hightower2001survey.pdf

[6] Y. Zhuang, Z. Shen, Z. Syed, J. Georgy, H. Syed, N. El-Sheimy, Autonomous WLAN heading and position for smartphones, in: 2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014, 2014, pp. 1113–1121. doi:10.1109/PLANS.2014.6851481.

[7] Y. Zhuang, Z. Syed, Y. Li, N. El-Sheimy, Evaluation of Two WiFi Positioning Systems Based on Autonomous Crowdsourcing of Handheld Devices for Indoor Navigation, IEEE Transactions on Mobile Computing 15 (8) (2016) 1982–1995. doi:10.1109/TMC.2015.2451641.

[8] B. Fidan, S. Dasgupta, B. Anderson, Adaptive range-measurement-based target pursuit, International Journal of Adaptive Control and Signal Processing 27 (1-2) (2013) 66–81.
URL http://onlinelibrary.wiley.com/doi/10.1002/acs.2353/full

[9] S. D. Bopardikar, F. Bullo, J. P. Hespanha, A pursuit game with range-only measurements, in: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, IEEE, 2008, pp. 4233–4238.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4738582

[10] O. Namaki-Shoushtari, A. P. Aguiar, A. K. Sedigh, A switched based control strategy for target tracking of autonomous robotic vehicles using range-only measurements, IFAC Proceedings Volumes 44 (1) (2011) 12819–12824.
URL http://www.sciencedirect.com/science/article/pii/S147466701645679X

[11] G. Chaudhary, A. Sinha, Capturing a target with range only measurement, in: Control Conference (ECC), 2013 European, IEEE, 2013, pp.

4400–4405.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6669213

[12] Y. Cao, J. Muse, D. Casbeer, D. Kingston, Circumnavigation of an unknown target using UAVs with range and range rate measurements, in: 52nd IEEE Conference on Decision and Control, IEEE, 2013, pp. 3617–3622.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6760439

[13] H. Teimoori, A. V. Savkin, Equiangular navigation and guidance of a wheeled mobile robot based on range-only measurements, Robotics and Autonomous Systems 58 (2) (2010) 203–215.
URL http://www.sciencedirect.com/science/article/pii/S0921889009001420

[14] S. Gezici, I. Guvenc, Z. Sahinoglu, On the Performance of Linear Least-Squares Estimation in Wireless Positioning Systems, in: 2008 IEEE International Conference on Communications, IEEE, Beijing, China, 2008, pp. 4203–4208. doi:10.1109/ICC.2008.789.
URL http://ieeexplore.ieee.org/document/4533825/

[15] G. P. Huang, S. I. Roumeliotis, Analytically-guided-sampling particle filter applied to range-only target tracking, in: 2013 IEEE International Conference on Robotics and Automation, IEEE, Karlsruhe, Germany, 2013, pp. 3168–3175. doi:10.1109/ICRA.2013.6631018.
URL http://ieeexplore.ieee.org/document/6631018/

[16] R. Olfati-Saber, N. F. Sandell, Distributed tracking in sensor networks with limited sensing range, in: 2008 American Control Conference, IEEE, Seattle, WA, 2008, pp. 3157–3162. doi:10.1109/ACC.2008.4586978.
URL http://ieeexplore.ieee.org/document/4586978/

[17] H. M. La, W. Sheng, Dynamic target tracking and observing in a mobile sensor network, Robotics and Autonomous Systems 60 (7) (2012) 996–1009. doi:10.1016/j.robot.2012.03.006.
URL http://linkinghub.elsevier.com/retrieve/pii/S0921889012000565

[18] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, E. Aboutanios, Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications, IEEE Communications Surveys & Tutorials 19 (2) (2017) 1327–1346. doi:10.1109/COMST.2016.2632427.
URL http://ieeexplore.ieee.org/document/7762095/

[19] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. Al-Ammar, H. Al-Khalifa, Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances, Sensors 16 (5) (2016) 707. doi:10.3390/s16050707.
URL http://www.mdpi.com/1424-8220/16/5/707

[20] S. Gezici, Zhi Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, Z. Sahinoglu, Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks, IEEE Signal Processing Magazine 22 (4) (2005) 70–84. doi:10.1109/MSP.2005.1458289.
URL http://ieeexplore.ieee.org/document/1458289/

[21] J. Liang, Q. Liang, RF Emitter Location Using a Network of Small Unmanned Aerial Vehicles (SUAVs), in: 2011 IEEE International Conference on Communications (ICC), 2011, pp. 1–6, iSSN: 1550-3607. doi:10.1109/icc.2011.5962487.

[22] A. Albert, L. Imsland, Combined Optimal Control and Combinatorial Optimization for Searching and Tracking Using an Unmanned Aerial Vehicle, Journal of Intelligent & Robotic Systems 95 (2) (2019) 691–706. doi:10.1007/s10846-018-0915-4.
URL https://doi.org/10.1007/s10846-018-0915-4

[23] Z. Liu, X. Fu, X. Gao, Co-Optimization of Communication and Sensing for Multiple Unmanned Aerial Vehicles in Cooperative Target Tracking, Applied Sciences 8 (6) (2018) 899, number: 6 Publisher: Multidisciplinary Digital Publishing Institute. doi:10.3390/app8060899.
URL https://www.mdpi.com/2076-3417/8/6/899

[24] F. Koohifar, A. Kumbhar, I. Guvenc, Receding Horizon Multi-UAV Cooperative Tracking of Moving RF Source, IEEE Communications Letters 21 (6) (2017) 1433–1436, conference Name: IEEE Communications Letters. doi:10.1109/LCOMM.2016.2603977.

[25] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, T. D. Murphey, Real-Time Area Coverage and Target Localization Using Receding-Horizon Ergodic Exploration, IEEE Transactions on Robotics 34 (1) (2018) 62–80, conference Name: IEEE Transactions on Robotics. doi:10.1109/TRO.2017.2766265.

[26] G. Eleftherakis, O. Paunovski, K. Rousis, A. J. Cowling, Emergent Distributed Bio-organization: A Framework for Achieving Emergent Properties in Unstructured Distributed Systems, in: G. Fortino, C. Badica, M. Malgeri, R. Unland (Eds.), Intelligent Distributed Computing VI, Vol. 446, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 23–28, dOI: 10.1007/978-3-642-32524-3_5.
URL http://link.springer.com/10.1007/978-3-642-32524-3_5

[27] L. Braubach, A. Pokahr, Developing Distributed Systems with Active Components and Jadex, Scalable Computing: Practice and Experience 13 (2) (2012) 100–120. doi:10.12694/scpe.v13i2.773.
URL http://www.scpe.org/index.php/scpe/article/view/773

[28] J. Broekens, M. Heerink, H. Rosendal, Assistive social robots in elderly care: a review, Gerontechnology 8 (2). doi:10.4017/gt.2009.08.02.002.00.

URL http://gerontechnology.info/index.php/journal/article/view/1011

[29] T. Lagkas, Hot-cold simulation code, original-date: 2020-03-15T16:55:36Z (Mar. 2020).
URL https://github.com/tlagkas/hot-cold_simulation

[30] B. Fry, Visualizing Data: Exploring and Explaining Data with the Processing Environment, "O'Reilly Media, Inc.", 2007, google-Books-ID: RRswXg4pJhcC.

[31] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, R. Bianchi, An empirically based path loss model for wireless channels in suburban environments, IEEE Journal on Selected Areas in Communications 17 (7) (1999) 1205–1211. doi:10.1109/49.778178.

[32] T. Rappaport, Wireless Communications: Principles and Practice, 2nd Edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[33] E. J. Meinilä, Ist-2003-507581 winner i, d5. 4, final report on link level and system level channel models, Tech. rep. (2005).

[34] P. Kyösti, J. Meinilä, L. Hentilä, X. Zhao, T. Jämsä, C. Schneider, M. Narandzić, M. Milojević, A. Hong, J. Ylitalo, et al., Winner ii channel models d1. 1.2 v1. 1, european commission, deliverable ist-winner d, Tech. rep., IST-4-027756 WINNER II.

[35] E. Wan, Sigma-point filters: An overview with applications to integrated navigation and vision assisted control, in: 2006 IEEE Nonlinear Statistical Signal Processing Workshop, 2006, pp. 201–202. doi:10.1109/NSSPW.2006.4378854.

[36] T. Lagkas, Hot-cold prototype code, original-date: 2020-03-22T10:39:27Z (Mar. 2020).
URL https://github.com/tlagkas/hot-cold_prototype

[37] leJOS NXJ Plug-In.
URL https://marketplace.eclipse.org/content/lejos-nxj-plug

[38] IEEE Standard for Low-Rate Wireless Networks, IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011) (2016) 1–709doi:10.1109/IEEESTD.2016.7460875.

[39] D. Johnson, N. Ntlatlapa, C. Aichele, A simple pragmatic approach to mesh routing using batman, in: In 2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, Pretoria, South Africa, 2008.