IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 11, 2019, accepted February 23, 2019, date of publication March 11, 2019, date of current version March 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904241

Algorithms for Estimating the Location of Remote Nodes Using Smartphones

DÁRIO PEDRO[1], SLAVISA TOMIC[2], LUIS BERNARDO[1,3], (Member, IEEE), MARKO BEKO[2,4], RODOLFO OLIVEIRA[1,3], (Senior Member, IEEE), RUI DINIS[1,3], (Senior Member, IEEE), PAULO PINTO[1,3], (Member, IEEE), AND P. AMARAL[1,3], (Member, IEEE)

[1]Instituto de Telecomunicações, 1049-001 Lisbon, Portugal
[2]COPELABS/ULHT, 1749-024 Lisbon, Portugal
[3]Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[4]UNINOVA, 2829-516 Caparica, Portugal

Corresponding author: Luis Bernardo (lflb@fct.unl.pt)

This work was supported in part by the Project CoSHARE (LISBOA-01-0145-FEDER-0307095 - PTDC/EEI-TEL/30709/2017), Project UID/EEA/50008/2019, Project UID/MULTI/04111/0216, Project foRESTER (PCIF/SSI/0102/2017), and Program Investigador FCT under Grant IF/00325/2015, in part by the Fundo Europeu de Desenvolvimento Regional (FEDER), through Programa Operacional Regional LISBOA under Grant LISBOA2020, and in part by the national funds, through Fundação para a Ciência e Tecnologia (FCT).

ABSTRACT Locating the position of a remote node on a wireless network is becoming more relevant, as we move forward in the Internet of things and in autonomous vehicles. This paper proposes a new system to implement the location of remote nodes. A new prototype Android application has been developed to collect real measurements and to study the performance of several smartphone's sensors and location algorithms, including an innovative one, based on the second order cone programming (SOCP) relaxation. The application collects the WiFi access points information and the terminal location. An internal odometry module developed for the prototype is used when Android's service is unavailable. This paper compares the performance of existing location estimators given in closed form, an existing SOCP one, and the new SOCP location estimator proposed, which has reduced complexity. An algorithm to merge measurements from non-identical terminals is also proposed. Cooperative and terminal stand-alone operations are compared, showing a higher performance for SOCP-based ones, that are capable of estimating the path loss exponent and the transmission power. The heterogeneous terminals were also used in the tests. Our results show that the accurate positioning of static remote entities can be achieved using a single smartphone. On the other hand, the accurate real-time positioning of the mobile terminal is provided when three or more scattered terminal nodes cooperate sharing the samples taken synchronously.

INDEX TERMS Indoor location, odometry, localizing remote nodes without known anchors, location algorithms, cooperative localization, android applications.

I. INTRODUCTION

Locating the position of a remote network element on a wireless network is becoming more relevant as these networks are interconnecting hand-held terminals, vehicles with different degrees of self driving-autonomy and network infrastructure elements. In order to achieve it, one node needs to know its own location and use the signals from remote nodes. However, providing an accurate and practical localization solution anywhere with portable devices is still a challenge in environments where global navigation satellite systems (GNSS) signals are not available or are degraded, and where alternative indoor localization systems are not deployed. On the other end, although a remote node may provide its location, sometimes it is important to validate the information received.

This paper addresses the challenge of providing a portable localization system that locates remote nodes without requiring any known position. It uses the smartphone's sensors and Android's services to estimate its own location, and the WiFi signals received to obtain the location of the remote nodes.

Fingerprinting and multilateration are two main approaches for WiFi positioning [1]. Fingerprinting-based WiFi positioning usually has two operating phases: the pre-survey phase and the online positioning phase. In the pre-survey phase, a radio map database is created with a set of known positions and the Received Signal Strength (RSS) values

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqiang Wang.

from available Access Points (APs) are collected. The radio map database is used during the online positioning phase to compare the observed RSS values with the stored ones in order to choose the best fit. Heuristic strategies such as [2] can be used for selecting a subset of the known APs that maximizes accuracy for non-cooperative and crowdsourcing based cooperative approaches. Multilateration-based WiFi positioning calculates the ranges between the device and APs using a wireless signal propagation model and trilateration or multilateration algorithms, such as the ones in [3] and [4], to estimate the position of the nodes. Fingerprinting usually provides more accurate position solutions for the node's own location [1]. On the other hand, multilateration may be used for locating itself and remote nodes. It can be used with only the information about the position of the APs, or even without that information. For instance, in [5], the RSS measurements and any border GNSS location information are sent to a central server and processed there using a genetic algorithm that models the physical constraints to estimate the nodes location.

Several signals can be explored to enhance or replace the GNSS localization service [6], including the cellular network in Assisted-GPS (A-GPS) [7], WiFi signals (e.g. [8]), Angle of Arrival (AoA) (e.g. [9], [10]) and the micro-electro-mechanical systems (MEMS) sensors present in portable devices (e.g. [1]). A good survey about smartphone based localization techniques is presented in [11]. MEMS sensors can be used to implement inertial navigation system (INS) and/or pedestrian dead reckoning (PDR) (e.g. [1], [12]) for vehicle or pedestrian navigation applications. The INS mechanization calculates the position, velocity and altitude by integrating raw data from accelerometers and gyroscopes. However, navigation errors increase rapidly with time due to the drift characteristics of MEMS sensors. PDR is proposed to reduce the accumulated navigation errors for pedestrians. It has four critical procedures: step detection [13], step/stride length estimation, heading estimation, and 2D position estimation. PDR provides a more accurate position solution than INS, without other aiding sources [1]. Nevertheless, its accuracy depends on other sensors, such as the gyroscope, and on estimated parameters. Therefore, PDR/INS are usually combined with other estimation models, and they may be switched on or off using algorithms that detect the user context (e.g. indoor or outdoor) [14], [15]. WiFi fingerprinting was used with PDR in HiMLoc [16] and WiFi multilateration was used with PDR/INS [1] to estimate the device location, requiring both some *a priori* knowledge of the site.

Simultaneous Localization and Mapping (SLAM) algorithms address the localization problem assuming no prior site information. Most SLAM solutions also combine PDR/INS and fingerprinting, although using more elaborate robotic sensors and mobility scenarios [17]. SLAM is a nonconvex problem and most SLAM algorithms are based on iterative nonlinear optimization. State-of-the-art iterative solvers fail to converge to a global minimum of the cost function for relatively small noise levels; hence, methods that explore the nonconvex nature in SLAM and alternative maximum likelihood (ML) formulations were proposed recently. These include convex relaxation methods, such as semidefinitive programming (SDP) [17]. Other authors proposed equally efficient and less complex convex relaxation solutions [18]. Although different, the evolution of multilateration algorithms followed a similar path, and some of the best performing algorithms are also based on convex relaxation solutions (e.g. [4], [10]). In spite of convergence to global minima is guaranteed with such (convex) methods, it does not necessarily (closely) correspond to the global minima of the original nonconvex problem (the quality of the solution depends on the tightness of the relaxations).

In this paper, the smartphone's sensors and Android's services are used to implement a PDR/GNSS terminal self-location solution. The received WiFi signals are processed using multilateration algorithms to locate remote nodes. Several localization algorithms are analyzed using RSS samples collected by a smartphone in indoor scenarios, considering cooperative and non-cooperative scenarios. This paper extends [19], which analyzed an outdoor scenario using GNSS and a non-cooperative scenario only. The paper's main contributions include:

- an analysis and performance evaluation of multilateration algorithms, comparing classical static ones, ML formulations, convex SDP and convex relaxation solutions using RSS data collected in pedestrian tests;
- development of a new convex relaxation solution for the multilateration problem;
- design of an algorithm that combines measurements from a heterogeneous set of terminals in multilateration, and the evaluation of its performance in a cooperative scenario;
- performance evaluation of the accuracy of different PDR/INS mechanisms in two different terminals.

This paper is organized as follows: the system overview is presented in section II. Section III analyses the performance of the Android PDR/INS odometry modules. Multilateration algorithms are reviewed and a new algorithm is proposed in section IV. A cooperation algorithm is presented in section V. Section VI evaluates the performance of the algorithms considered for odometry, and non-cooperative and cooperative localization, using RSS experimental values. Section VII summarizes the main conclusions.

II. SYSTEM ARCHITECTURE

The system is designed to track the location of entities possessing a WiFi interface using a mobile terminal (MT). The approach followed is illustrated in figure 1. A moving terminal collects a set of RSS measurements of the signals received from the tracked entities on multiple locations, registering the MT's location at each measurement. These RSS measurement vectors are used to run multilateration algorithms, which require a minimum of three or four different measurements to estimate a location. Given that all RSS measurements should be done with the entities at the same position, the measuring

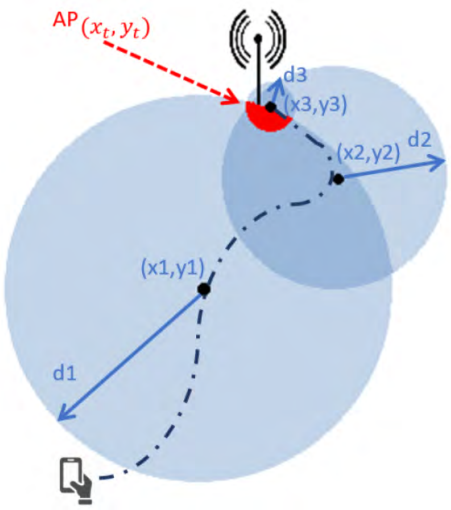


FIGURE 1. Localization scheme.

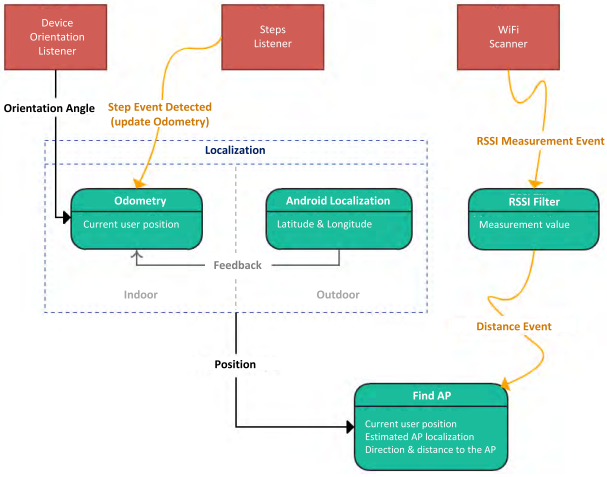


FIGURE 2. Terminal operation scheme.

MT should move much faster than the entities being tracked if a non-cooperative mode is used. When multiple MTs cooperate, the individual measurements can be combined and shared between the MTs, reducing or eliminating the necessity for MT movement. On the other hand, homogenization techniques are required to handle the measurements taken from different MTs that have different sensors, radios and antennas.

The main component of the architecture is the software running in the MT, which implements the modules depicted in figure 2 [19].

The localization module is capable of switching between outdoor mode and indoor mode. In outdoor mode, it uses the Android localization service, which combines GNSS, WiFi and cellular signals to estimate the MT's location. An odometry module is used in indoor mode, to track the relative position of the MT from the last known global position. The current version is optimized for human use - it implements

[1]The app source code can be downloaded from `https://github.com/dario-pedro/wifi_finder`.

FIGURE 3. Find AP interface.

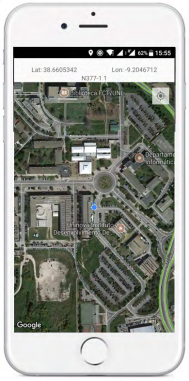


FIGURE 4. Map interface.

the PDR algorithms described in section III. An WiFi scanner module is run concurrently, which collects the RSS vectors of a selected AP using the *android.net.wifi* Android's library. In the non-cooperative mode, the vectors are fed to the multilateration algorithm, implemented at the FindAP module, which returns the location of the selected remote entity. In the cooperative mode, the RSS vectors are sent to a server, which stores them, and may run the localization algorithms over all received vectors. The main goal of this paper is to propose effective algorithms for both operation modes. A prototype Android application was implemented for the non-cooperative mode [19][1]. The current prototype application provides two visualization mechanisms for the remote entities' location: it can provide an arrow pointing to the direction towards the entity (fig. 3); or, it can provide the location of the entities in a map (fig. 4). The application can be easily extended to other scenarios that do not involve human users: the FindAP module can send the entities' locations directly to an external module (e.g. autonomous driving unit inside a vehicle) through a specific API.

For testing purposes, the application was extended to collect AP's RSS measurements, time, odometry/Android location, allowing the evaluation of multiple algorithms with real measured data, such as the ones presented in this paper.

III. ODOMETRY MODULE

The odometry module implements a 2D location estimator based on a PDR. It counts steps, measures the MT orientation,

and estimates the new location using an average step size estimative. The odometry module is composed of two sub-modules: a step detector and a direction detector.

The step detector was implemented based on the design presented in [20], using the accelerometer sensor scanned with a 320Hz sampling rate. It analyses the sequence of extreme values of the linear combination of the 3D acceleration components, $A_d = \sum_s F_s$, looking for local minima, $p_{\min}$, and local maxima, $p_{\max}$. The algorithm compares the last difference of extreme values $p_{\text{diff}} = |p_{\max} - p_{\min}|$ and the value of p_{diff} measured in the previous interval (denoted by u_p). A new step is detected during a local extreme when the following conditions are verified: $2/3 < p_{\text{diff}}/u_p < 3$, the acceleration direction is inverted in relation to the last peak detected, and at least $t_{\min}$ ms elapsed since the last step detected. Additionally, for comparison purposes, two additional Android step detection objects were tested: the Step Counter Sensor (SCS) and the Step Detector Sensor (SDS) APIs [21]. SCS and SDS implement Kalman filters, to identify the step patterns in the accelerometer. Although they are potentially more precise, they require a training period to calibrate the filters, which introduces an initial delay when the services start (it needs an average of 27 steps). On the other hand, after the training phase is completed, SCS and SDS converge to almost the same number of steps, but take longer to detect a step. This is critical in our implementation, given that the orientation may be changing. Therefore, all results presented in this paper used the first detector presented, based exclusively on the accelerometer sensor.

A relevant contribution of this paper is the evaluation of the accuracy of different direction sensor devices available in the Android operating system in two different MTs. Multiple direction detection sub-modules were implemented and tested, using the Android Motion Sensors API [21]. This API offers a set of software modules that interface the physical sensors in the MTs. Direction was measured using the Accelerometer, MagneticField (compass), Gyroscope and Rotation Vector (RV) sensors, and fusions of signals from multiple sensors using the [22] library. In section VI-A we compare the performance of the fusions of: accelerometer and compass; gravity and compass; gyroscope and RV; and the isolated sensors calibrated gyroscope and RV sensors. A user defined average step size is used in the application prototype, and the tests applied a known fixed step size to avoid introducing step-size related errors in the measured results.

IV. LOCALIZATION ALGORITHMS

The problem of localizing a remote node is dual to the classical localization problem. Let t denote the target with unknown coordinates and s_i, $i = 1, ..., N$, the set of known positions where the MT collected the target's RSS values, P_i^r. The distance corresponding to a given RSS value P^r can be obtained using the Friis equation,

$$d = 10^{\frac{\alpha - P^r - 10\gamma \log_{10}(f)}{10\gamma}}, \tag{1}$$

where γ represents the path loss exponent (PLE), f the carrier frequency and α the contributions of the antennas gains and the transmission power (assumed uniform) [23]. The measured distance is defined by

$$d_i = \|s_i - t\| + \varepsilon_i, \tag{2}$$

where ε_i is the error in ith measurement that includes contributions from the MT location error and due to errors in the estimated PLE value. Errors are assumed independent. Considering a vectorial notation, (2) can be generalized to

$$D = h(t) + \varepsilon, \tag{3}$$

where D denotes the measurement vector, h the vector-valued measurement function and ε the error vector. The objective of a positioning method is to find the position t that minimizes the residual between the true distances and the measured ones, which defines a multilateration problem. One way for obtaining an estimate of the target's location, t^*, is via the least square (LS) criterion, i.e,

$$t^* = \arg\min_t \|D - h(t)\|^2 = \arg\min_t \sum_{i=1}^{N} (d_i - h_i(t))^2 . \tag{4}$$

The following sections review several closed form estimators [3] for overdetermined systems. They also review an estimator based on a convex relaxation solution [4] and propose a new one with reduced complexity in section IV-H. Note that convex-based estimators always converge to the global optima, but have a higher overhead.

A. SIMPLE INTERSECTION

The simple intersection method is derived by expanding the squared range equations to $2s_i^T t = \|t\|^2 + \|s_i\|^2 - d_i^2$, and subtracting a "reference" one from all the other ones, which produces a system with $N - 1$ linear equations. The final solution (for more details please see [3]) is given by,

$$\widehat{t} = (A^T C^T C A)^{-1} A^T C^T C B, \tag{5}$$

where

$$A = \begin{bmatrix} 2s_1^T \\ \vdots \\ 2s_N^T \end{bmatrix}, \quad B = \begin{bmatrix} \|s_1\|^2 - d_1^2 \\ \vdots \\ \|s_N\|^2 - d_N^2 \end{bmatrix}, \quad C = \begin{bmatrix} -1_{(N-1)\times 1} \; I_{N-1} \end{bmatrix}.$$

B. RANGE-BANCROFT

Bancroft's method starting point is also based on an expansion of the squared range equations presented above, for the simple intersection, which can be written using matrices as $At = \mathbf{1}\|t\|^2 + B$, where $\mathbf{1}$ denotes a vector with all entries equal to one. This equation can be solved by multiplying both sides with the Moore-Penrose pseudo-inverse of A and taking the square norm. The solution candidates, x_i, are obtained from the two roots of (6) [3], $x_i = \|t\|^2$, $i = 1, 2$,

$$\|p\|^2 \|t\|^4 + (2p^T q - 1)\|t\|^2 + \|q\|^2 = 0, \tag{6}$$

where $p = (A^TA)^{-1}A^T\mathbf{1}$ and $q = (A^TA)^{-1}A^TB$. The solution [3], is the value $\widehat{t} = px_i + q$ for the i that gives the smallest residual value $\sum_{i=1}^{N}\left(d_i - \left\|s_i - \widehat{t}\right\|\right)^2$.

C. BECK

Beck *et al.* [24] defined a procedure for computing the exact least-quartic solution for the range equations. The procedure uses the bisection method to find a root of a univariate strictly monotonous function on an interval that is easily computed, so it has guaranteed convergence. Consider the vector B defined above, $\lambda = \|t\|^2$ and

$$E = \begin{bmatrix} 2s_1^T & -1 \\ \vdots & \vdots \\ 2s_N^T & -1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad w = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2} \end{bmatrix}.$$

The *exact* solution, $\widehat{z}$, is obtained as

$$\widehat{z}(\lambda) = (E^TE + \lambda P)^{-1}(E^TB - \lambda w)\,, \tag{7}$$

where the bisection procedure is used to find λ^*, the zero of

$$\phi(\lambda) = \widehat{z}(\lambda)^T P\widehat{z}(\lambda) + 2w^T\widehat{z}(\lambda). \tag{8}$$

The estimated position is given by the first j components of $\widehat{z}(\lambda^*)$, where j represents the dimension of the considered space.

D. CHEUNG

Cheung *et al.* [25] provided a constrained weighted least squares solution for range measurements. The unknowns to be solved are the position and squared distance to first position, s_1. They assume that each measurement error ε_i is a zero-mean white Gaussian process with known variance σ_i^2, which is estimated using the sample values.

Consider the matrices E, B, P and w defined above,

$$F = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_N \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_N^2 \end{bmatrix}, \quad \Psi = F\Sigma F.$$

Computing U and Λ from the eigenvalue decomposition

$$\left(E^T\Psi^{-1}E\right)^{-1}P = U\begin{bmatrix} \gamma_1 & & \\ & \gamma_2 & \\ & & 0 \end{bmatrix}U^{-1} \tag{9}$$

yields $c = 2U^Tw$, $g = 2U^T(E^T\Psi^{-1}E)^{-1}w$, $e = (\Psi^{-1}EU)^TB$ and $f = U^{-1}(E^T\Psi^{-1}E)^{-1}E\Psi^{-1}B$.

The root λ^* closest to zero is calculated for the following five-root equation,

$$\begin{aligned} c_3f_3 - \frac{\lambda}{2}c_3g_3 + \sum_{i=1}^{2}\frac{c_if_i}{1+\lambda\gamma_i} - \frac{\lambda}{2}\sum_{i=1}^{2}\frac{c_ig_i}{1+\lambda\gamma_i} \\ + \sum_{i=1}^{2}\frac{e_if_i\gamma_i}{(1+\lambda\gamma_i)^2} - \frac{\lambda}{2}\sum_{i=1}^{2}\frac{(e_ig_i + c_if_i)\gamma_i}{(1+\lambda\gamma_i)^2} \\ + \frac{\lambda^2}{4}\sum_{i=1}^{2}\frac{c_ig_i\gamma_i}{(1+\lambda\gamma_i)^2} = 0\,, \end{aligned} \tag{10}$$

using the method described in [3], which manipulates the equation into a fifth-degree polynomial on λ.

The estimated value of t, $\widehat{t}$, corresponds to the first j elements of the augmented state, calculated using

$$\widehat{z} = (E^T\Psi^{-1}E + \lambda^*P)^{-1}(E^T\Psi^{-1}B - \lambda^*w)\,. \tag{11}$$

E. GAUSS-NEWTON

Gauss-Newton (GN) method is an iterative method to minimize a sum of squared function values for scenarios with additive noise with finite variance. Some disadvantages are: its greater computational cost compared to the methods above, the need for an initial starting position, only finding one candidate solution, and may exhibit divergence problems in the presence of high noise levels or get trapped into local minima which may result in high estimation error.

In this paper we consider the regularized GN method [3], which reduces the divergence problems using an algorithm that is equivalent to a Bayesian maximum-a-posteriori algorithm with prior distribution chosen around the center point of the MT positions. Let t_d denote the mean point of s_i, $i = 1, ..., N$ and c a regularization coefficient.

The algorithm starts by calculating the initial starting position, $t_0 = t_d$ and setting $k = 0$. The iteration cycle is:

1) Compute the Jacobian

$$J_k(t) = \begin{bmatrix} \frac{(s_1-t)^T}{\|(s_1-t)\|} \\ \vdots \\ \frac{(s_N-t)^T}{\|s_N-t\|} \end{bmatrix} \tag{12}$$

2) Define $t_{k+1} = t_k + \Delta t_k$ where Δt_k is the least-squares solution to

$$\left(-\Sigma^{-\frac{1}{2}}J_k + cI\right)\Delta t_k = \Sigma^{-\frac{1}{2}}(h(t_k) - d) + c(t - t_d), \tag{13}$$

where $\Sigma^{-\frac{1}{2}}$ denotes the matrix square root inverse operator.

3) If stopping condition $\|\Delta t_k\| < \delta$ is not satisfied and $k < K_{max}$, increment k and repeat the cycle.

The results presented in section VI were obtained by using a maximum number of iterations of $K_{max} = 8$, $c = 10^{-4}$ and the stopping tolerance $\delta = 2 * 10^{-2}$.

F. LEVENBERG-MARQUARDT

The Levenberg-Marquardt (LM) algorithm is an alternative iterative technique that locates the minimum of a function expressed as the sum of squares of nonlinear functions [26]. LM can be thought of as a combination of gradient descent and the GN method. When the current solution is far from the correct one, the algorithm behaves like a gradient descent method: slow, but guaranteed to converge. When the current solution is close to the correct solution, it becomes a GN method. This is implemented using a dampening factor ϑ, which is calculated in each iteration. A weight $w_i, i = 1, ..., N$ is used in the sum, with a value $w_i = 1/{d_i}^2$. To avoid

divergence problems due to matrices over-dimensioning, the measured distances are bound to $d_i = \max(d_i, 10^{-7})$.

As before, the algorithm starts by calculating the initial starting position, $t_0 = t_d$ and defining the stopping tolerance δ (10^{-4} times the function tolerance). In each step, the Jacobian J is calculated, and $t_{k+1} = t_k + \Delta t_k$ where Δt_k is the least-squares solution to

$$(J^T J + \vartheta \,\mathrm{diag}(J^T J))\Delta t = J^T \varepsilon, \tag{14}$$

where $\mathrm{diag}(J^T J)$ is a diagonal matrix consisting of the diagonal elements of $J^T J$. The dampening factor ϑ is updated based on the success of the last t_k update. The cycle stops when $\|\Delta t_k\| < \delta$, or when a K_{max} is reached.

LM algorithm was initially implemented using the Matlab library, and latter selected for the non-collaborative Android prototype, where it was implemented using the *org.apache.commons.math3.fitting.leastsquares.LevenbergMarquardtOptimizer* library, with a maximum number of iterations $K_{max} = 1000$.

G. TOMIC SOCP RELAXATION

Tomic *et al.* [4] derived a convex estimator, by tightly approximating the ML estimator for small noise. This estimator is based on SOCP relaxation technique.

Contrarily to the previous algorithms, this algorithm handles γ and the transmission power P_T as unknowns, estimating them simultaneously with the t's position. The following path loss model (dB) was considered,

$$L_i = L_0 + 10\,\gamma\,\log_{10}\frac{\|t - s_i\|}{d_0} + v_i\,,\ i = 1, \ldots, N, \tag{15}$$

where L_0 denotes the value of losses along the path at a short reference distance d_0 ($\|t - s_i\| \geq d_0$) and v_i is the log-normal shadowing term modeled as a zero-mean Gaussian random variable with variance σ_i^2, i.e. $v_i \sim \mathcal{N}\left(0, \sigma_i^2\right)$. This formulation combines (1) and (2) in a single equation.

To formulate the joint localization problem, the ML criterion was considered, but finding the ML estimator, $\widehat{\theta}$, implies solving the non-linear and non-convex least-squares problem,

$$\widehat{\theta} = \underset{\theta=[t;L_0;\gamma]}{\arg\min} \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \times \left[\left(L_i - h^T\theta\right) - 10 g^T\theta \log_{10}\frac{\left\|C^T\theta - s_i\right\|}{d_0}\right]^2, \tag{16}$$

where $h = [0_{2\times1}; 1; 0]$, $g = [0_{3\times1}; 1]$ and $C = [I_2; 0_{2\times2}]$. The problem defined by (16) is not convex and has no closed-form solution. Therefore, the solution proposed in [4] estimates the position of interest following the iterative procedure:

1) Set the initial estimate of γ, $\widehat{\gamma^0} \in [\gamma_{min}, \gamma_{max}]$, and set the iterator counter $k = 1$.
2) Solve the SOCP problem [4]:

$$\underset{t,g,z,\eta,p}{\text{minimize}}\ p$$

$$\text{subject to} \left\|\begin{bmatrix} 2z \\ p - s_i \end{bmatrix}\right\| \leq p + 1,\ \|t - s_i\| \leq g_i,$$
$$z_i = \psi_i g_i - \eta d_0, \quad i = 1, \ldots, N. \tag{17}$$

3) Use $\widehat{\gamma}^{k-1}$ and $\widehat{t}^{k-1}$ to compute the ML estimative of L_0, $\widehat{L_0}$

$$\widehat{L}_0^k = \frac{\sum_{i=1}^{N}\left(L_i - 10\widehat{\gamma}^{k-1}\log_{10}\frac{\|\widehat{t}^{k-1} - s_i\|}{d_0}\right)}{N} \tag{18}$$

4) Use $\widehat{t}^{k-1}$ and $\widehat{L}_0^k$ to find the ML estimative of γ, $\widehat{\gamma}$

$$\widehat{\gamma}^k = \frac{\sum_{i=1}^{N} 10\log_{10}\frac{\|\widehat{t}^{k-1} - s_i\|}{d_0}\left(L_i - \widehat{L}_0^k\right)}{\sum_{i=1}^{N}\left(10\log_{10}\frac{\|\widehat{t}^{k-1} - s_i\|}{d_0}L_i - \widehat{L}_0^k\right)^2} \tag{19}$$

If $\widehat{\gamma}^k \notin [\gamma_{min},\ \gamma_{max}]$ the process should be stopped at this point and use $\widehat{t}^{k-1}$ as the final estimate.

5) If $k > K_{max}^T$ (K_{max}^T represents the maximum number of iterations) it must stop and consider $\widehat{t}^k$ as an estimate; Otherwise repeat step 2 with the help of $\widehat{\gamma}^k$ and $\widehat{L}_0^k$ and increment k.

This type of algorithm (which uses convex optimization) has a large processing time that is typically 10^3 times higher than the previous iterative ones.

H. PEDRO-TOMIC SOCP RELAXATION

The Pedro-Tomic (PT) SOCP relaxation algorithm is a new SOCP based algorithm proposed in this paper, which trades off accuracy for a lower complexity. Rewriting the problem propagation model (15) in function of the RSS, we get

$$P_i = P_0 - 10\,\gamma\,\log_{10}\frac{\|t - s_i\|}{d_0} - v_i\,,\ i = 1, \ldots, N \tag{20}$$

where P_0 denotes the RSS value at a reference distance d_0 ($\|t - s_i\| \geq d_0$) and v_i is modelled by a zero-mean Gaussian random variable with variance σ_i^2, i.e. $v_i \sim \mathcal{N}\left(0, \sigma_i^2\right)$.

Solving the equation with respect to the distance between the emitter t and the measurement point s_i we obtain,

$$\|t - s_i\| = e^{\frac{P_0}{\rho}} e^{-\frac{P_i}{\rho}} e^{-\frac{v_i}{\rho}} \tag{21}$$

where $\rho = \frac{10\gamma}{\ln 10}$. Defining $\lambda_i = e^{\frac{P_i}{\rho}}$ and considering a first order Taylor series approximation for the last term in (21) we define,

$$\lambda_i \|t - s_i\| \approx e^{\frac{P_0}{\rho}}\left(1 - \frac{v_i}{\rho}\right) \approx e^{\frac{P_0}{\rho}} - \epsilon_i, \tag{22}$$

where $\epsilon_i \sim \mathcal{N}\left(0, e^{\frac{2P_0}{\rho}}\frac{\sigma_i^2}{\rho^2}\right)$. By squaring (22) and defining $\theta = e^{\frac{2\,P_0}{\rho}}$, we obtain

$$\theta \approx \lambda_i^2 \|t - s_i\|^2 + 2\lambda_i \|t - s_i\| \epsilon_i + \epsilon_i^2\,. \tag{23}$$

Considering $\epsilon_i \approx 0$, we can neglect the second order noise term. In this way, we arrive at our non-linear and non-convex least squares problem

$$(\widehat{t_i}, \widehat{\theta_i}) = \underset{t,\theta}{\arg\min} \sum_{i=1}^{N}\left(\frac{\theta - \lambda_i^2\|t - s_i\|^2}{2\lambda_i\|t - s_i\|}\right)^2. \tag{24}$$

Starting from $\|t - s_i\|^2$ expression and introducing $y = \|t\|^2$, it can be written as

$$\begin{aligned} &\underset{t,\theta,y}{\text{minimize}} \sum_{i=1}^{N} \left[\frac{\theta - \lambda_i^2 \left(y - 2 s_i^T t + \|s_i\|^2 \right)}{2\lambda_i \|t - s_i\|} \right]^2 \\ &\text{subject to } \; y = \|t\|^2 \end{aligned} \tag{25}$$

Additionally, an epigraph variable ζ was introduced and a SOCP relaxation[2] was applied, to get

$$\begin{aligned} &\underset{t,\theta,y,\zeta}{\text{minimize}} \; \zeta \\ &\text{subject to} \\ &\qquad \left\| \begin{bmatrix} 2\theta - 2\lambda_i^2 \left(y - 2\, s_i^T t + \|s_i\|^2 \right) \\ 4\lambda_i^2 \left(y - 2\, s_i^T t + \|s_i\|^2 \right) - \zeta \end{bmatrix} \right\| \\ &\qquad \leq 4\lambda_i^2 \left(y - 2\, s_i^T t + \|s_i\|^2 \right) + \zeta \; , \\ &\qquad \left\| \begin{bmatrix} 2t \\ y - 1 \end{bmatrix} \right\| \leq y + 1 \end{aligned} \tag{26}$$

This minimization problem can be treated iteratively using the following algorithm:

1) Set the initial estimate of γ, $\widehat{\gamma^0} \in [\gamma_{min}, \gamma_{max}]$.
2) Solve the SOCP problem defined by (26) for the initial estimate of t, $\widehat{t^0}$.
3) Use $\widehat{\gamma}^0$ and $\widehat{t}^0$ to compute the ML estimative of P_0^0, $\widehat{P}_0^0$, using
$$\widehat{P}_0 = \frac{\sum_{i=1}^{N} \left(P_i - 10\gamma^0 \log_{10} \frac{\|\widehat{t}^0 - s_i\|}{d_0} \right)}{N} \; . \tag{27}$$
4) Obtain
$$f_0 \;\; = \;\; \sum_{i=1}^{N} \left[\left(\widehat{P}_0^0 - P_i \right) - 10\, \widehat{\gamma}^0 \; \log_{10} \frac{\|\widehat{t}^0 - s_i\|}{d_0} \right]^2,$$
the initial cost function value, using $\left\{ \widehat{t}^0, \; \widehat{P}_0^0, \; \widehat{\gamma}^0 \right\}$.
5) Set the iterator counter $k = 1$.
6) Use $\widehat{t}^{k-1}$ and $\widehat{L}_0^{k-1}$ to compute the ML estimative of γ,
$$\widehat{\gamma}^k = \frac{\sum_{i=1}^{N} 10 \log_{10} \frac{\|\widehat{t}^{k-1} - s_i\|}{d_0} \left(\widehat{P}_0^{k-1} - P_i \right)}{\sum_{i=1}^{N} \left[10 \log_{10} \frac{\|\widehat{t}^{k-1} - s_i\|}{d_0} \left(\widehat{P}_0^{k-1} - P_i \right) \right]^2} \; . \tag{28}$$
If $\widehat{\gamma}^k \notin [\gamma_{min}, \; \gamma_{max}]$ the process should be stopped at this point and use $\widehat{t}^{k-1}$ as the final estimate. Otherwise, θ is estimated using $\widehat{\theta}^k = \exp\left(\frac{2\widehat{P}_0^k - 1}{\widehat{\rho}^k - 1} \right)$.
7) Use $\widehat{\gamma}^k$, $\widehat{P}_0^{k-1}$ and $\widehat{\theta}^k$ to solve (26) and get t estimate, $\widehat{t}^k$.
8) Use (27) to get P_0 estimate, $\widehat{P}_0^k$.
9) Get $f_k \;\; = \;\; \sum_{i=1}^{N} \left[\left(\widehat{P}_0^k - P_i \right) - 10\, \widehat{\gamma}^k \; \log_{10} \frac{\|\widehat{t}_k - s_i\|}{d_0} \right]^2$, the cost function value. If $\frac{|f_k - f_{k-1}|}{f_{k-1}} < \varepsilon$ (ε denotes the stopping threshold) or $k > K_{max}$ (K_{max} represents the maximum number of iterations) it must stop and consider $\widehat{t}^k$ as estimate; Otherwise repeat step 6 with the help of $\widehat{\gamma}^k$ and $\widehat{P}_0^k$ and increment k.

[2]SOCP relaxation used: $\left(\frac{x}{y}\right)^2 = z \Rightarrow \left\| \begin{bmatrix} 2x \\ y^2 - z \end{bmatrix} \right\| \leq y^2 + z$.

V. COOPERATIVE LOCALIZATION

The localization algorithms presented above require the collection of a set of RSS measurements at different locations of the remote node. If the remote node is allowed to move, the MT has to move much faster. Cooperation between MTs can be used as an alternative mechanism to speed the measurements collection, allowing the tracking of mobile objects. The communication mechanism can also be used to offload the computational cost from the MT to an infrastructure's server, contributing to save the MT's battery.

Cooperation between heterogeneous MTs using the location provided by the odometry module introduces new challenges:

- The MT's coordinates must refer to the same referential;
- The different antenna gains G_r must be homogenized, to allow the direct combination of RSS values measured by different MT's types;
- A communication service is required to coordinate RSS vector exchange, handle the data homogenization tasks and provide the computation services, when available.

Most of these challenges can be easily addressed when a set of anchor APs is deployed with known location and transmission powers, which can be used to calibrate the individual measurements. When no anchor APs are available, the MT's coordinates and the antenna gains can be estimated using the local RSS measurements to a set of common remote entities that can be located by the cooperating MTs. We consider from now on, and without any loss of generality, that remote entities are APs because the algorithms use the SSID signal. If another signal was used, the nodes could be ordinary nodes. The MTs coordinate's offset and axis misalignment can be estimated comparing the location of a set of APs. The antenna gain can be easily estimated running an algorithm that estimates P_0 (the transmission power of the common AP), such as the SOCP ones presented in sections IV-G and IV-H. In order to use measurements from different MTs, all individual measurements have to be corrected in relation to a common reference and power level. The results presented in this paper consider a single common AP, and have their different path loss coefficient values and the distances to the axis origin compensated. This algorithm does not compensate unaligned axis directions in the measurements collected by different MTs, which would require using more common APs to calibrate the directions.

The algorithm applied to correct the RSS values of each individual measurement was the following:

1) One AP is selected as a reference and its position is calculated using one of the described algorithms and local RSS measurements exclusively;
2) The Tomic SOCP algorithm is run to estimate P_0's value, $\widehat{P}_0$, and γ's value, $\widehat{\gamma}$.

Given $\widehat{P}_0$ and $\widehat{\gamma}$, the individual measurements can be corrected to a given reference value P_0^r and d_0^r by replacing the

α value in (1) by $\widehat{\alpha_i}$, calculated using (29).

$$\widehat{\alpha_i} = 20\log_{10}\left(d_0^r f\, 10^{\frac{\widehat{P}_0 - P_0^r}{10\widehat{\gamma}}}\right) - \left|P_0^r\right|. \tag{29}$$

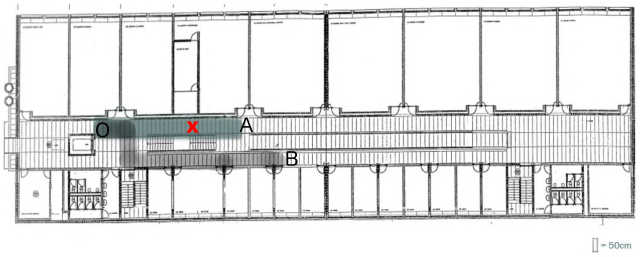


FIGURE 5. Testing scenarios in NOVA EE building.

VI. PERFORMANCE EVALUATION

The performance of the odometry module and the localization algorithms was analyzed considering an indoor scenario, at the third floor of the Electrical Engineering (EE) Department building at NOVA University of Lisbon Campus, represented in figure 5. Two two-way walking paths were considered: i) OA path in a 16.5 m straight line path, executed in 33 steps with an average of 50 cm each; ii) OB path is a 29 m path with two turns, performed in 58 steps. A Cisco Aironet 1100 AP is placed on the ceiling, at the position depicted by a red cross, and using frequency 2452 MHz (channel 6). The smartphone is placed 1.72 m below the ceiling. Three smartphones were used in the tests: one Sony XPeria Z1 Compact running Android 5.1.1 (MT1), one One-Plus 2 running Android 6.0.1 (MT2) and one Xiaomi Redmi Note 5A Prime (MT3) running Android 7.1.2. The same application was run in all MTs.

The first subsection evaluates the influence of several direction detecting sensors on the accuracy of the MT's location estimation (s_i) by the odometry module; the second analyses the variation of the path loss in the test scenario and the third and fourth ones evaluate the accuracy of the localization algorithms in locating the AP represented in figure 5 considering respectively non-cooperative and cooperative modes.

A. ODOMETRY ACCURACY

This section's main objective is to evaluate the MT's location accuracy, focusing on the influence of the direction estimation sensors, in a path with a known number of fixed sized steps (OA). The step detector counted 33 ± 1 steps in more than 99% of the tests, showing that most of the odometry's location errors are due to direction errors.

Figures 6 and 7 depict one sample run using respectively MT1 and MT2. It can be seen that the overall direction detected in the path's farthest location varies with the sensors and the MT used. The true location of the path's end point is also shown, located at $X = 16.4$ m, $Y = 2$ m. The figures show that the two methods based on the compass sensor (Acc+Comp and Grav+Comp) estimate approximately the correct direction at the extreme points of the paths reasonably well. However, they are strongly influenced by variations

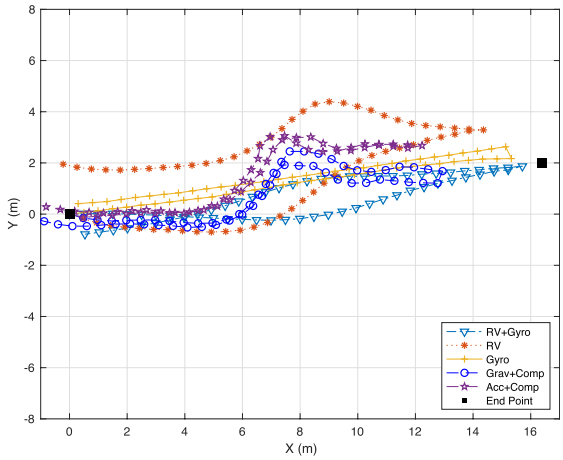


FIGURE 6. Odometry detected path using MT1.

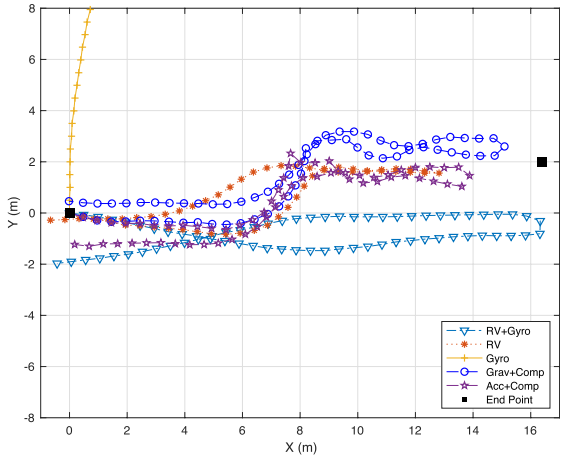


FIGURE 7. Odometry detected path using MT2.

in the magnetic field near $X \approx 8$ m, producing similar deviations in MT1 and MT2. MT1 and MT2 had completely different outputs for the calibrated gyroscope sensor (Gyro), demonstrating a very strong dependence on the hardware. The calibrated gyroscope sensor had the best performance for MT1 and the worst for MT2. The Android's RV software sensor combines the information of the accelerometer, gyroscope and compass. RV sensor produced a location path where it is still visible the magnetic field variation near $X \approx 8$ m and some direction deviation for MT1 and MT2. The best immunity to the magnetic field variation was achieved combining the RV sensor and the calibrated gyroscope using the Kalman filter implemented in [22] (RV+Gyro), although introducing some deviation in the overall direction for MT2. In order to reduce the influence of the step detection errors in the accuracy metric, we measured the average deviation for all tests using MT1 and MT2 at the intermediate farthest point of the path and the deviation at the end of the tests (which should coincide with the starting point), denoted respectively as ε_{int} and ε_{end}. The results represented in Table 1 show that the best average accuracy was achieved using the RV sensor & Calibrated Gyroscope and the Gravity Sensor & Compass sensors, with a slight advantage for the first one, with an

TABLE 1. Odometry average deviation.

Algorithm	ε_{int} (m)	ε_{end} (m)	Requires Training
Accelerometer & Compass	3.40	1.05	No
Gravity Sensor & Compass	2.46	0.91	No
Calibrated Gyroscope	15.52	4.59	Yes
RV Sensor	3.23	1.34	Yes
RV Sensor & Cal. Gyrosc.	1.76	1.63	**Yes**
RV Sensor & Uncal. Gyrosc.	6.70	7.89	**No**

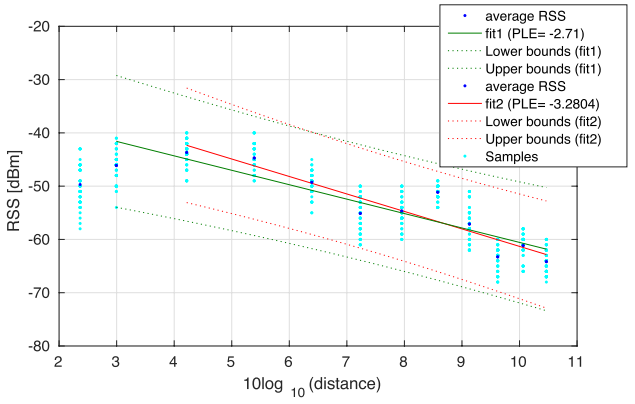


FIGURE 8. RSS fitting using MT1 measurements in OA path.

average location error below 2 m at both measurement points. They also highlight the importance of the calibration of the gyroscope. When the gyroscope is used without being calibrated with the RV sensor, the results show that the Kalman filter is not capable of compensating the estimation error. Given its greater immunity to the variations of the magnetic field, the RV sensor & Calibrated Gyroscope was selected to be used in the localization accuracy tests.

B. PATH LOSS VARIATION

Most of the localization algorithms assume a constant PLE. To validate this assumption, 75 samples of RSS were measured using MT1 in path OA in the positions with horizontal distances of 0, 1, ..., 11 m to the AP, faced to the AP. Figure 8 depicts the measured values and the average values for each distance. Considering horizontal distances equal to or above 2 m (corresponding to a total distance to the AP of 2.64 m, equivalent to 4.2 dB), the linear regression of the average values of the measured RSS (fit2) obtains a good estimate ($rss = -3.2804 \times 10\log_{10}(d) - 28.5016$ dBm), with a Root Mean Square Error (RMSE) of 2.66 dB and a R^2 value of 0.8806. The estimated PLE was 3.2804, with a 95% confidence interval of [2.296, 4.265]. On the other hand, considering horizontal distances equal to or above 1 m (corresponding to a total distance to the AP of 1.99 m $\equiv$ 2.99 dB), fit1, the best estimate is $rss = -2.71 \times 10\log_{10}(d) - 33.48$ dBm, with a RMSE of 3.20 dBm and a R^2 value of 0.8288 (less accurate). The estimated PLE was 2.7075, with a 95% confidence interval of [1.7797, 3.6353]. These high variances are compatible with the measurements spread shown for each distance depicted in figure 8. The 99% confidence interval is plotted for both fitting curves, showing that all samples are within the interval. When the distance 0 m is considered (not plotted), the linear regression fit estimates a smaller PLE (2.17) but introduces a higher error (R^2 of 0.7185 and a RMSE of 3.9397 dB). When the horizontal distance is 0 m or 1 m, the average received RSS tends to decrease compared to value measured at 2 m, due to the narrower propagation paths available between the smartphone and the AP, related with the directionality of the antennas. This shows that depending on the sample set considered, a different PLE value would be more adequate, probably contained in the interval [2, 4]. The RMSE was calculated using,

$$\text{RMSE} = \sqrt{\sum_{i=1}^{M} \frac{\left\| t - \widehat{t}_i \right\|^2}{M}}, \tag{30}$$

where M denotes the size of the measurement set.

C. NON-COOPERATIVE LOCALIZATION ACCURACY

The localization accuracy tests were done using 672 measurements collected using MT1 (85%), MT2 (5%) and MT3 (10%) in OA and OB paths depicted in figure 5, as described above. During the tests, the Android application was configured to collect periodically the last AP's RSS measurement, the MT's location from the odometry service and the sample time. Each test was done moving at an average speed ranging from 1 to 3 km/h (they lasted between 45 s and 105 s for path OA and between 50 s and 140 s for path OB) and using two sampling periods: one and five seconds. All algorithms described in section IV were implemented in MATLAB and run using the data collected, to allow a fair comparison. Moreover, the parameter values $\gamma^0 = 3$, $\gamma_{min} = 2$, $\gamma_{max} = 4$ and $K^T_{max} = 3$ were considered as input parameters for the algorithms, taking into account the PLE values measured in the previous section. The Euclidean distance to the (known) exact location of the AP was the metric used to compare the accuracy of one estimated location. The overall performance metric considered for each algorithm was the RMSE.

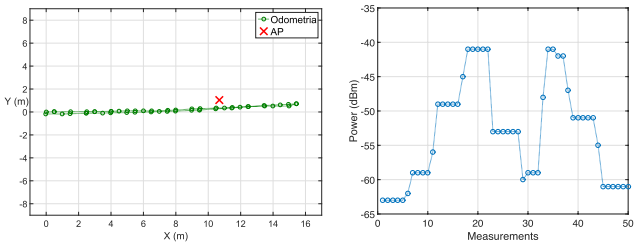


FIGURE 9. Odometry estimated location and RSS measured by MT1. (a) Odometry. (b) RSS.

As an example, figures 9 and 10 depict the RSS values and the MT's location measured in two of the runs on path OA with a sampling period of one second, respectively using MT1 and MT2. The AP position is depicted in relation to each MT's measured direction, showing that the coordinates

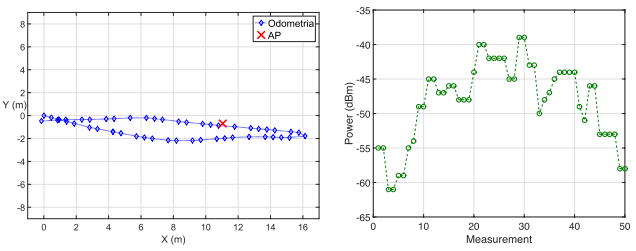


FIGURE 10. Odometry estimated location and RSS measured by MT2. (a) Odometry. (b) RSS.

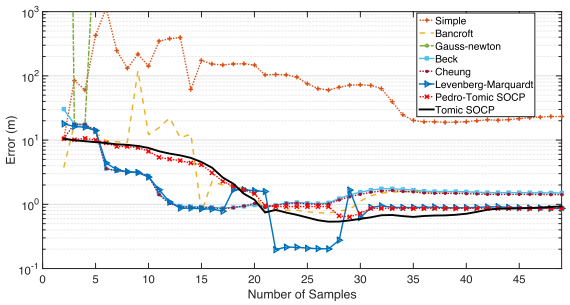


FIGURE 11. Average AP's non-cooperative location accuracy for MT1.

are not fully aligned - an offset of 2 m exists between MT1's and MT2's estimated AP position. The figures also show the presence of a significant additive noise in the measured RSS, possibly due to the presence of other APs using the same channel placed on the wall in the corridor of the EE department building. The corridor is an open space of the three floors, and therefore the signals interfere easily. This effect was identified in initial RSS measurements made with a static MT, where RSS fluctuations of up to 10 dBm were measured. The expected variation of the RSS during the run was measured by MT1, with two relative peak values when the MT passed near the AP and lower values at the end and at the begin of the tests. On the other hand, the additive noise seems to be higher for MT2, which measured a less defined pattern than when MT1 passed near the AP.

Figure 11 shows the accuracy achieved when the algorithms presented in section IV were used to estimate the position of the AP using the RSS samples depicted in figure 9. The figure illustrates how the accuracy evolves as more samples are added to the RSS vector during the run. It shows that the Gauss-Netwon method tends not to converge and the simple intersection tends to exhibit a low accuracy which may persist through all the run. All other algorithms tend to stay below a baseline accuracy after an initial transition phase, as an increased range of RSS values becomes available and the increased independence of the samples starts to allow reducing the effect of the RSS measurement errors. The highest accuracy is measured after having 20 samples, after first passing through the AP. The fastest algorithms to achieve a consistent accuracy below 5 m were Beck, Cheung and Levenberg-Marquardt (LM), after 6 samples for MT1. The LM algorithm has the additional advantage of achieving a higher accuracy when samples with higher RSS values are available near the AP, as is shown between samples 22 and 28 for MT1. On the other hand, the figure shows that the SOCP methods (Tomic and PT SOCP Relaxation) tend to require more samples to achieve the 5 m accuracy (around 15 samples), but converge with a higher accuracy. Besides the AP location, SOCP algorithms estimate the path loss and transmission powers. Therefore, they require more samples to converge, but are able to adapt to scenarios where these parameters diverge from the predefined ones.

The best accuracy measured in the experiment depicted in figure 11 is overly optimistic because the localization algorithms were applied over all samples collected in the experiment, where a significant set of RSS and MT location samples have very low interfering noise and location error. In practice, the number of samples used in the algorithms is upper limited by CPU or RAM limitations. Therefore, the analysis that follows for the accuracy of the algorithms considers a window size, W, which defines the exact length of the vector of samples provided to the algorithms.

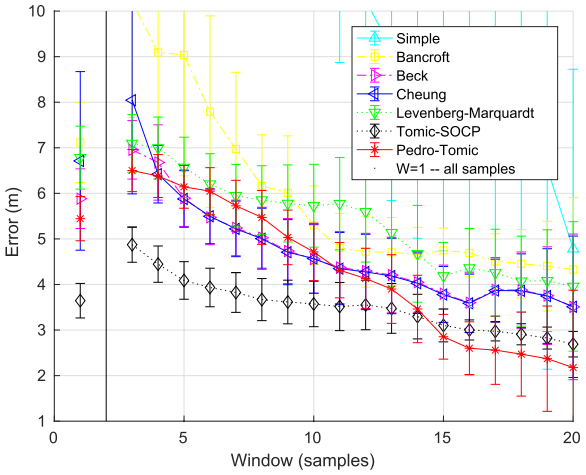


FIGURE 12. ocation accuracy for path OA with different window sizes and $T = 5$ s.

Figures 12 and 13 depict the location accuracy measured with a sampling period of $T = 5$s respectively on path OA (209 samples) and OB (97 samples), using MT1 and MT3. Figure 14 shows the location accuracy measured with $T = 1$ s for OA and OB (respectively with 271 and 95 samples) using MT1, MT2 and MT3. Figures 12, 13 and 14 depict the average error and the 95% confidence interval for a varying window size starting at $W = 3$. They also depict the average error when all the measurements collected until a given instant are used, starting with 3 samples, represented at $W = 1$. It can be seen that the measured average error is above 2 m for all algorithms and setups tested, but the error tends to reduce for a larger window, as expected. As more samples are included, there is an higher probability of having points with higher accuracy in the sample set, and the algorithms tend to weight more these points. On the other hand, the variation is

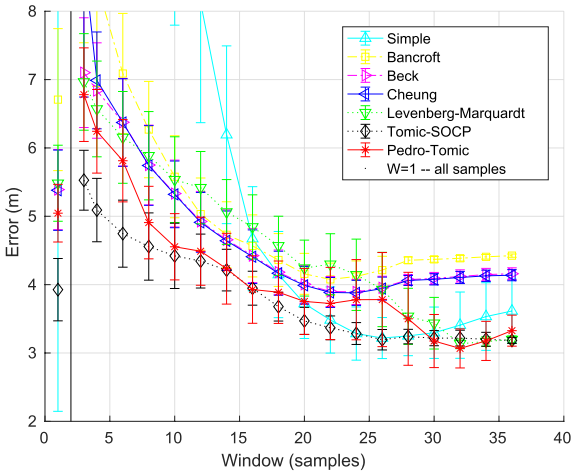


FIGURE 13. Location accuracy for path OB with different window sizes and $T = 5$ s.

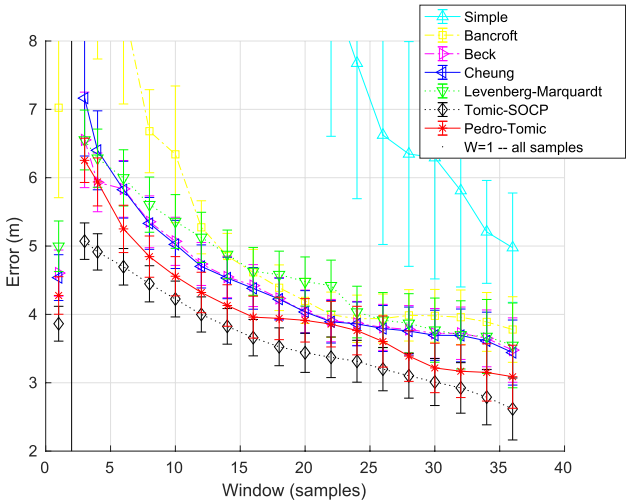


FIGURE 14. Location accuracy for mixed paths with different window sizes for $T = 1$ s.

not regular. The influence of the longer runs is higher for higher window values, producing a variation in the contribution of the runs to the average. The measurements with an unlimited window include measurements with size starting with $W = 3$ until the maximum number of samples reached. Therefore, they tend to be less precise compared with the error measured for a large fixed window size (specially for $T = 5$ s), which does not account for errors with small window sizes. Tomic SOCP algorithm outperforms the others considering a fixed window size and a unlimited window, except for PT SOCP for larger window sizes on path OA with $T = 5$ s. PT SOCP has the second lower error in most conditions. It is less complex than Tomic SOCP and it seems to require a window of 10 or more samples, to approach the performance of Tomic SOCP. Beck and Cheung algorithms have very similar accuracies, except for very small windows, where Beck outperforms Cheung. LM algorithm has (figures 11 and 13) some periods and window values with high performance. But it is not fully

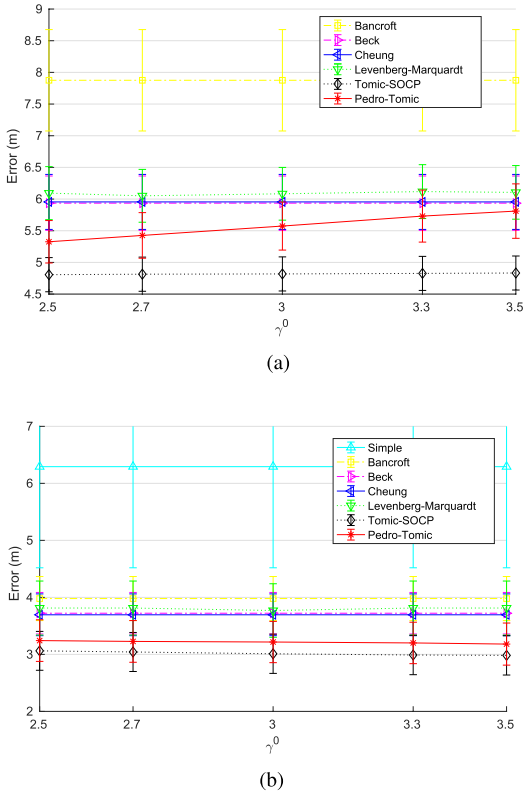


FIGURE 15. Sensitivity of the location accuracy to γ^0 for mixed paths with different window sizes for $T = 1$ s. (a) $W = 4$. (b) $W = 30$.

consistent for all tested scenarios, showing some performance degradation for intermediate window values. Gauss-Newton algorithm produced the highest errors (above 10 m). The results show that when the window size is high, the simple algorithm produces acceptable results, specially for a larger sample time, where the correlation between samples is lower.

The sensitivity of the location algorithms to γ^0 value is depicted in figure 15 for the conditions of figure 14 in the interval [2.5, 3.5]. It is shown that for a short window size of $W = 4$, Pedro-Tomic SOCP algorithm is more sensitive than the other ones, but this effect is minimized when a larger window size is considered ($W = 30$).

It is not surprising that SOCP algorithms outperform least squares ones. The latter ones are based on sequences of linear approximations which usually lead to poor localization accuracy in environments with high noise powers. On the other hand, SOCP methods (and convex-based methods in general) apply tight approximations that do not require linearization of the measurement model, and are thus more robust to noise-corrupted measurements. Nevertheless, this gain in the accuracy of SOCP methods comes with a certain cost in terms of the computational complexity. Figure 16 depicts the average processing time for all the algorithms on all the samples used in this section, measured in a Linux machine with an Intel Core i7-4790 processor and 8GB of DDR3-1600 RAM, running Matlab and CVX. The 95% confidence interval

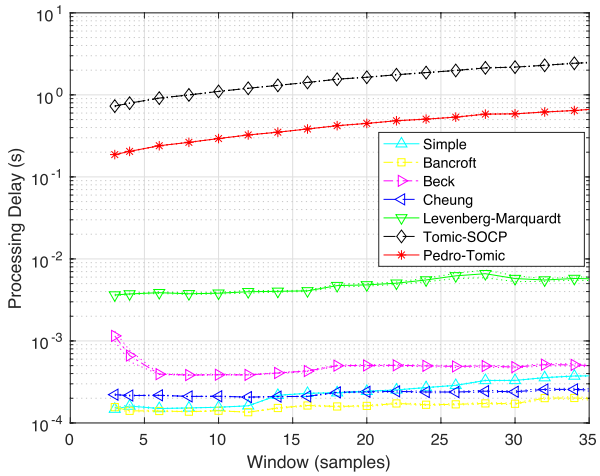


FIGURE 16. Processing time for different vector sizes.

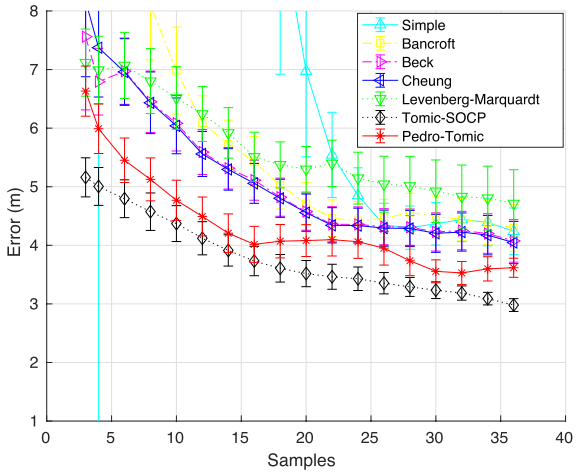


FIGURE 17. Location accuracy indoor for mixed path without cooperation.

TABLE 2. RMSE and calculation time for $W = 30$.

Algorithm	$\text{RMSE}_{T=1}$ (m)	$\text{RMSE}_{T=5}$ (m)	Runtime (ms)
Simple	11.05	3.36	3.299
Bancroft	4.44	4.37	0.172
Beck	4.16	4.09	0.480
Cheung	4.14	4.08	0.240
Levenberg Marquardt	4.47	3.50	5.763
PT SOCP	3.71	3.25	2.44×10^3
Tomic-SOCP	3.49	3.22	586.9

is represented, which is small for all algorithms. It shows a trade-off between accuracy and processing complexity, where the higher accuracies measured also have an higher processing cost. The SOCP ones have higher processing requirements, even for a low number of samples, increasing with the vector size. The simple algorithm also has a significant increase with the vector size due to the matrix inversion operation. It can be seen that LM is the third more computationally intensive algorithm in the Matlab implementation after the SOCP ones. For small window sizes, Beck's algorithm shows some misbehaving when the samples are correlated - when it is looking for valid bissection limits.

Table 2 presents the RMSE for a window of $W = 30$ with a sampling period $T = 5$ s for path OB and for $T = 1$ s, whose average errors are depicted in figures 13 and 14. Table 2 also presents the average calculation time (Runtime) used by the algorithms in MATLAB for a measurement vector with 30 samples. In order to run the Android application in real-time, mitigate old readings effects and process estimations without influencing the application fluidity, we decided to limit the RSS measurements stored in the Android application prototype to 30 samples. Using the *org.apache.commons.math3* library, we were able to measure an average run time of 13.47 ms with the prototype application running in MT2. Therefore, due to its good trade-off between accuracy and execution time, this algorithm was

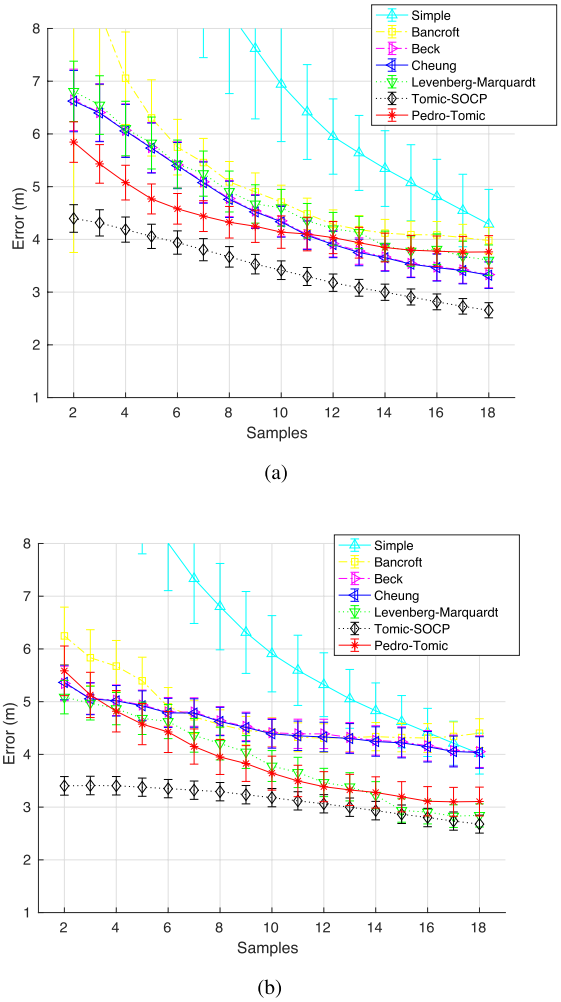
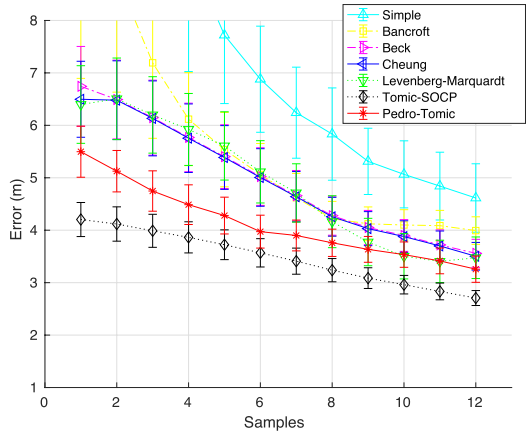
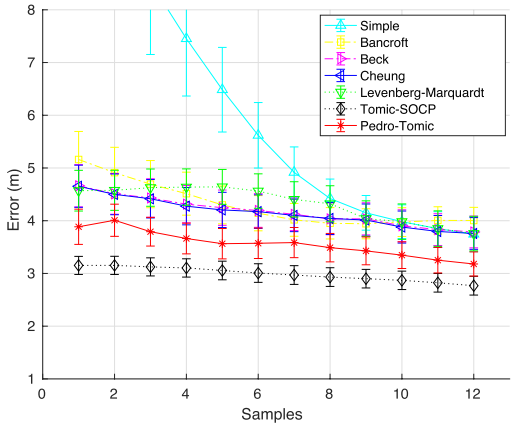


FIGURE 18. Location accuracy indoor for mixed path with cooperation between 2 MTs. (a) Synchronized. (b) Scattered.

chosen for the Android application implementation for non-cooperative localization [19]. PT SOCP algorithm is four times lighter than Tomic SOCP, but it is still about 100 times slower than LM. Therefore, although SOCP algorithms are

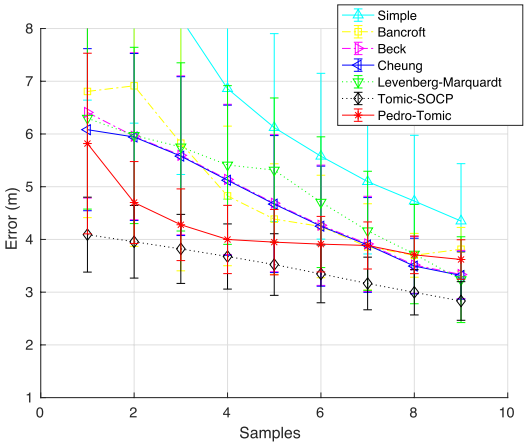


(a)

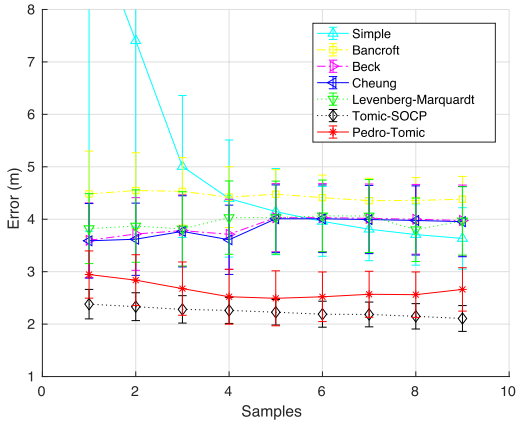


(b)

FIGURE 19. Location accuracy for mixed path with cooperation between 3 MTs. (a) Synchronized. (b) Scattered.



(a)



(b)

FIGURE 20. Location accuracy for mixed path with cooperation between 4 MTs. (a) Synchronized. (b) Scattered.

more accurate, they are not suited yet for running in a typical mobile phone. But they may be suitable for a cooperative centralized service implementation.

D. COOPERATIVE LOCALIZATION ACCURACY

This section evaluates the accuracy of the location algorithms when multiple measurements of MT1, MT2 and MT3 are combined using the algorithm described in section V. Non-cooperative mode requires at least three measurements in different locations to localize a remote node. When G MTs cooperate, and the samples collected by each individual MT are shared by all of them synchronously, the individual number of samples required to have a window W is reduced to $\lceil W/G \rceil$, where $\lceil\ \rceil$ denotes the minimum integer value above a real number.

A subset of the initial measurements was considered in the next results, composed by 176 samples captured in 4 runs (2 in OA and 2 in OB), 2 of MT1, 1 of MT2 and 1 of MT3, with $T = 1$ s. Given that the effectiveness of the algorithms depends on how independent the samples collected are, two extreme scenarios were considered in the experiments: in a synchronized scenario, all MTs start from position O and follow their path until the first returns to the starting point; in a scattered scenario, the MTs start from different positions in a cooperation group (from O, A or B). This was implemented using the same sequences of RSS samples shifted to start in position A or B. The $\widehat{\alpha_i}$ and confidence intervals estimated by MT1, MT2 and MT3 using the cooperation algorithm proposed were respectively of -40.72 ± 0.64 dBm, -37.12 ± 1.23 dBm, and -38.55 ± 0.59 dBm, denoting higher antenna gains for the largest terminals, MT2 and MT3.

The experiments compare the average accuracy achieved in the following conditions: all four runs when MTs do not cooperate; the six combinations of runs cooperation 2 by 2; the four combinations of runs cooperation 3 by 3; all 4 runs cooperate (two MTs started from 0 in the scattered scenario). The accuracy measured for a number of MT samples for cooperation of 2, 3 and 4 MTs are depicted respectively in figures 18, 19 and 20. The average accuracy when no cooperation is used is depicted in figure 17. The results show that the

cooperative algorithm is capable of handling measurements from heterogeneous MTs, and to increase the accuracy and reduce the sampling period. They show that the accuracy is substantially improved for a given number of samples when more MTs cooperate, and that with 3 or 4 cooperating, it is possible to use a single sample to localize a remote node with an accuracy of 4 m or lower using Tomic SOCP algorithm. It is visible that scattering contributes significantly to reduce the localization error, allowing a high accuracy with a very low number of samples. The best accuracy achieved using only one sample is shown in figure 20(b) - with 4 MTs cooperating it was possible to have an accuracy consistently below 3 m using Tomic SOCP and below 3.5 m using PT SOCP algorithms (slightly higher for the other algorithms, except for the Simple). With 3 MTs cooperating, it is possible to have an accuracy near 3 m using Tomic SOCP algorithm. Therefore, these results show that real-time tracking of remote nodes is possible by running coordinated scattered scan nodes that exchange RSS samples.

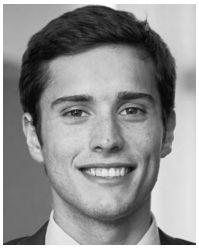
VII. CONCLUSION

This paper compared the performance of several localization algorithms, using real RSS data measurements acquired through smartphones inside a building, considering cooperative and stand-alone operations. It also presented a prototype Android application that implements the best suited non-cooperative algorithm. Our experimental results showed that it is possible for a MT to estimate the position of an AP node inside a building with a bounded error. The work evaluated the accuracy of the odometry module implemented to complement the Android's location service and of the AP's estimated location. They showed that PT-SOCP and Tomic-SOCP trade off an higher accuracy in exchange for an higher processing overhead, compared to the other algorithms tested. On the other hand, when three or more scattered MTs cooperate, it is possible to localize a remote node using a single local sample with a small error for Tomic-SOCP or PT-SOCP.

This paper only studied odometry based smartphone's location measurements, but these can be combined with GNSS based ones. Future work will include the study of alternative mechanisms to handle heterogeneous sets of APs and terminals, namely the use of a more extensive set of reference APs, to further reduce the errors introduced by measurements from different MTs. New localization algorithms are also envisioned, which combine SOCP and LM approaches.

REFERENCES

[1] Y. Zhuang, H. Lan, Y. Li, and N. El-Sheimy, "PDR/INS/WiFi integration based on handheld devices for indoor pedestrian navigation," *Micromachines*, vol. 6, p. 793–812, Jun. 2015.
[2] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of RSS fingerprinting based indoor localization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, vol. 26, Apr./May 2015, pp. 2479–2487.
[3] N. Sirola, "Closed-form algorithms in mobile positioning: Myths and misconceptions," in *Proc. 7th Workshop Positioning Navigat. Commun. (WPNC)*, Mar. 2010, pp. 38–44.
[4] S. Tomic, M. Beko, and R. Dinis,, "RSS-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2037–2050, May 2015.
[5] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*. New York, New York, USA, 2010, p. 173.
[6] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3607–3644, 4th Quart., 2018.
[7] P. A. Zandbergen, "Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning," *Trans. GIS*, vol. 13, pp. 5–25, Jun. 2009.
[8] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, vol. 2, Mar. 2000, pp. 775–784.
[9] S. Tomic, M. Beko, and R. Dinis, "Distributed RSS-AoA based localization with unknown transmit powers," *IEEE Wireless Commun. Lett.*, vol. 5, no. 4, pp. 392–395, Aug. 2016.
[10] S. Tomic, M. Beko, and R. Dinis, "A closed-form solution for RSS/AoA target localization by spherical coordinates conversion," *IEEE Wireless Commun. Lett.*, vol. 5, no. 6, pp. 680–683, Dec. 2016.
[11] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1347–1370, 2nd Quart., 2017.
[12] M. Zhang, Y. Wen, J. Chen, X. Yang, R. Gao, and H. Zhao, "Pedestrian dead-reckoning indoor localization based on OS-ELM," *IEEE Access*, vol. 6, pp. 6116–6129, 2018.
[13] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, 2013, pp. 225–234.
[14] N. Capurso, T. Song, W. Cheng, J. Yu, and X. Cheng, "An Android-based mechanism for energy efficient localization depending on indoor/outdoor conte," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 299–307, Apr. 2016.
[15] L. Wang, L. Feng, and M. Zawodniok, "Ubiquitous tracking using motion and location sensor with application to smartphone," in *Proc. IEEE Int. Conf. Smart Comput.*, May 2017, pp. 1–8.
[16] V. Radu and M. K. Marina, "HiMLoc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced WiFi fingerprinting," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2013, pp. 1–10.
[17] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
[18] M. Liu, S. Huang, G. Dissanayake, and H. Wang, "A convex optimization based approach for pose SLAM problems," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, Oct. 2012, pp. 1898–1903.
[19] D. Pedro *et al.*, "Localization of static remote devices using smartphones," in *Proc. IEEE Veh. Technol. Conf. Spring (VTCs)*, Jun. 2018, pp. 1–5.
[20] D. M. R. Bonick. *Adaptiv: An Adaptive Jerk Pace Buffer Step Detection Algorithm*. Accessed: Jul. 2017. [Online]. Available: https://github.com/danielmurray/adaptiv
[21] A. Developers. (Jul. 2017). *Api guides: Location and sensors - motion sensors*. [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_motion.html
[22] A. Pacha, "Sensor fusion for robust outdoor augmented reality tracking on mobile devices," M.S. thesis, Dept. Institut Für Software Syst. Eng., Augsburg Univ., Minneapolis, MN, USA, 2013.
[23] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, Jan. 2002.
[24] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, May 2008.
[25] K. W. Cheung, H. C. So, W.-K. Ma, and Y. T. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1121–1128, Apr. 2004.
[26] H. Gavin, "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems," Dept. Civil Environ. Eng., Duke Univ., Durham, NC, USA, 2011, pp. 1–15. [Online]. Available: https://scholar.google.com/scholar?cluster=19048121968001100 79&hl=en&as_sdt=5,31&sciodt=0,31



DÁRIO PEDRO received the M.Sc. degree (Hons.) in electrical and computer engineering from the Faculdade de Ciências e Tecnologia Universidade Nova de Lisboa, in 2017. He is currently pursuing the Ph.D. degree. His work is focused in autonomous vehicles and artificial intelligence, where he aims to create an architecture for completely autonomous vehicles, with a high degree of intelligence and capability to adapt to the most unexpected scenarios. He was a Teaching Assistant with the Department of Electrical Engineering, a Researcher with the Instituto de Telecomunicações and a Consultant with Deloitte. In 2018, he joined the Research and Development Department, PDMFC company.

SLAVISA TOMIC received the M.S. degree in traffic engineering according to the postal traffic and telecommunications study program from the University of Novi Sad, Serbia, in 2010, and the Ph.D. degree in electrical and computer engineering from the University Nova of Lisbon, Portugal, in 2017. He is currently an Assistant Professor with the Universidade Lusófona de Humanidades e Tecnologias, Lisbon. His research interests include target localization in wireless sensor networks, and non-linear and convex optimization.

LUIS BERNARDO (M'03) received the Ph.D. degree from the Instituto Superior Técnico, Technical University of Lisbon, Portugal, in 2002, and the Habilitation in telecommunications from the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, in 2013, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering. He is also affiliated as a Researcher with the Instituto de Telecomunicações, Lisbon. His research interests include medium access control and application protocols for wireless communications, cross-layer optimization of wireless systems, routing protocols, and network modeling.

MARKO BEKO received the Dipl. Eng. degree from the University of Belgrade, Belgrade, Serbia, in 2001, and the Ph.D. degree in electrical and computer engineering from the Instituto Superior Tecnico, Lisbon, Portugal, in 2008. He is currently a Full Professor with the Universidade Lusófona de Humanidades e Tecnologias, Portugal. He is also a Researcher with the UNINOVA, Campus da FCT/UNL, Monte de Caparica, Portugal. His current research interests are in the area of signal processing for wireless communications and nonsmooth and convex optimization. He serves as an Associate Editor for the IEEE Access and the *Physical Communication* (Elsevier). He was awarded a Starting Grant under the Investigador FCT Programme of the Portuguese Science and Technology Foundation, in 2016. He is the winner of the 2008 IBM Portugal Scientific Award.

RODOLFO OLIVEIRA (S'04–M'10–SM'15) received the Licenciatura degree in electrical engineering from the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa (UNL), Lisbon, in 2000, the M.Sc. degree in electrical and computer engineering from the Instituto Superior Técnico, Technical University of Lisbon, in 2003, and the Ph.D. degree in electrical engineering from UNL, in 2009. From 2007 to 2008, he was a Visiting Researcher with the University of Thessaly. From 2011 to 2012, he was a Visiting Scholar with the Carnegie Mellon University. He is currently with the Department of Electrical and Computer Engineering, UNL, and is also affiliated as a Researcher with the Instituto de Telecomunicações, where he researches in the areas of wireless communications and networking.

RUI DINIS (S'96–M'00–SM'14) received the Ph.D. degree from the Instituto Superior Técnico (IST), Technical University of Lisbon, Portugal, in 2001, and the Habilitation in telecommunications from the Faculdade de Ciências e Tecnologia (FCT), Universidade Nova de Lisboa (UNL), in 2010.

From 2001 to 2008, he was a Professor with IST. In 2003, he was an Invited Professor with Carleton University, Ottawa, Canada. He was a Researcher with the Centro de Análise e Processamento de Sinal, IST, from 1992 to 2005, and a Researcher with the Instituto de Sistemas e Robótica, from 2005 to 2008. Since 2009, he has been a Researcher with the Instituto de Telecomunicações. He is currently an Associate Professor with FCT-UNL. He has been actively involved in several national and international research projects in the broadband wireless communications area. His research interests include transmission, estimation, and detection techniques.

Dr. Dinis is an Editor of the IEEE Transactions on Wireless Communications, the IEEE Transactions on Communications (Transmission Systems - Frequency-Domain Processing and Equalization), and the IEEE Transactions on Vehicular Technology. He was also a Guest Editor of the *Physical Communications* (Elsevier) (Special Issue on Broadband Single-Carrier Transmission Techniques).

PAULO PINTO (M'98) received the Ph.D. degree in computer science from the University of Kent, Canterbury, and the Diploma degree in electrical engineering from the Instituto Superior Técnico, Lisbon, Portugal. He is currently a Full Professor with the Universidade Nova de Lisboa, Portugal, and a Researcher with the Instituto de Telecomunicações. His current research interests include traffic management, MAC protocols, routing protocols, and software-defined networking.

P. AMARAL (M) received the M.Sc. degree in computer engineering, and the Ph.D. degree in electric and computer engineering from the Universidade Nova de Lisboa, in 2006 and 2013, respectively. He is currently an Assistant Professor with the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, and a Researcher with the Instituto de Telecomunicações, Lisboa. His current research interests include routing modelling, very low delay network queuing, and intelligent network management.