

Hybrid UWB-Inertial TDoA-Based Target Tracking With Concentrated Anchors

Marco Martalò¹, *Senior Member, IEEE*, Simone Perri, Gianmichele Verdano, Francesco De Mola, Francesco Monica, and Gianluigi Ferrari², *Senior Member, IEEE*

Abstract—In this article, hybrid radio/inertial mobile target tracking for accurate and smooth path estimation is considered. The proposed tracking approach builds upon an ultra wide-band (UWB)-based positioning algorithm, based on the linear hyperbolic positioning system (LinHPS), with Time-Difference-of-Arrival (TDoA) processing and anchors concentrated on a single hotspot at the center of the environment where the target moves. First, we design an adaptive radio-based extended Kalman filter (AREKF), which does not require a priori statistical knowledge of the noise in the target movement model and estimates the measurement noise covariance, at each sampling time, according to a proper lookup table (LUT). In order to improve the performance of AREKF, we incorporate inertial data collected from the target and propose three “hybrid” radio/inertial algorithms, denoted as hybrid inertial measurement unit (IMU)-aided radio-based EKF (HIREKF), hybrid noisy control EKF (HNCEKF), and hybrid control EKF (HCEKF). Our results on experimentally acquired paths show that the proposed algorithms achieve an average instantaneous position estimation error on the order of a few centimeters. Moreover, the minimum target path length estimation error, obtained with HCEKF, is on the order of 6% and 1% for two paths with lengths equal to approximately 17 and 46 m, respectively.

Index Terms—Inertial measurement unit (IMU), target tracking, Time Difference of Arrival (TDoA), ultra wideband (UWB).

I. INTRODUCTION

MODERN communication networks require a more and more accurate knowledge of the positions of users and objects to allow context awareness and location-based services, such as navigation, network optimization, augmented reality,

location-sensitive billing, security, etc. This aspect is crucial in Internet of Things (IoT)-based applications, in which devices may be placed in indoor scenarios, where environmental characteristics (e.g., obstructions, multipath, and interferences) may lead to non-line-of-sight (NLOS) propagation conditions, which are detrimental for accurate device localization [1].

Positioning algorithms are based on properly processing the estimated distances between the target and some reference (with known positions) nodes, denoted as *anchors*, from relevant parameters of the used radio technologies, such as Bluetooth, WiFi and cellular (4G/5G) [2]. Examples of these parameters are the received signal strength indicator (RSSI), Angle of Arrival (AoA), Time of Arrival (ToA) and Time Difference of Arrival (TDoA) [2]. In particular, TDoA exploits the difference, in terms of signal propagation times from the target (transmitting a beaconing signal) to the anchors, provided that the anchors are synchronized with each other. An appealing technology for localization is ultra wideband (UWB), which can provide, jointly with TDoA processing, accurate position estimation, and limited vulnerability to changes of environmental conditions in many scenarios (see [3] and references therein).

While existing algorithms allow to determine the position of a static user, proper tracking algorithms are needed if a mobile user is considered [4]. ToA- and AoA-based tracking with NLOS mitigation is discussed in [5]. In [6], target tracking is performed without using information from fixed anchors, odometers, or inertial measurement units (IMUs), but leveraging position-related information contained in multipath components of the received radio signal. In [7], TDoA-based target tracking with UWB communications is considered, focusing on measurement noise decorrelation for performance improvement. The joint use of IMU and UWB measurements in Kalman-based target tracking is also considered in the literature. In [8], the joint use of IMU and Time of Flight (ToF) data is proposed to improve the robustness and accuracy; in particular, a constant acceleration model for the target movement is considered. ToF and IMU data are also exploited in [9], where experimental results are obtained using the InvenSense MPU-9250 IMU [10] and DWM1000 UWB radio node [11]. A similar scenario is considered in [12]. In [13], TDoA and AoA data from UWB UbiSense devices [14] are fused with InvenSense MPU-6050 IMU [15] data and the corresponding performance is investigated.

Unlike the previous literature, which focuses on “classical” scenarios where the anchors are distributed along the perimeter of the area to be monitored and the target lies inside this area,

Manuscript received 3 February 2022; revised 5 January 2023; accepted 28 February 2023. Date of publication 6 March 2023; date of current version 7 July 2023. This article was presented in part at the IEEE International Conference on Computing Communication and Security (ICCCS), Rome, Italy, October 2019 [DOI: 10.1109/CCCS.2019.8888099]. (Corresponding author: Marco Martalò.)

Marco Martalò was with the Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy. He is now with the Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy, and also with the Research Unit of Cagliari, Consorzio Nazionale Interuniversitario per le Telecomunicazioni, 43124 Parma, Italy (e-mail: marco.martalò@unica.it).

Simone Perri was with the R&D & Innovation Department, Elettric80 S.p.A., 42030 Viano, Italy. He is now with Able Tech SrL, 25017 Lonato del Garda, Italy (e-mail: simone.perri23@gmail.com).

Gianmichele Verdano, Francesco De Mola, and Francesco Monica are with the R&D & Innovation Department, Elettric80 S.p.A., 42030 Viano, Italy (e-mail: verdano.g@elettric80.it; demola.f@elettric80.it; monica.f@elettric80.it).

Gianluigi Ferrari is with the IoT Lab, Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy, and also with the Research Unit of Parma, Consorzio Nazionale Interuniversitario per le Telecomunicazioni, 43124 Parma, Italy (e-mail: gianluigi.ferrari@unipr.it).

Digital Object Identifier 10.1109/JIOT.2023.3252915

we consider anchors placed on a single “hotspot” with limited dimensions, whereas the target to be localized lies *outside* the polytope identified by the anchors, as proposed in [16]. This scenario may be relevant in smart warehouses, where human operators and automatic guided vehicles (AGVs) coexist. In this case, the anchors may be placed on the AGV to localize human operators around it. The use of this architecture allows to avoid situations in which the anchors are in NLOS conditions with the target, thus, making the localization process ineffective.

While in [16] and [17] a TDoA-based 3-D radio positioning algorithm with anchors concentrated on a single hotspot is designed for static target detection, in this article we extend it to a scenario with a mobile target (which moves around the hotspot). The key idea is to apply standard solutions, e.g., those based on Kalman filtering, to exploit inertial information from the mobile target to improve localization accuracy. In particular, we focus on 2-D target tracking.¹ Since the accuracy of the considered radio positioning algorithm varies with the relative distance and angle between the hotspot and the target [16], we consider two classes of algorithms.

- 1) First, we propose an adaptive radio-based extended Kalman filter (AREKF), which does not require to fix a priori statistical knowledge of the noise in the target movement model (which is indeed adaptively estimated) and the measurement noise covariance is estimated at each time epoch according to a proper lookup table (LUT).
- 2) In order to improve the performance of the AREKF, we extend this algorithm by taking into account inertial data collected from the target [18], [19]. Depending on how inertial information (representative of the target movement) is combined with radio-based tracking, we will consider IMU-aided or IMU-controlled radio-based tracking.

Even if Kalman filtering is a well-established technique, its extension to the considered architecture with concentrated anchors on the hotspot is novel—this implies that a direct comparison with other approaches in the literature cannot easily be performed. Our results show that the proposed algorithms can track the true target path with an average estimation error on the order of a few centimeters. In particular, the hybrid IMU-controlled radio-based tracking algorithm, denoted as HCEKF, is the best solution among those investigated.

The remainder of this article is organized as follows. In Section II, the system model is discussed. In Section III, the standard radio-based Kalman filter is presented. In Section IV, the AREKF algorithm is proposed. Then, in Section V, the hybrid radio/inertial solutions are presented, with a discussion on potential system implementations. In Section VI, numerical results are shown. Finally, concluding remarks are given in Section VII.

II. SYSTEM MODEL

A. Reference Architecture

The considered reference scenario is shown in Fig. 1: a target to be localized moves in a monitored environment,

¹The extension to 3-D tracking is an interesting research extension, but goes beyond the scope of this article.

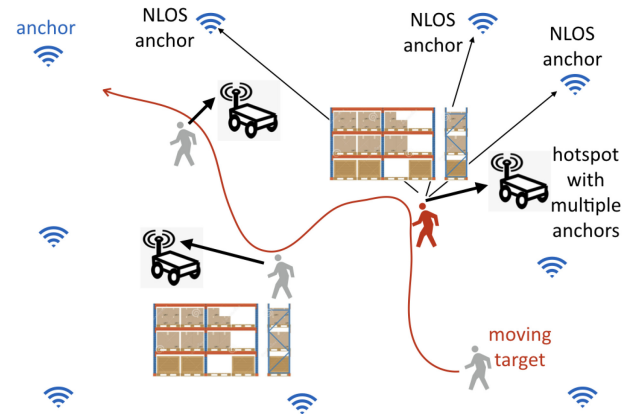


Fig. 1. Considered reference scenario.

e.g., a human operator walks inside a large warehouse with AGVs. In this article, we consider a single target to be tracked, but our TDoA-based localization strategy can be extended to multiple targets, provided that they can be identified from the exchanged packets.²

Standard localization approaches envision the distribution of several anchors in the environment. However, this may not be feasible in large environments, such as warehouses. Moreover, some of the anchors may not be in direct visibility with the target due to the presence of possibly large obstacles, e.g., goods’ shelves. Such anchors, in NLOS communication conditions with respect to the target, may be detrimental for localization accuracy and, in some cases, make the entire system unusable. To this end, in this article, we assume that the anchors are concentrated on a single hotspot [16]. The hotspot comprises a sufficient number of anchors to locate the target. Moreover, the hotspot can be opportunistically placed in the monitored environment, so that at least one set of anchors is in direct visibility line of sight (LOS) with the target.

Note that we do not focus on specific requirements on target–hotspot distances. In fact, the proposed architecture allows to track targets also in very large environments, by simply inserting multiple hotspots at proper positions. Moreover, our preliminary results in [16] show sufficient localization accuracy even for relatively large target–hotspot distances (on the order of a few tens of meters).

B. Three-Dimensional Topology

The system model is shown in Fig. 2, where a representative target trajectory (solid black line) is also shown. The target The target position and velocity at time epochs $t = kT_s$, $k = 0, 1, \dots$ (being T_s , dimension: [s], the sampling interval) can be denoted, in the 3-D space, as follows³:

$$\mathbf{p}_k^{(3d)} = [p_{x,k}, p_{y,k}, p_{z,k}]^T \quad (1)$$

$$\mathbf{v}_k^{(3d)} = [v_{x,k}, v_{y,k}, v_{z,k}]^T = \frac{\mathbf{p}_k^{(3d)} - \mathbf{p}_{k-1}^{(3d)}}{T_s}. \quad (2)$$

²The use of ToA-based algorithms incurs higher latency, e.g., due to the use of TWR techniques.

³We remark that, in the remainder of this article, boldface lowercase and uppercase letters denote vectors and matrices, respectively.

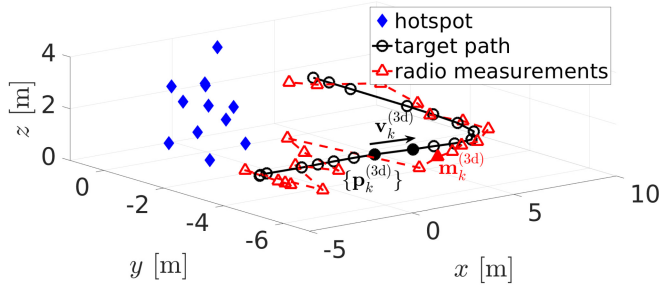


Fig. 2. System model and illustrative trajectory (true and estimated).

This approach can be extended to take into account the acceleration which the target is subject to. However, our results, not shown here for lack of space, show that no significant performance improvement is obtained by considering the acceleration.

The target has to be detected by a hotspot on which N anchor nodes are placed at the following fixed positions:

$$\mathbf{a}_i^{(3d)} = [a_{x,i}, a_{y,i}, a_{z,i}]^T \quad i = 1, \dots, N. \quad (3)$$

The Center of Gravity (CoG) of the hotspot is denoted as

$$\bar{\mathbf{a}}^{(3d)} = \frac{1}{N} \sum_{i=1}^N \mathbf{a}_i^{(3d)}. \quad (4)$$

The 3-D distance $r_{i,k}$ between the i th anchor and the k th target position can be expressed as follows:

$$r_{i,k} = \left\| \mathbf{p}_k^{(3d)} - \mathbf{a}_i^{(3d)} \right\| = c \tau_{i,k} \quad (5)$$

where $\|\cdot\|$ denotes the Euclidean norm, $c \simeq 3 \cdot 10^8$ m/s is the speed of light, and $\tau_{i,k}$ is the ToF between the i th anchor and the k th target position.

At each time epoch, the target sends a UWB beacon and the hotspot, running a TDoA-based radio positioning algorithm on the signals received by the anchors, derives a position estimate denoted as

$$\mathbf{m}_k^{(3d)} = [m_{x,k}, m_{y,k}, m_{z,k}]^T. \quad (6)$$

C. TDoA Measurements

TDoA algorithms exploit the difference (rather than absolute values) between the ToFs (from the target) measured at different anchors upon the reception of the beacon transmitted by the target.⁴ Even if ToF-based methods can be potentially more accurate than TDoA-based methods, they require that the anchors, besides being synchronized among themselves, are also synchronized with the target. Another possible solution is the use of two-way ranging (TWR), which, however, is not scalable to scenarios where multiple targets have to be tracked. On the other hand, in TDoA-based methods, only the anchors need to be synchronized with each other: this can be effectively obtained in the considered scenario, since they are physically close. This is also confirmed in [20].

⁴Note that the number of anchors receiving the transmitted beacon may be smaller than N , e.g., due to environmental interference and obstacles. However, in the remainder of this article, we will assume that all the anchors are receiving a beacon and can collaborate to target localization.

Let us consider the k th time epoch. Denoting $\mathbf{a}_{j,k}^{(3d)}$ ($j \in \{1, 2, \dots, N\}$) as the first anchor receiving the transmitted beacon, the i th TDoA $\Delta\tau_{i,k}$ ($i \neq j$) can be defined as the difference between the ToF $\tau_{i,k}$ measured by the i th anchor and the ToF τ'_k measured by a reference anchor, i.e.,

$$\Delta\tau_{i,k} = \tau_{i,k} - \tau'_k \quad i \in \{1, \dots, N\}/j. \quad (7)$$

Note that, by definition, $\Delta\tau_{i,k} > 0$. In practice, however, only measured TDoAs, rather than the true TDoAs, are available. These TDoA measurements are denoted as

$$\tilde{\Delta}\tau_{i,k} = \tilde{\tau}_{i,k} - \tilde{\tau}'_k = \frac{\tilde{r}_{i,k} - \tilde{r}'_k}{c} \quad i \in \{1, \dots, N\}/j \quad (8)$$

where the following notation will be used in the remainder of this article: $\tilde{\chi}$ will denote the measurement of the true value χ . One can easily observe that the accuracy of (7) strictly depends on the clock synchronization between the anchors, i.e., the ToFs $\tau_{i,k}$ and τ'_k need to refer the same time axis. In the following, we will assume perfect synchronization among anchors—the impact of imperfect time synchronization among anchors of UWB nodes in TDoA-based schemes is discussed in [16] and [20]. The results in [16] show that a 1-ns accuracy between anchors is needed to achieve good positioning accuracy in the scenario of interest for this article. Almost perfect synchronization can be practically achieved by wire-connecting the anchors to a central unit, which acts as a very accurate clock source and distributes a signal synchronization to the anchors [21]. Other solutions are based on wireless communications among the anchors to achieve the desired synchronization accuracy (see [22] and references therein).

We remark that the ToFs are computed with reference to the 3-D distances traveled by radio signals.

D. UWB Communication Model

The specific statistical characterization of the measured ToFs $\{\tilde{\tau}_{i,k}\}$ in (8) depends on UWB signaling and channel status. All the target–anchor communication links are assumed to be in LOS. In [16], the following simple, yet accurate and experimentally validated, model for noisy ToF estimation in UWB LOS links is considered

$$\tilde{\tau}_{i,k} = \tau_{i,k} + \delta_{i,k} \quad (9)$$

where $\delta_{i,k} \sim \mathcal{N}(\mu_{i,k}, \sigma_{i,k}^2)$. According to [16], $\mu_{i,k}$ and $\sigma_{i,k}$ can be approximated as linear functions of $\tau_{i,k}$, i.e.,

$$\mu_{i,k} \simeq q_1 \tau_{i,k} + q_2/c \quad (10)$$

$$\sigma_{i,k} \simeq s_1 \tau_{i,k} + s_2/c. \quad (11)$$

Expressions (10) and (11) are intuitive, since the longer the target–anchor distance, the higher the expected measurement noise.

Following the approach proposed in [16], here, we consider $q_1 = 0.0042$, $q_2 = 0.01$ m, $s_1 = -0.0003$, and $s_2 = 0.0302$ m. These values are obtained using a standard least-square (LS) estimation model based on experimental data collected with decaWave DW1000 UWB nodes [11], which are currently considered for experimental implementation of the proposed localization system.

E. Two-Dimensional Tracking

Even though we refer to a 3-D topology (as described in Section II-B), in the remainder of this article, we are interested in 2-D tracking. This is meaningful in a scenario where the target is moving on a flat surface and the goal is to reconstruct its 2-D trajectory.

In the following, unless there is a superscript $(\cdot)^{(3d)}$, a vector will be 2-D and its components will refer to x and y coordinates. In particular, the target tracking algorithms presented in Sections III–V will refer to 2-D scenarios.

F. IMU Measurement Model

We assume that the target is equipped with an IMU which allows to derive an estimate $\hat{\mathbf{v}}_k$ of the true instantaneous velocity \mathbf{v}_k at the k th time epoch.⁵ Estimation of the target velocity vector can be tackled, as in standard inertial navigation problems, by resorting to in-sensor processing at the IMU and leveraging kinematic laws. A possible (but not the only one) method in the literature can be found in [23].

The use of the velocity estimate may be expedient to improve the accuracy of the tracking algorithm, as explained in Section V. The estimated target velocity can be modeled as

$$\hat{\mathbf{v}}_k = \mathbf{v}_k + \mathbf{q}_k \quad (12)$$

in which $\mathbf{q}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\text{IMU}})$ is the IMU measurement noise, assumed to be additive Gaussian with zero mean and the following covariance matrix:

$$\mathbf{R}_{\text{IMU}} = \text{diag}\left(\left[\sigma_{\text{IMU}}^2 \quad \sigma_{\text{IMU}}^2\right]\right) \quad (13)$$

where $\text{diag}(\boldsymbol{\xi})$ is a diagonal matrix with elements of vector $\boldsymbol{\xi}$ on the main diagonal; and σ_{IMU}^2 is the per-component noise variance (assumed, for simplicity, to be the same for both components). We implicitly assume that the reference systems of the IMU and of the radio positioning algorithm are the same, i.e., the orientation estimate provided by the IMU is error-free. An accurate experimental estimation of the velocity vector is an open research question. However, in Section VI, we will provide numerical results with various values of σ_{IMU} to encompass the cases of good or poor velocity estimation. The impact of an imperfect orientation estimation on the performance will be the subject of future work [24].

III. (STANDARD) RADIO-BASED TARGET TRACKING

We now briefly recall the basics of a standard radio-based EKF, whose goal is to provide smooth estimation of the true positions $\{\mathbf{p}_k\}$ starting from the radio-based position measurements $\{\mathbf{m}_k\}$ [25]. The linearized equations that describe a Kalman filter, namely, the state transition equation and the state observation equation, are as follows [25]:

$$\mathbf{s}_{k+1} = \mathbf{A}\mathbf{s}_k + \mathbf{w}_k \quad (14)$$

$$\mathbf{m}_k = \mathbf{H}\mathbf{s}_k + \mathbf{n}_k \quad (15)$$

where the main terms can be described as follows.

⁵We remark, for instance, that, according to Section II-E, \mathbf{v}_k is the 2-D velocity in the x – y plane, i.e., $\mathbf{v}_k = [v_{x,k}, v_{y,k}]^T$.

- 1) The vector \mathbf{s}_k denotes a properly defined state to be estimated (at time epoch k). In this case, we define a 4-D state as the combination of the position and the velocity at each time epoch, i.e.,

$$\mathbf{s}_k = [\mathbf{p}_k^T, \mathbf{v}_k^T]^T. \quad (16)$$

- 2) The matrix \mathbf{A} is the state transition matrix, which describes the state update model. In the remainder of this article, we will consider a constant velocity model, where the state transition matrix \mathbf{A} , of size 4×4 , has the following block matrix form:

$$\mathbf{A} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \otimes \mathbf{I}_2 \quad (17)$$

in which \otimes denotes the Kronecker product between matrices [26] and \mathbf{I}_n is the notation for the identity matrix of size n .

- 3) The vector \mathbf{w}_k is the additive noise affecting the movement with respect to the considered model given by matrix \mathbf{A} . The model noise \mathbf{w}_k is a 4-D Gaussian vector with zero mean and the following covariance matrix:

$$\mathbf{Q} = \text{diag}\left(\left[0 \quad 0 \quad \sigma_Q^2 \quad \sigma_Q^2\right]\right). \quad (18)$$

Therefore, the model (14) is representative of a target moving with a velocity which can vary around a constant value affected by a (Gaussian) noise characterized by its standard deviation σ_Q .

- 4) The vector \mathbf{m}_k has, as elements, the measurements of the components of state \mathbf{s}_k available at the input of the EKF at time epoch k . The measurements correspond to position and velocity estimates obtained with a proper positioning algorithm (as will be described at the beginning of Section IV). The state observation equation is characterized by the matrix \mathbf{H} , of size 2×4 , which can be written as

$$\mathbf{H} = [1 \ 0] \otimes \mathbf{I}_2. \quad (19)$$

- 5) The vector $\mathbf{n}_k = [n_{x,k}, n_{y,k}]^T$ is a 2-D additive noise vector which affects the considered measurements (i.e., radio-based position estimates). This vector is modeled as Gaussian with zero mean and the following covariance matrix:

$$\mathbf{R} = \text{diag}\left(\left[\sigma^2 \quad \sigma^2\right]\right) \quad (20)$$

where $\mathbf{R} = \mathbb{E}[\mathbf{n}_k \mathbf{n}_k^T]$ is the covariance matrix of the measurement noise, being $\mathbb{E}[\cdot]$ the mean value operator.

The standard EKF algorithm for dynamic state estimation involves two main steps: 1) prediction and 2) update [25]. In the *prediction* step, the current state is predicted (on time) and the a priori estimate of the error covariance is obtained for the next step—these estimates will be identified, notation-wise, by a reversed hat symbol ($\hat{\cdot}$). The *update* step is then responsible for a feedback correction that incorporates a new measurement into the a priori estimate to obtain an improved *a posteriori* estimate. The updated estimate will be identified by a hat symbol ($\hat{\cdot}$).

The *prediction* step at epoch k is characterized by the following operations:

$$\check{s}_k = \mathbf{A}\hat{s}_{k-1} \quad (21)$$

$$\check{\mathbf{F}}_k = \mathbf{A}\hat{\mathbf{F}}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (22)$$

where the symbol \mathbf{F} is used to estimate the state covariance matrix, namely: $\hat{\mathbf{F}}_{k-1}$ is the *updated* estimate at the previous epoch and $\check{\mathbf{F}}_k$ is the *predicted* estimate at the current epoch. In the *update* step, the following operations are carried out:

$$\mathbf{G}_k = \check{\mathbf{F}}_k\mathbf{H}^T(\mathbf{H}\check{\mathbf{F}}_k\mathbf{H}^T + \mathbf{R})^{-1} \quad (23)$$

$$\hat{\mathbf{F}}_k = (\mathbf{I}_4 - \mathbf{G}_k\mathbf{H})\check{\mathbf{F}}_k \quad (24)$$

$$\hat{s}_k = \check{s}_k + \mathbf{G}_k(\mathbf{m}_k - \mathbf{H}\check{s}_k) \quad (25)$$

where \mathbf{G}_k is referred to as the Kalman gain (at epoch k). The recursion is initialized with $\hat{\mathbf{F}}_0 = \mathbf{I}_4$, $\hat{\mathbf{p}}_0 = \mathbf{m}_0$ (\mathbf{m}_0 being the first radio-based measured position), and $\hat{\mathbf{v}}_0 = \mathbf{0}_2$, where $\mathbf{0}_n$ is the notation for the all-zero vector of size n .

IV. ADAPTIVE RADIO-BASED TARGET TRACKING

The radio-based position estimates $\{\mathbf{m}_k\}$ fed at the input of the EKF, as explained in Section III, correspond to the x - y coordinates of the position estimates coming from a TDoA-based 3-D positioning algorithm (see [16] for more details). In particular, we rely on the linear hyperbolic positioning system (LinHPS) algorithm [27], which provides a position estimate as an LS solution of the following linear system of equations:

$$\mathbf{m}_k^{(3d)} = \mathbf{E}_k^+ \mathbf{b}_k \quad (26)$$

where $\mathbf{m}_k^{(3d)}$ is the 3-D position estimate; \mathbf{E}_k and \mathbf{b}_k are proper matrix and vector containing the coordinates of anchors and the range estimates between them and the target; and \mathbf{E}_k^+ represents the pseudoinverse of \mathbf{E}_k , obtained by means of the Moore–Penrose pseudoinversion operation [26]— \mathbf{E}_k is not, in general, a square matrix.

The matrix \mathbf{R} in (23) characterizes the estimation error in the 3-D TDoA-based UWB positioning algorithm: for this reason, we refer to \mathbf{n}_k as “measurement” noise. In [16], we have shown that, for the LinHPS algorithm, the position error does not have zero mean and the variance is not constant, as it depends on the relative position (in terms of distance and angle) between the hotspot and the target. In particular, the longer the distance, the higher the noise intensity [16].

According to [16], the position error can then be modeled as $\mathbf{n}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{R}_k)$, with

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}(d_k, \theta_k) \quad (27)$$

$$\mathbf{R}_k = \begin{bmatrix} \sigma_x^2(d_k, \theta_k) & \rho_k \sigma_x(d_k, \theta_k) \sigma_y(d_k, \theta_k) \\ \rho_k \sigma_x(d_k, \theta_k) \sigma_y(d_k, \theta_k) & \sigma_y^2(d_k, \theta_k) \end{bmatrix} \quad (28)$$

where $d_k \triangleq |\mathbf{p}_k - \bar{\mathbf{a}}|$ and $\theta_k \triangleq \angle \mathbf{p}_k - \bar{\mathbf{a}}$ are the true 2-D distance and the true angle between the target and the hotspot CoG, respectively; and $\rho_k = \rho(d_k, \theta_k)$ is the correlation coefficient between the errors on the x and y coordinates, i.e.,

$$\rho_k = \frac{\mathbb{E}\{[n_{x,k} - \mu_x(d_k, \theta_k)][n_{y,k} - \mu_y(d_k, \theta_k)]\}}{\sigma_x(d_k, \theta_k) \sigma_y(d_k, \theta_k)}$$

which depends, in general, on d_k and θ_k .

A. Offline LUT Construction

The parameters $\boldsymbol{\mu}_k$ and \mathbf{R}_k can be estimated offline, through a preliminary calibration phase for different distances and angles, and properly stored in an LUT for online use by the EKF, as explained in the following section. To this end, the target is placed at various positions characterized by different distances and angles from the hotspot. For each target position, N_m 2-D position estimates $\{\mathbf{m}_k^{(1)}, \mathbf{m}_k^{(2)}, \dots, \mathbf{m}_k^{(N_m)}\}$ of the true position \mathbf{p}_k are collected. The j th (vector) position error (with sign) can be written as

$$\mathbf{e}_k^{(j)} = \mathbf{m}_k^{(j)} - \mathbf{p}_k \quad j = 1, 2, \dots, N_m \quad (29)$$

where the difference is performed elementwise. Assuming a Gaussian distribution of the position error (the same for each consecutive estimate corresponding to the same true position), one can compute the sample mean and the standard deviation of the position error as follows [28]:

$$\boldsymbol{\mu}_k = \frac{1}{N_m} \sum_{j=1}^{N_m} \mathbf{e}_k^{(j)} \quad (30)$$

$$\boldsymbol{\sigma}_k = \sqrt{\frac{1}{N_m - 1} \sum_{j=1}^{N_m} [\mathbf{e}_k^{(j)} - \boldsymbol{\mu}_k]^2} \quad (31)$$

where $\boldsymbol{\mu}_k = [\mu_{x,k}, \mu_{y,k}]^T$ and $\boldsymbol{\sigma}_k = [\sigma_{x,k}, \sigma_{y,k}]^T$. The correlation coefficient can be estimated by means of the sample Pearson correlation coefficient as [28, Ch. 4]

$$\rho_k = \frac{1}{N_m - 1} \sum_{i=1}^{N_m} \left[\frac{e_{x,k}^{(i)} - \mu_{x,k}}{\sigma_{x,k}} \right] \left[\frac{e_{y,k}^{(i)} - \mu_{y,k}}{\sigma_{y,k}} \right]. \quad (32)$$

From $\boldsymbol{\sigma}_k$ in (31) and ρ_k in (32), one can derive \mathbf{R}_k in (28). Note that the LUT parameters are accurately estimated if the number N_m of collected measurements is sufficiently large. This comes at the price of an estimation delay. However, since the LUT construction is performed offline, this is not an issue and N_m can be set to a sufficiently large value to guarantee convergence of (30)–(32).

B. Online Operations

The standard EKF has to be properly modified to take into account the statistical characterization of the position error derived in the previous section. In particular, we propose to use the LUT to obtain, at each time epoch, an estimate of the covariance matrix $\hat{\mathbf{R}}_k$ in the following way. Since the exact target–hotspot distance and angle (i.e., the polar coordinates) are not available in practice, the following estimates of these quantities, based on the current position estimate \mathbf{m}_k output by the LinHPS algorithm, are considered

$$\hat{d}_k = |\mathbf{m}_k - \bar{\mathbf{a}}| \quad (33)$$

$$\hat{\theta}_k = \angle \mathbf{m}_k - \bar{\mathbf{a}}. \quad (34)$$

At this point, the LUT is first read at coordinates given by \hat{d}_k and $\hat{\theta}_k$, obtaining⁶

⁶Note that the LUT is obtained at discretized values of d_k and θ_k . If the LUT is not available at distance \hat{d}_k and angle $\hat{\theta}_k$, proper linear interpolation among adjacent available values is performed.

$$(\hat{\boldsymbol{\mu}}_k, \hat{\mathbf{R}}_k) = \text{LUT}(\hat{d}_k, \hat{\theta}_k). \quad (35)$$

Since the position estimate \mathbf{m}_k has (from the LUT) a nonzero mean error, the LUT reading at $(\hat{d}_k, \hat{\theta}_k)$ does not provide the correct statistical characterization. In particular, the true target position \mathbf{p}_k is better characterized by a modified estimate \mathbf{m}'_k , which can be computed as

$$\mathbf{m}'_k = \mathbf{m}_k - \hat{\boldsymbol{\mu}}_k.$$

One can then compute the following polar coordinates corresponding to \mathbf{m}'_k :

$$\hat{d}'_k = |\mathbf{m}'_k - \bar{\mathbf{a}}| \quad (36)$$

$$\hat{\theta}'_k = \angle \mathbf{m}'_k - \bar{\mathbf{a}} \quad (37)$$

and, then, read the corresponding LUT entry, i.e.,

$$(\hat{\boldsymbol{\mu}}'_k, \hat{\mathbf{R}}'_k) = \text{LUT}(\hat{d}'_k, \hat{\theta}'_k). \quad (38)$$

The parameters $\hat{\boldsymbol{\mu}}'_k$ and $\hat{\mathbf{R}}'_k$ provide a more accurate statistical characterization of the position error. At this point, the Kalman gain computation in (23) can be modified as follows:

$$\mathbf{G}_k = \check{\mathbf{F}}_k \mathbf{H}^T (\mathbf{H} \check{\mathbf{F}}_k \mathbf{H}^T + \hat{\mathbf{R}}'_k)^{-1}. \quad (39)$$

Moreover, the update (25) can be modified as

$$\hat{\mathbf{s}}_k = \check{\mathbf{s}}_k + \mathbf{G}_k (\mathbf{m}''_k - \mathbf{H} \check{\mathbf{s}}_k) \quad (40)$$

where $\mathbf{m}''_k = \mathbf{m}_k - \hat{\boldsymbol{\mu}}_k$ is a corrected (zero-mean) version of \mathbf{m}_k . The final estimated position by AREKF is $\hat{\mathbf{s}}_k$.

The adaptive estimation of the covariance matrix of the movement model is obtained according to the method presented in [29], where an iterative estimation is proposed on the basis of the following innovation term:

$$\check{\mathbf{e}}_k = \mathbf{m}_k - \mathbf{H} \check{\mathbf{s}}_k. \quad (41)$$

In particular, following the approach in [29], one obtains

$$\hat{\mathbf{Q}}_k = \lambda \hat{\mathbf{Q}}_{k-1} + (1 - \lambda) \mathbf{G}_k \check{\mathbf{e}}_k \check{\mathbf{e}}_k^T \mathbf{G}_k^T \quad (42)$$

where $\lambda \in [0, 1]$ is a proper forgetting factor.

Since the above described approach uses the radio-based EKF scheme introduced in Section III with adaptive estimation of the covariance matrices, we refer to it as adaptive radio-based EKF (ARKEF). A complete block diagram of the AREKF algorithm is shown in Fig. 3.

V. HYBRID INERTIAL/RADIO-BASED TARGET TRACKING

We now propose an improvement of the AREKF derived in Section IV by leveraging the use of inertial measurements, independent of the radio ones, according to the IMU measurement model in Section II-F. Depending on how inertial information is combined with radio-based tracking, we will consider two classes of hybrid radio/inertial tracking algorithms: 1) IMU-aided (Section V-A) or 2) IMU-controlled (Section V-B).

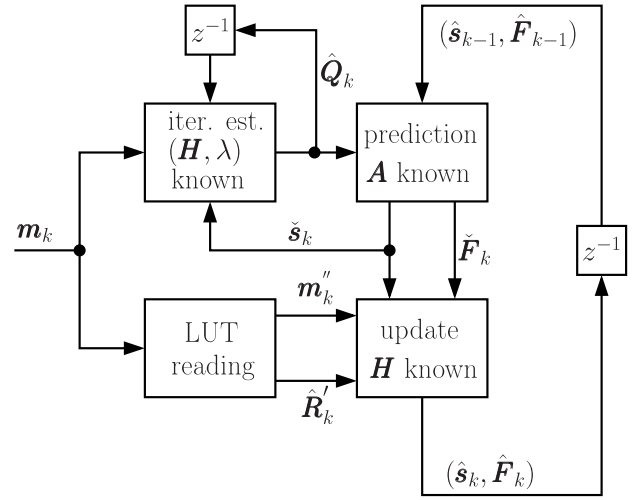


Fig. 3. Block diagram of the AREKF algorithm.

A. IMU-Aided Radio-Based EKF

A heuristic hybrid solution, denoted as hybrid IMU-aided radio-based EKF (HIREKF), can be obtained as follows. First, one can use the AREKF in Section IV and, then, correct the AREKF-based position estimate using the IMU-based position estimate (predicted by the velocity vector). In particular, denoting as $\hat{\mathbf{p}}_{k,\text{radio}}$ the position estimate provided by the update step (40) in AREKF,⁷ the final position estimate can be written as

$$\hat{\mathbf{p}}_k = \alpha \hat{\mathbf{p}}_{k,\text{radio}} + (1 - \alpha) \hat{\mathbf{p}}_{k,\text{IMU}} \quad (43)$$

where $\alpha \in [0, 1]$ is a proper weight; $\hat{\mathbf{p}}_{k,\text{radio}}$ is obtained using the AREKF with $\hat{\mathbf{p}}_{k-1}$ used as $\hat{\mathbf{s}}_{k-1}$ in (21); and $\hat{\mathbf{p}}_{k,\text{IMU}}$ is the position estimate predicted by the IMU-based estimated velocity $\hat{\mathbf{v}}_k$ starting from the last overall position estimate, i.e.,

$$\hat{\mathbf{p}}_{k,\text{IMU}} = \hat{\mathbf{p}}_{k-1} + \mathbf{B} \hat{\mathbf{v}}_k \quad (44)$$

being $\mathbf{B} = \text{diag}([T_s, T_s])$.

In other words, the HIREKF approach starts from the last position estimate $\hat{\mathbf{p}}_{k-1}$ and derives two possible position estimates: one from the AREKF ($\hat{\mathbf{p}}_{k,\text{radio}}$) and the other by integrating the velocity vector over the sampling interval ($\hat{\mathbf{p}}_{k,\text{IMU}}$). The final position estimate can be computed as a weighed average of these two estimates as in (43), where the coefficient α can be tuned according to the accuracy of the two measurements. If the IMU has low reliability (e.g., due to poor orientation estimation), the AREKF should weigh more and one should set $\alpha > 0.5$; if, on the other hand, the AREKF is not reliable (e.g., due to the presence of obstructions in the communication links that degrades the LinHPS estimate), then one should set $\alpha < 0.5$. This tuning can be typically performed offline on the basis of the measurement accuracy provided by the device constructor. In Section VI, we will analyze a scenario where no reliability information is a priori available and, therefore, α is set to 0.5, i.e., an arithmetic

⁷Note that $\hat{\mathbf{p}}_{k,\text{radio}}$ corresponds to $\hat{\mathbf{p}}_k$ in Section IV and that $\hat{\mathbf{p}}_k$ is composed by the first two components of the state $\hat{\mathbf{s}}_k$ estimated by the AREKF, according to the block diagram in Fig. 3 and the final equation (40).

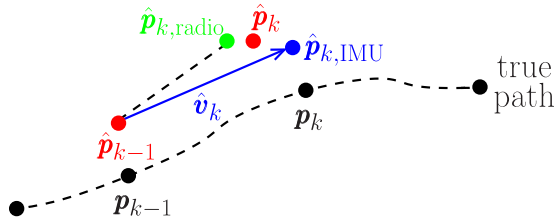


Fig. 4. Pictorial description of a step of the HIREKF algorithm.

average is performed in (43). A pictorial description of a step of the HIREKF algorithm is shown in Fig. 4.

One may observe that (43) and (44) do not depend on the parameter σ_{IMU} , which, therefore, does not need to be accurately characterized from a statistical point of view. Considering adaptive setting of the value of α , this approach may be helpful when the target cannot be radio localized, e.g., in an indoor scenario with poor (NLOS) UWB connectivity. In this case, an IMU-based estimate could still be obtained and, by setting $\alpha = 0$, the algorithm could rely on it. We remark that a purely inertial tracking could be used for a limited amount of time to prevent inaccurate estimation due to the drift problem, which should be properly taken into account [24]. Tuning can also be performed online, if real-time estimates of the accuracies of UWB and IMU measurements are available. Adaptive selection of the value of α is an interesting research direction which, however, goes beyond the scope of this article.

B. Radio-Based EKF With IMU Control

The linearized Kalman filter can be extended to take into account the presence of an input control as follows:

$$\mathbf{s}_{k+1} = \mathbf{A}\mathbf{s}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \quad (45)$$

$$\mathbf{m}_k = \mathbf{H}\mathbf{s}_k + \mathbf{n}_k \quad (46)$$

where \mathbf{u}_k is the control variable and \mathbf{B} is a properly defined matrix that may also include known disturbances [25]. This Kalman filtering can be applied to target tracking provided that \mathbf{u}_k can include external inputs that can control the target movement. In our case, the state can be defined as the target position, i.e., $\mathbf{s}_k = \mathbf{p}_k$, and the control may be the velocity \mathbf{v}_k . Therefore, the state transition matrix becomes $\mathbf{A} = \mathbf{I}_2$, whereas, by resorting to a constant velocity model, \mathbf{B} is defined as in Section V-A. The model noise, of size 2×1 , is $\mathbf{w}_k \sim (\mathbf{0}, \mathbf{Q}_{\text{mod}})$, where, in this case, the covariance matrix \mathbf{Q}_{mod} described the uncertainty of the state, i.e., the target position, in the considered constant velocity model. The measurement model only takes into account the position estimates given by the radio positioning algorithm. Therefore, it holds that $\mathbf{H} = \mathbf{I}_2$ and $\mathbf{n}_k \sim (\mathbf{0}, \hat{\mathbf{R}}'_k)$. The rationale behind this approach is that the linearized Kalman filter can provide reliable estimates when relying on a well-measured control, since the constant velocity model becomes accurate.

A general block diagram of the Kalman algorithm with IMU control is shown in Fig. 5. In the following, we will derive the algorithm with two possible instances, i.e., using different equations in the prediction and update stages.

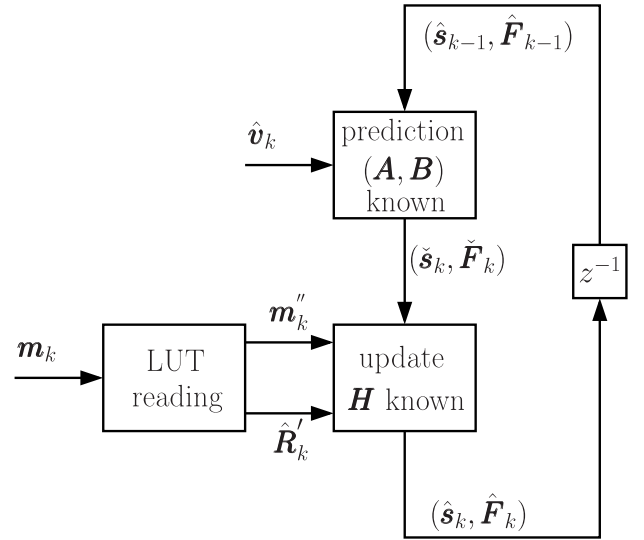


Fig. 5. General block diagram of the Kalman-based algorithm with IMU control.

1) *Instance 1 (HNCEKF)*: If the control variable is noisy (as in our case with $\mathbf{u}_k = \hat{\mathbf{v}}_k$), one can resort to the approach in [30], where a robust unbiased linear Kalman filter with noisy control input is derived. We refer to this approach as hybrid noisy control EKF (HNCEKF). The prediction step is characterized by the following operations:

$$\check{\mathbf{s}}_k = \mathbf{A}\hat{\mathbf{s}}_{k-1} + \mathbf{B}\hat{\mathbf{v}}_k \quad (47)$$

$$\check{\mathbf{F}}_k = \mathbf{A}\hat{\mathbf{F}}_{k-1}\mathbf{A}^T + \mathbf{B}\mathbf{R}_{\text{IMU}}\mathbf{B}^T + \hat{\mathbf{Q}}_{k,\text{mod}} \quad (48)$$

where $\hat{\mathbf{Q}}_{k,\text{mod}}$ is an estimate of the model noise covariance matrix obtained through the iterative innovation-based approach presented at the end of Section IV. In the update step, the only operation different from the radio-based EKF in Section IV is the Kalman gain computation, which becomes

$$\mathbf{G}_k = \check{\mathbf{F}}_k\mathbf{H}^T \left(\mathbf{H}\check{\mathbf{F}}_k\mathbf{H}^T + \hat{\mathbf{R}}'_k - \mathbf{R}_{\text{IMU}} \right)^{-1} \quad (49)$$

where the k th estimation of the measurement covariance matrix $\hat{\mathbf{R}}'_k$ is obtained using the LUT method.

2) *Instance 2 (HCEKF)*: The HNCEKF approach is based on the fact that the state update equation has an overall additive noise, resulting from the sum of: 1) the noisy measurement of the velocity and 2) the model noise, with covariance matrix $\mathbf{Q}_{\text{mod}} + \mathbf{B}\mathbf{R}_{\text{IMU}}$. If we neglect the noise of the constant velocity model, with respect to the IMU measurement noise, only the noise component with covariance matrix $\mathbf{Q}' = \mathbf{B}\mathbf{R}_{\text{IMU}}$ survives. Therefore, the standard EKF can be applied with the following prediction step:

$$\check{\mathbf{s}}_k = \mathbf{A}\hat{\mathbf{s}}_{k-1} + \mathbf{B}\hat{\mathbf{v}}_k \quad (50)$$

$$\check{\mathbf{F}}_k = \mathbf{A}\hat{\mathbf{F}}_{k-1}\mathbf{A}^T + \mathbf{B}\mathbf{Q}'\mathbf{B}^T. \quad (51)$$

The update step remains formally as in Section IV [see (23)–(25)]. The key advantage of this approach, referred to as hybrid control EKF (HCEKF), is the absence of the need to estimate the covariance matrix of the state update equation, i.e., the covariance matrix of the noise of the movement with respect to the considered constant velocity model.

It is worth noting that both the HNCEKF and the HCEKF algorithms are based on the assumption of perfect knowledge of σ_{IMU} . The impact of an imperfect statistical characterization of the IMU noise, i.e., σ_{IMU} , may be subject of future work.

C. Implementation Details

Even if in Section VI, we will assess the system performance by means of numerical simulations (based, however, on experimentally acquired UWB data over realistic paths), we now provide a few details on a potential system implementation of the hybrid schemes previously described. In particular, the following steps may be envisioned.

- 1) The target is equipped with a tag embedding both an IMU, from which the instantaneous velocity vector can be obtained, and a UWB transceiver. The design of a robust algorithm for velocity estimation from IMU-based data is an open and interesting research topic. As expected, the shorter the sampling interval of the IMU, the more accurate the potentially estimated velocity vector.
- 2) The AGV is equipped with N anchors able to receive the UWB signal transmitted by the tag. The anchors are wire connected to a central processing unit responsible for, among others, anchors' synchronization.
- 3) The target sends, at each UWB transmission act, the bidimensional estimated velocity vector in the payload of the UWB packet according to a predefined syntax. Since the microcontroller clock of the tag manages both the IMU and the UWB chips, it can provide (almost) synchronized data.
- 4) The anchors receive the UWB packet and send it to the central processing unit, which extracts the velocity vector, performs TDoA processing, and runs the Kalman-based hybrid tracking algorithm. As anticipated in Section II-C, in this article, all the anchor–target links are assumed to be LOS. In the presence of NLOS links, instead, a proper mitigation strategy should be designed, e.g., by identifying and removing (or correcting) the data received through NLOS links.

VI. NUMERICAL RESULTS

In this section, we provide a performance analysis of the AREKF and the hybrid radio/inertial algorithms described in Section V-A (HIREKF) and Section V-B (HNCEKF and HCEKF), respectively. The performance assessment is obtained through simulations based on experimental measurements. These measurements are acquired with two goals: 1) to validate the UWB noise model in Section II-D and 2) to test the devised algorithms on realistic target trajectories, whose estimates can be visualized. These measurements are associated with a realistic warehouse environment, where various operators co-exist, namely, humans (pedestrians), forklifts, and automated guided vehicles. The target may be placed on any of these operators. Such operators move around the hotspot with the anchors and various shelf racks (with different types of goods) are present.

TABLE I
ANCHORS' POSITIONS (IN METERS)

Anchor	3D coordinates [m]	Anchor	3D coordinates [m]
$\mathbf{a}_0^{(3d)}$	$[0.149, -0.4625, 2.9121]^T$	$\mathbf{a}_6^{(3d)}$	$[1.6449, 0.7988, 0.382]^T$
$\mathbf{a}_1^{(3d)}$	$[-0.1125, -0.809, 0.1077]^T$	$\mathbf{a}_7^{(3d)}$	$[0.929, 0.8029, 1.6708]^T$
$\mathbf{a}_2^{(3d)}$	$[1.6166, -0.8188, 0.5103]^T$	$\mathbf{a}_8^{(3d)}$	$[0.2561, 0.8106, 0.1516]^T$
$\mathbf{a}_3^{(3d)}$	$[0.6756, -0.8127, 1.5578]^T$	$\mathbf{a}_9^{(3d)}$	$[1.4268, -0.0111, 3.9815]^T$
$\mathbf{a}_4^{(3d)}$	$[1.6604, -0.388, 1.7668]^T$	$\mathbf{a}_{10}^{(3d)}$	$[-0.124, 0.4615, 2.5203]^T$
$\mathbf{a}_5^{(3d)}$	$[1.7687, 0.5172, 1.4905]^T$	$\mathbf{a}_{11}^{(3d)}$	$[0.8348, -0.013, 2.5831]^T$

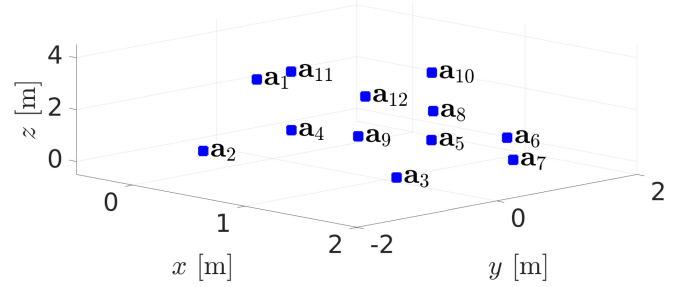


Fig. 6. Anchors' placement for simulations.

All the results are relative to an anchors' placement corresponding to an illustrative setup, composed of $N = 12$ anchors placed on the hotspot at the positions summarized in Table I. The hotspot has an approximate size of $1.5 \text{ m} \times 1.5 \text{ m} \times 4 \text{ m}$ and the chosen placement is compliant with practical anchors' positions on an AGV. A 3-D pictorial description of the anchors' setup is shown in Fig. 6. The CoG of the hotspot is

$$\bar{\mathbf{a}}^{(3d)} = [0.8938, 0.0063, 1.6362]^T. \quad (52)$$

Since the LinHPS is a 3-D positioning algorithm, in all cases, the target is assumed to be at a height equal to 1.5 m. This is representative of a target placed on the upper part of the trunk of a human operator (e.g., in a pocket) moving into the industrial environment. We recall that, however, 2-D tracking is of interest in this article.

The performance assessment presented, here, has been carried out using target paths experimentally acquired in a realistic industrial environment. In particular, the following target paths, representative of a pedestrian human operator moving around the hotspot, have been tested.⁸ The true target positions $\{\mathbf{p}_k\}_{k=0}^{L_{\text{path}}-1}$, where L_{path} is the number of collected positions over a path, have been optically acquired through a Topcon Total Station⁹ with a sampling interval $T_s = 0.5 \text{ s}$. The first experimental target path is shown as the solid line with squares in Fig. 7(a)–(c): the target moves from the bottom left corner (denoted as “start”) to the upper right one (denoted as “end”), making a single turn. The second experimental target path is shown as the solid line with triangles in Fig. 7(d)–(f): the target moves from the left point (denoted as start) to the right one (denoted as end). In all cases, each point along the path corresponds to a true acquired target position (with sampling interval T_s). It is worth noting that these

⁸Similar considerations for other acquired target paths and anchors' placement and are not shown, here, for lack of space.

⁹For more details, see <https://www.topconpositioning.com/total-station-solutions>.

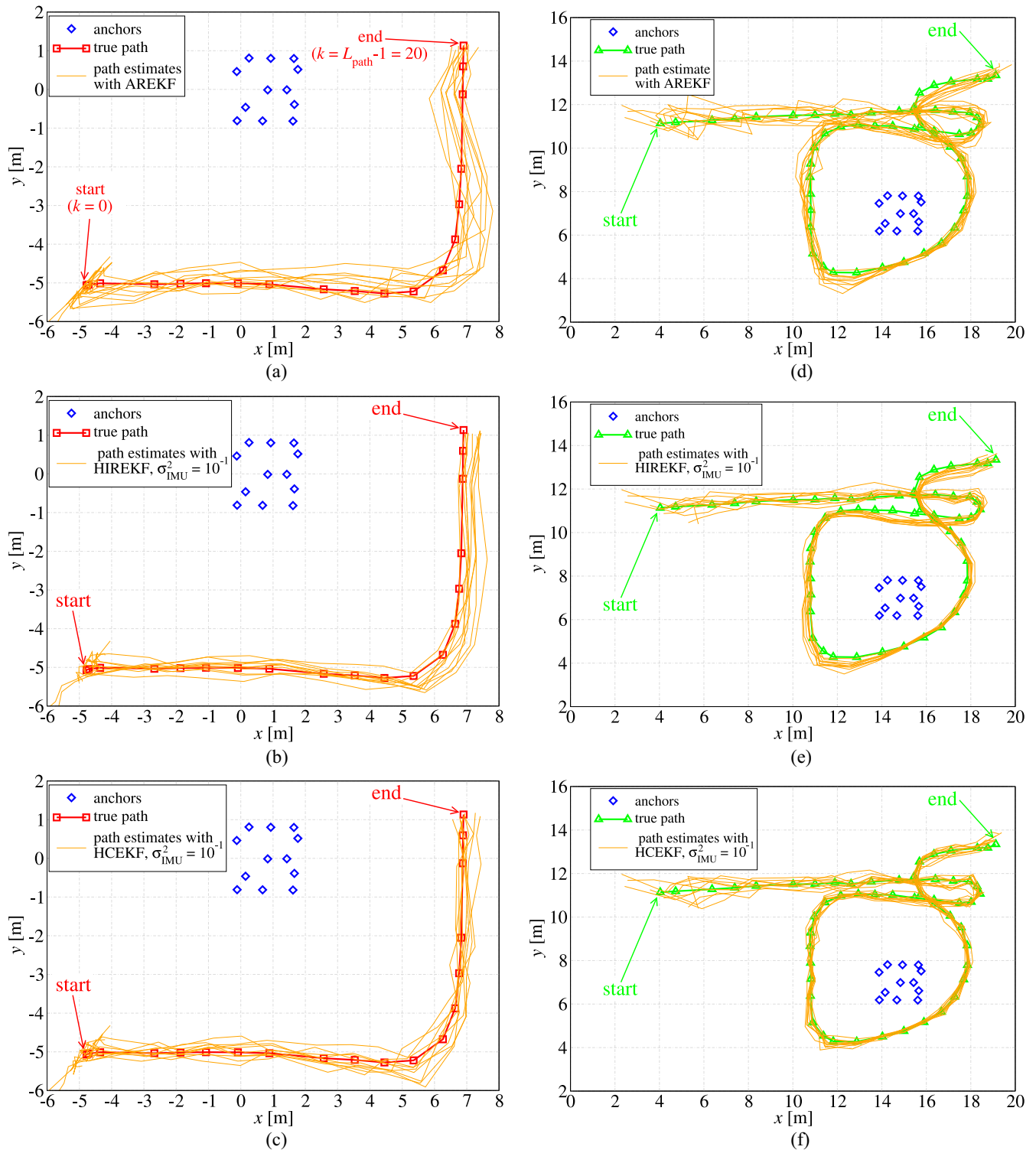


Fig. 7. Path estimates with (a) and (d) AREKF, (b) and (e) HIREKF, and (c) and (f) HCEKF algorithms. Cases (a)–(c) correspond to the first target trajectory, whereas cases (d)–(f) to the second target trajectory. In all hybrid cases, $\sigma_{\text{IMU}}^2 = 10^{-1} \text{ m}^2/\text{s}^2$. The solid lines with squares or triangles correspond to the true experimentally acquired target paths and the start and end points are explicitly indicated.

paths do not fulfill the constant velocity assumption used by the radio-based EKF algorithm. Therefore, one of the goals of our performance analysis is also to evaluate the impact of the constant velocity model, assumed in the derivation of the Kalman-based solutions, on the performance with a realistic movement characterized by a different mobility pattern.

The LUT has a resolution of 1 m in terms of distance (for target–hotspot distances¹⁰ between 2 and 20 m) and 2 degrees in terms of angle (from 0 to 2π). The forgetting factor λ of the AREKF algorithm in (42) is set to 0.3.

¹⁰For distances shorter than 2 m or longer than 20 m, the LUT “saturates” and returns 2 or 20 m, respectively.

We first analyze the effectiveness of the proposed hybrid radio/inertial algorithms, with respect to AREKF, in terms of the “shape” of the estimated target path. In Fig. 7, we show ten path estimates with (a) and (d) AREKF, (b) and (e) HIREKF, and (c) and (f) HCEKF algorithms, respectively.¹¹ Cases (a)–(c) correspond to the first type of target trajectory, whereas cases (d)–(f) to the second type. In all hybrid cases, $\sigma_{\text{IMU}}^2 = 10^{-1} \text{ m}^2/\text{s}^2$. It can be observed that the estimated paths with the hybrid algorithms are more accurate than those obtained with the AREKF, i.e., they are smoother. Moreover, the HCEKF seems to provide the most accurate estimated path, i.e., very close to the actual one, even with a relatively strong IMU noise intensity. Finally, the same conclusions can be drawn also for the second path, i.e., the more complicated one with the target moving around the hotspot (with several turns).

In order to obtain a concise performance evaluation, the following indicators are considered. The first is the *position estimation error* defined, for the k th time epoch of the tracking algorithm, as follows:

$$\varepsilon_k \triangleq |\mathbf{p}_k - \hat{\mathbf{p}}_k| \quad (53)$$

where \mathbf{p}_k and $\hat{\mathbf{p}}_k$ are the true and the estimated target positions, respectively. The performance indicator (53) is small when the instantaneous position estimate is close to the true one. However, even if ε_k is small, the estimated path may continuously “hop” around the true trajectory. In order to quantify the “smoothness” of the path estimate provided by the tracking algorithm, we also consider the *estimated path length*, defined as follows:

$$\hat{L} \triangleq \sum_{k=0}^{L_{\text{path}}-1} |\hat{\mathbf{p}}_k - \hat{\mathbf{p}}_{k-1}|. \quad (54)$$

The rationale behind (54) is that, for a fixed small position estimation error (53), the smoother the estimated path, the smaller (i.e., the closer to the true value) the estimated path length. In particular, this value can be compared with the true path length, which is approximately equal to 17.12 and 45.88 m for the first and second considered target paths, respectively.

In Fig. 8, we carry out a comparative performance analysis between all the considered tracking algorithms (AREKF, HIREKF, HNCEKF, and HCEKF) and the radio-based positioning algorithm, based on LinHPS, proposed in [16] for the first considered target trajectory. We investigate the robustness of these algorithms against the IMU noise—obviously, purely radio-based positioning (LinHPS) and tracking (AREKF) algorithms are not affected by IMU noise. In case (a), the mean estimation error is shown as a function of σ_{IMU}^2 , whereas in case (b), the mean estimated path length is considered. The mean performance is obtained by averaging over 10^5 independent simulation runs.¹² In particular, in each run, different

¹¹The path estimates with the HNCEK algorithm are not shown for lack of space, but they are worse than those with the HCEKF.

¹²Note that in Fig. 7 it seems the estimation error may range from a few centimeters up to 1 m in some cases. However, in this figure, only ten sample paths are considered. By averaging over a large number of independent realizations, most of the instantaneous errors are concentrated to small values.

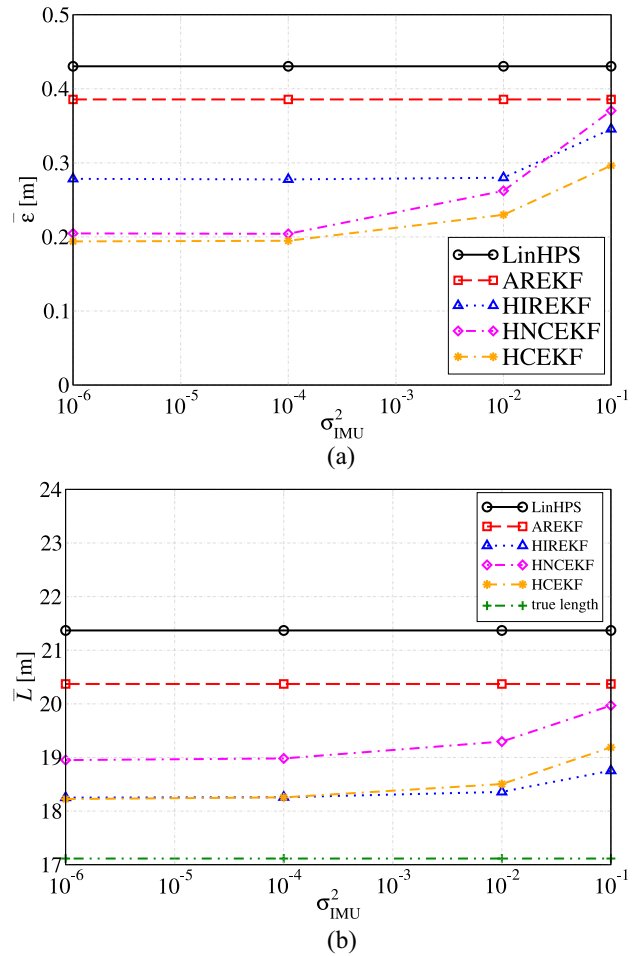


Fig. 8. Mean estimation error [case (a)] and mean estimated path length [case (b)], as functions of σ_{IMU}^2 , for the designed tracking strategies and the first considered target trajectory [shown in Fig. 7(a)–(c)].

noisy TDoAs are computed at the hotspot leading to a different position estimate generated by the LinHPS algorithm. Moreover, a different realization of the noisy velocity measured by the IMU is taken into account. It can be observed that the HCEKF algorithm provides the lowest mean position estimation error [Fig. 8(a)], but the HIREKF algorithm leads to a shorter mean estimated length [Fig. 8(b)]. Since the mean estimated length with HCEKF is slightly longer than that with HIREKF, the HCEKF algorithm can be considered as the best solution. However, since the performance with HIREKF is slightly dependent on the IMU noise, whose statistical characterization is not required by this algorithm, the HIREKF solution is attractive from an implementation point of view. It is also worth noting that HCEKF has a better performance than HNCEKF. This is due to the fact that the estimation of the covariance matrix in the state update equation performed by the HNCEKF is noisy. Since the algorithm is recursive, the estimation error may propagate, thus, degrading the performance. Therefore, it may be more effective to avoid estimating such a matrix, especially, for large IMU noise intensity. Finally, note that, although the hybrid radio/inertial tracking algorithms (HCEKF and HIREKF) show a performance degradation when the IMU noise increases,

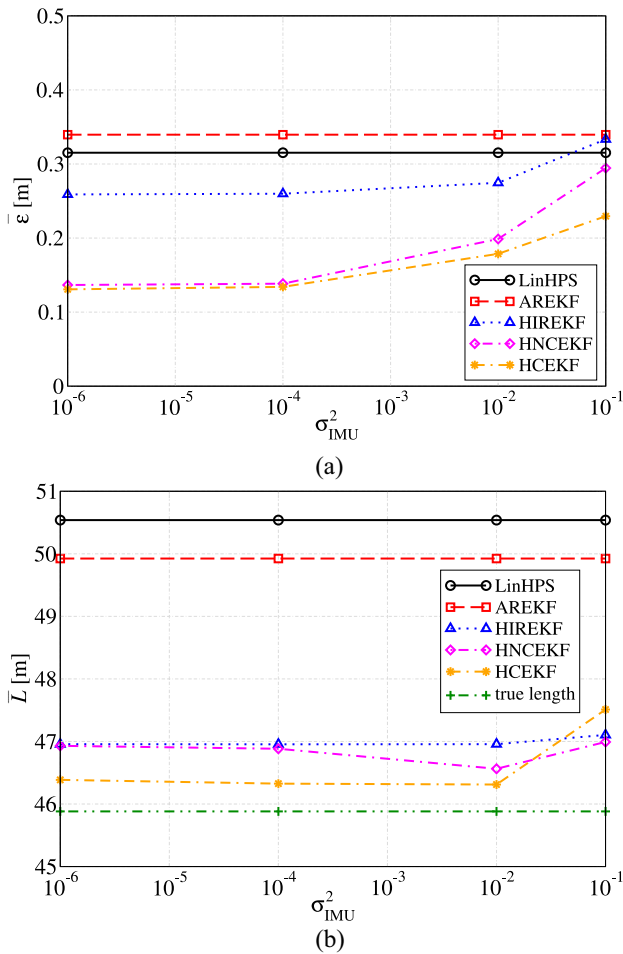


Fig. 9. Mean estimation error [case (a)] and mean estimated path length [case (b)], as functions of σ_{IMU}^2 , for the designed tracking strategies and the second considered target trajectory [shown in Fig. 7(d)–(f)].

this degradation is limited and these algorithms outperform the purely radio-based ones (either positioning, LinHPS, or tracking, AREKF) also for large values of σ_{IMU}^2 .

In Fig. 9, the same analysis carried out in Fig. 8 is performed for the second considered target trajectory. Considerations similar to those presented for the results in Fig. 8 for the first (simpler) path can be carried out in this case as well. However, one can observe that, for large values of the IMU noise intensity σ_{IMU}^2 , the estimation error of the considered tracking strategies may be larger than that with pure radio positioning. This is due to the fact that the considered path is more “irregular” and, therefore, an imprecise (because of the large value of σ_{IMU}^2) side information on the target movement may hinder the accuracy of Kalman-based tracking. Note that, in this case, the average estimation error of AREKF is worse than that of LinHPS. This is due to the irregular shape of the trajectory, because of the presence of several curves. In this case, in fact, the target’s constant velocity assumption is not satisfied during its movement.

We finally propose a performance comparison of the proposed hybrid radio-inertial solutions with respect to reasonable performance benchmarks given by: 1) pure radio positioning and 2) radio-based Kalman tracking—the LinHPS

TABLE II
RELATIVE IMPROVEMENT (PERCENTAGE), IN TERMS OF AVERAGE LOCALIZATION ERROR, OF THE PROPOSED HYBRID RADIO-INERTIAL SOLUTIONS WITH RESPECT TO THE CONSIDERED BENCHMARKS. VARIOUS TARGET TRAJECTORIES AND IMU NOISE INTENSITIES ARE CONSIDERED

trajectory	σ_{IMU}^2	% gain wrt LinHPS			% gain wrt AREKF		
		IREKF	HNCKEF	HCEKF	IREKF	HNCKEF	HCEKF
first Fig. 7 (a)–(c)	10^{-6}	35.30	52.42	54.94	27.80	46.90	49.71
	10^{-4}	35.48	52.55	54.73	28.00	47.05	49.48
	10^{-2}	34.95	39.07	46.58	27.41	32.01	40.39
	10^{-1}	19.68	13.92	31.14	10.30	3.94	23.15
second Fig. 7 (d)–(f)	10^{-6}	17.87	56.65	58.53	23.75	59.76	61.50
	10^{-4}	17.57	56.10	57.46	23.4	59.25	60.51
	10^{-2}	12.88	36.95	43.32	19.12	41.46	47.38
	10^{-1}	-5.74	6.56	27.20	1.83	13.26	32.41

TABLE III
RELATIVE IMPROVEMENT (PERCENTAGE), IN TERMS OF AVERAGE ESTIMATED PATH LENGTH, OF THE PROPOSED HYBRID RADIO-INERTIAL SOLUTIONS, WITH RESPECT TO THE CONSIDERED BENCHMARKS. VARIOUS TARGET TRAJECTORIES AND IMU NOISE INTENSITIES ARE CONSIDERED

trajectory	σ_{IMU}^2	% gain wrt LinHPS			% gain wrt AREKF		
		IREKF	HNCKEF	HCEKF	IREKF	HNCKEF	HCEKF
first Fig. 7 (a)–(c)	10^{-6}	14.59	11.32	14.72	10.41	6.97	10.54
	10^{-4}	14.56	11.17	14.57	10.37	6.82	10.38
	10^{-2}	14.09	9.70	13.41	9.88	5.27	9.16
	10^{-1}	12.24	6.55	10.21	7.94	1.97	5.81
second Fig. 7 (d)–(f)	10^{-6}	7.09	7.14	8.22	5.94	6.00	7.09
	10^{-4}	7.09	7.23	8.34	5.95	6.09	7.21
	10^{-2}	7.09	7.87	8.36	5.94	6.73	7.24
	10^{-1}	6.80	7.02	5.99	5.65	5.87	4.83

and AREKF algorithms, respectively.¹³ In Table II, the relative improvement (percentage), in terms of average localization error, of the proposed hybrid radio-inertial solutions, with respect to the considered benchmarks, is shown. Various target trajectories and IMU noise intensities are considered. It can be observed that, for small IMU noise intensity, the considered hybrid strategies can achieve a gain, with respect to pure positioning (LinHPS), up to approximately 58% (for the second considered trajectory). Such a gain reduces for very large IMU noise intensity, still remaining approximately equal to 31% (for the first considered trajectory). If AREKF is considered as a benchmark, instead, these gains are approximately 61% and 32%, respectively.

In Table III, the relative improvement (percentage), in terms of average estimated path length, of the proposed hybrid radio-inertial solutions, with respect to the considered benchmarks, is shown. Various target trajectories and IMU noise intensities are considered. In this case, the relative gain is much more limited, ranging from approximately 15%–10%.

We now investigate the system performance with different system deployments in terms of number of anchors and their positions. In particular, we focus on the path in Fig. 7(a)–(c). We first compare the performance of the various considered tracking strategies with the Cramer–Rao lower bound (CRLB) of the positioning system, i.e., the TDoA-based LinHPS, which was already derived and analyzed in [16]. In Fig. 10, the CRLB is compared with the root-mean square (RMSE) of the

¹³Note that further comparisons with other state-of-the-art solutions is not viable, due to the specific anchors’ placement (concentrated on the hotspot) considered in this article. This makes the statistical characterization of our radio positioning measurements differ from the models considered in the literature.

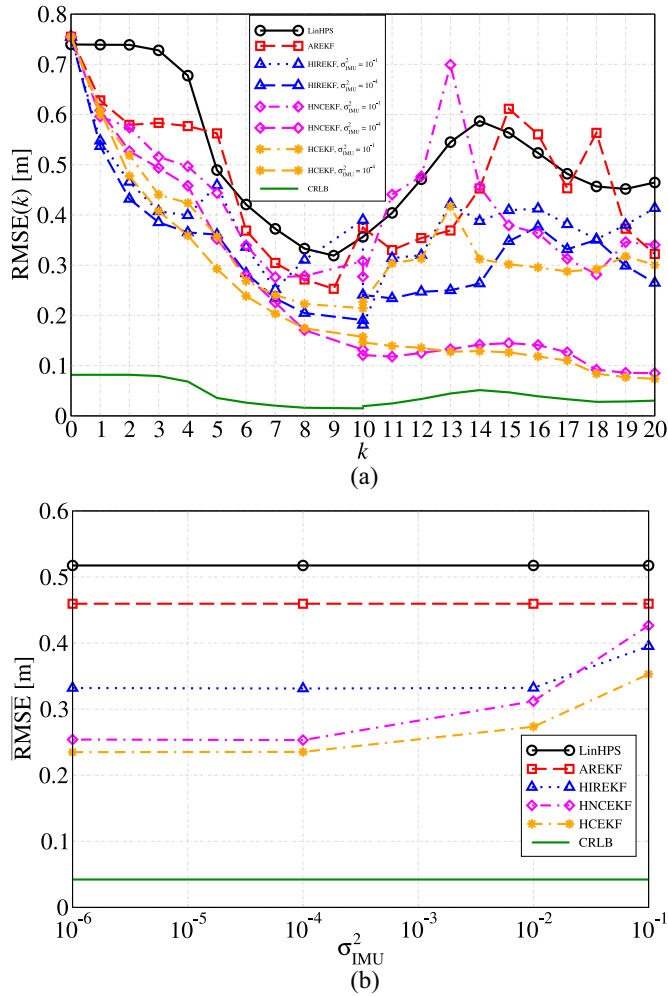


Fig. 10. CRLB and RMSE of the position estimates of the considered localization and tracking strategies: in case (a), the instantaneous RMSE (for each target position) is shown for different IMU noise intensities, whereas in case (b), the mean RMSE (over the entire path) is shown as a function of the IMU noise intensity.

position estimates of the considered localization and tracking strategies. The RMSE, for a given target position \mathbf{p}_k , $k \in \{0, 1, \dots, L_{\text{path}} - 1\}$, is defined as

$$\text{RMSE}(k) \triangleq \sqrt{\frac{1}{T} \sum_{i=1}^T \|\mathbf{p}_k - \hat{\mathbf{p}}_k^{(i)}\|^2} \quad (55)$$

where $\hat{\mathbf{p}}_k^{(i)}$ is the position estimate, of the true position \mathbf{p}_k , at the i th simulation run ($i = 1, 2, \dots, T$). In Fig. 10(a), the RMSE is shown as a function of the target position ($L_{\text{path}} = 21$). Various IMU noise intensities are considered in the hybrid algorithms. It can be observed that, with HNCEKF and HCEKF, the RMSE approaches the CRLB at the end of the path if the IMU noise intensity is sufficiently small (e.g., $\sigma_{IMU}^2 = 10^{-3}$). In order to investigate an average performance, we consider the average RMSE over the entire path

$$\overline{\text{RMSE}} \triangleq \frac{1}{L_{\text{path}}} \sum_{k=0}^{L_{\text{path}}-1} \text{RMSE}(k).$$

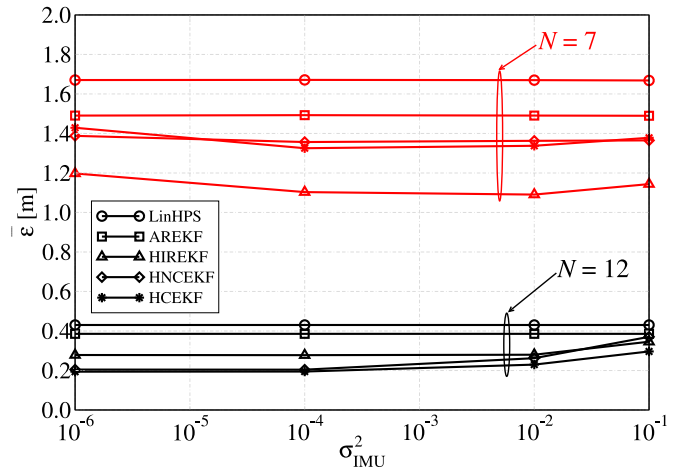


Fig. 11. Mean estimation error, as a function of σ_{IMU}^2 , for the considered tracking strategies and two values of N : 12 (i.e., all the anchors) and 7 (the considered subset).

In Fig. 10(b), the average RMSE is shown as a function of the IMU noise intensity. It can be observed that the considered hybrid radio/inertial mobile target tracking strategies achieve performance sufficiently close to the theoretical one predicted by the CRLB. In particular, the gap of the radio-based tracking with IMU control strategies (i.e., HNCEKF and HCEKF) with respect to the CRLB is half of that achieved by the pure radio tracking (i.e., AREKF).

We then investigate the localization error of the considered tracking strategies with different values of the number of anchors N . To this end, we refer to the subset selection approach originally presented in [16] to improve the positioning accuracy together with data fusion. The results in [16] show that $N = 7$ is the smallest value that can guarantee a sufficient localization accuracy. In particular, we investigate the performance of considered tracking methods with the known best subset (of size $N = 7$) that contains anchors \mathbf{a}_1 , \mathbf{a}_2 , \mathbf{a}_5 , \mathbf{a}_6 , \mathbf{a}_8 , \mathbf{a}_{10} , and \mathbf{a}_{11} (see Fig. 6 and Table I for more specific anchors' positions).

In Fig. 11, the mean estimation error is shown, as a function of σ_{IMU}^2 , for the considered tracking strategies and two values of N : 12 (i.e., all the anchors) and 7 (the considered subset). It is worth noting the performance loss induced by a smaller number of considered anchors that cannot be recovered by the Kalman-based algorithms. However, with $N = 7$ the performance of the hybrid algorithms is slightly influenced by the IMU noise intensity, showing that this class of algorithms is sufficiently robust.

Finally, we investigate the impact of the distance of the hotspot (i.e., the anchors) from the considered path on the estimation error accuracy. In particular, according to Fig. 12(a), we move the anchors' CoG, originally placed at $x = 0.8938$ and $y = 0.0063$, by ± 2 m on both the coordinates. We consider, as a representative concise distance parameter, the distance between the CoG and the point associated with the target position at $k = 13$, denoted as D_{CoG} (dimension: [m]). In Fig. 12(b), the mean estimation error is shown, as a function of D_{CoG} , for the considered target tracking strategies and different values of σ_{IMU}^2 . One can observe that the

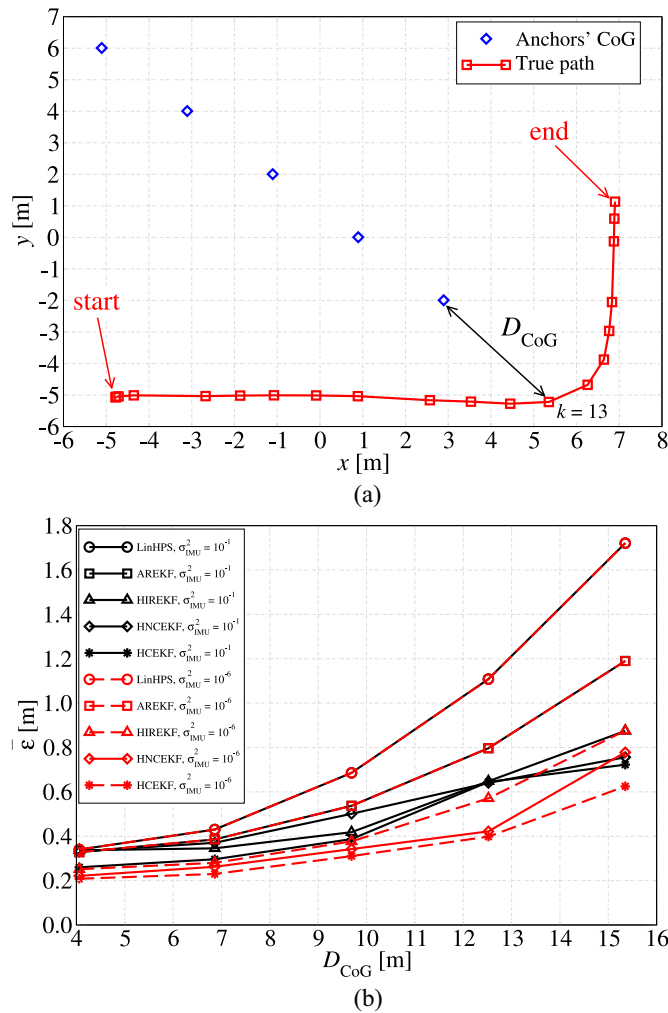


Fig. 12. Impact of the path-hotspot distance: (a) considered scenarios and (b) mean estimation error, as a function of D_{CoG} , for the considered target tracking strategies and different values of σ_{IMU}^2 .

LinHPS positioning scheme and the pure radio tracking strategy (AREKF) are less robust; the longer D_{CoG} , the higher the mean estimation error. On the other hand, the hybrid radio/inertial strategies (HIREKF, HNCEKF, and HCEKF) are more robust and the performance degradation is more limited. In particular, the lower the IMU noise intensity, the smaller the mean estimation error.

VII. CONCLUSION

In this article, we have addressed the problem of target tracking, when the anchors are placed on a single hotspot and the target moves around the hotspot itself. EKF-based solutions have been proposed, in which measurements come from a radio-based positioning algorithm (namely, the LinHPS, exploiting UWB communications, and TDoA processing at the anchors). First, a purely radio EKF, denoted as AREKF, has been proposed, with adaptive estimation of the distribution of the noise in the movement model—the covariance of the measurement noise is estimated according to an LUT offline precomputed during a calibration phase. Then, hybrid radio/inertial algorithms (HIREKF, HNCEKF, and HCEKF) have been proposed to improve the performance of AREKF

by taking into account inertial data (more precisely, the instantaneous velocity) collected from the target. Our results show that the proposed hybrid algorithms can track the true target path with an average estimation error, with respect to the true position, on the order of a few centimeters. Overall, the HCEKF scheme is the best solution among those proposed in this article.

REFERENCES

- [1] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi, "Collaborative sensor network localization: Algorithms and practical issues," *Proc. IEEE*, vol. 106, no. 6, pp. 1089–1114, Jun. 2018, doi: [10.1109/JPROC.2018.2829439](https://doi.org/10.1109/JPROC.2018.2829439).
- [2] F. Zafari, A. Gkelias, and K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019, doi: [10.1109/COMST.2019.2911558](https://doi.org/10.1109/COMST.2019.2911558).
- [3] A. Rabbachin, T. Q. S. Quek, P. C. Pinto, I. Oppermann, and M. Z. Win, "Non-coherent UWB communication in the presence of multiple Narrowband interferers," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3365–3379, Nov. 2010, doi: [10.1109/TWC.2010.091510.080203](https://doi.org/10.1109/TWC.2010.091510.080203).
- [4] B. Sobhani, E. Paolini, A. Giorgetti, M. Mazzotti, and M. Chiani, "Target tracking for UWB multistatic radar sensor networks," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 1, pp. 125–136, Feb. 2014, doi: [10.1109/JSTSP.2013.2286771](https://doi.org/10.1109/JSTSP.2013.2286771).
- [5] S. W. Chen, C. K. Seow, and S. Y. Tan, "Elliptical lagrange-based NLOS tracking Localization scheme," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3212–3225, May 2016, doi: [10.1109/TWC.2016.2518174](https://doi.org/10.1109/TWC.2016.2518174).
- [6] J. Kulmer, E. Leitinger, S. Grebien, and K. Witrisal, "Anchorless cooperative tracking using multipath channel information," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2262–2275, Apr. 2018, doi: [10.1109/TWC.2018.2790936](https://doi.org/10.1109/TWC.2018.2790936).
- [7] H. J. Kim, Y. Xie, H. Yang, C. Lee, and T. L. Song, "An efficient indoor target tracking algorithm using TDOA measurements with applications to ultra-Wideband systems," *IEEE Access*, vol. 7, pp. 91435–91445, 2019, doi: [10.1109/ACCESS.2019.2927005](https://doi.org/10.1109/ACCESS.2019.2927005).
- [8] D. Feng, C. Wang, C. He, Y. Zhuang, and X. Xia, "Kalman-filter-based integration of IMU and UWB for high-accuracy indoor positioning and navigation," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3133–3146, Apr. 2020, doi: [10.1109/JIOT.2020.2965115](https://doi.org/10.1109/JIOT.2020.2965115).
- [9] L. Yao, Y.-W. A. Wu, L. Yao, and Z. Z. Liao, "An integrated IMU and UWB sensor based indoor positioning system," in *Proc. Int. Conf. Indoor Position. Indoor Navigat. (IPIN)*, Sapporo, Japan, Sep. 2017, pp. 1–8, doi: [10.1109/IPIN.2017.8115911](https://doi.org/10.1109/IPIN.2017.8115911).
- [10] "MPU-9250 nine-axis MEMS MotionTracking device." Accessed: Mar. 13, 2023. [Online]. Available: <https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/>
- [11] "decaWave DW1000." Accessed: Mar. 13, 2023. [Online]. Available: <https://www.qorvo.com/products/p/DWM1000>
- [12] X. Li and Y. Wang, "Research on the UWB/IMU fusion positioning of mobile vehicle based on motion constraints," *Acta Geodaetica et Geophysica*, vol. 55, pp. 237–255, Jun. 2020, doi: [10.1007/s40328-020-00291-8](https://doi.org/10.1007/s40328-020-00291-8).
- [13] A. Benini, A. Mancini, A. Marinelli, and S. Longhi, "A biased extended Kalman filter for indoor Localization of a mobile agent using low-cost IMU and UWB wireless sensor network," in *IFAC Proc. Vol.*, vol. 45, no. 22, pp. 735–740, 2012, doi: [10.3182/20120905-3-HR-2030.00144](https://doi.org/10.3182/20120905-3-HR-2030.00144).
- [14] "Ubisense dimension4." Accessed: Mar. 13, 2023. [Online]. Available: <https://ubisense.com/sensor-systems/>
- [15] "MPU-6050 six-axis MEMS MotionTracking devices." Accessed: Mar. 13, 2023. [Online]. Available: <https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>
- [16] M. Martalò, S. Perri, G. Verdano, F. De Mola, F. Monica, and G. Ferrari, "Improved UWB TDoA-based positioning using a single Hotspot for industrial IoT applications," *IEEE Trans. Ind. Inform.*, vol. 18, no. 6, pp. 3915–3925, Jun. 2022, doi: [10.1109/TII.2021.3111449](https://doi.org/10.1109/TII.2021.3111449).
- [17] M. Martalò, G. Ferrari, S. Perri, G. Verdano, F. De Mola, and F. Monica, "UWB TDoA-based positioning using a single hotspot with multiple anchors," in *Proc. IEEE Int. Conf. Comput. Commun. Secur. (ICCCS)*, Rome, Italy, Oct. 2019, pp. 1–7.
- [18] J. A. Corrales, F. A. Candelas, and F. Torres, "Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interact. (HRI)*, Amsterdam, The Netherlands, Mar. 2008, pp. 193–200, doi: [10.1145/1349822.1349848](https://doi.org/10.1145/1349822.1349848).

- [19] M. Kolakowski, "Utilizing acceleration measurements to improve TDOA based localization," in *Proc. Signal Process. Symp. (SPS)*, Debe, Poland, Sep. 2017, pp. 1–4, doi: [10.1109/SPS.2017.8053694](https://doi.org/10.1109/SPS.2017.8053694).
- [20] R. Zandian and U. Witkowski, "Implementation challenges of synchronisation of UWB nodes in TDoA structures," in *Proc. Int. Conf. Indoor Position. Indoor Navigat. (IPIN)*, Nantes, France, Sep. 2018, pp. 1–8, doi: [10.1109/IPIN.2018.8533796](https://doi.org/10.1109/IPIN.2018.8533796).
- [21] S. Leugner, M. Pelka, and H. Hellbrück, "Comparison of wired and wireless synchronization with clock drift compensation suited for U-TDOA localization," in *Proc. Workshop Position. Navigat. Commun. (WPNC)*, Bremen, Germany, 2016, pp. 1–4, doi: [10.1109/WPNC.2016.7822846](https://doi.org/10.1109/WPNC.2016.7822846).
- [22] S. Bottigliero, D. Milanesio, M. Saccani, and R. Maggiore, "A low-cost indoor real-time locating system based on TDOA estimation of UWB pulse sequences," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021, doi: [10.1109/TIM.2021.3069486](https://doi.org/10.1109/TIM.2021.3069486).
- [23] M. Brossard, A. Barrau, and S. Bonnabel, "AI-IMU dead-reckoning," *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 585–595, Dec. 2020, doi: [10.1109/TIV.2020.2980758](https://doi.org/10.1109/TIV.2020.2980758).
- [24] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Found. Trends Signal Proc.*, vol. 11, nos. 1–2, pp. 1–153, Apr. 2017, doi: [10.1561/20000000094](https://doi.org/10.1561/20000000094).
- [25] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*. Hoboken, NJ, USA: Wiley, 2015, doi: [10.1002/9781118984987](https://doi.org/10.1002/9781118984987).
- [26] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 2013, doi: [10.1017/9781139020411](https://doi.org/10.1017/9781139020411).
- [27] R. Bucher and D. Misra, "A synthesizable VHDL model of the exact solution for three-dimensional hyperbolic positioning system," *VLSI Design*, vol. 15, no. 2, pp. 507–520, 2002, doi: [10.1080/1065514021000012129](https://doi.org/10.1080/1065514021000012129).
- [28] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York, NY, USA: McGraw-Hill, 1991.
- [29] S. Akhlaghi, N. Zhou, and Z. Huang, "Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation," in *Proc. IEEE Power Energy Soc. General Meeting*, Chicago, IL, USA, Jul. 2017, pp. 1–5, doi: [10.1109/PESGM.2017.8273755](https://doi.org/10.1109/PESGM.2017.8273755).
- [30] W. Ma, J. Qiu, J. Liang, and B. Chen, "Linear Kalman filtering algorithm with noisy control input variable," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 7, pp. 1282–1286, Jul. 2019, doi: [10.1109/TCSII.2018.2878951](https://doi.org/10.1109/TCSII.2018.2878951).



Gianmichele Verdano received the M.Sc. degree in computer engineering from the University of Parma, Parma, Italy, in 2010.

He worked as a Research and Development Engineer for a motorsport company from 2011 to 2016. Since February 2016, he has been working with Elettric80 as a Research and Development, Viano, Italy, and since 2020 as a Research and Development Project Leader.



Francesco De Mola received the M.Sc. degree in computer engineering and the Ph.D. degree in ICT—computer engineering and science from the University of Modena and Reggio Emilia, Modena, Italy, in 2005 and 2008, respectively.

Since 2008, he has been working with E80 Research and Development Team, Viano, Italy, and since 2014, he holds the role of a Research and Development Team Leader.



Francesco Monica received the M.Sc. degree in computer engineering and the Ph.D. degree in ICT from the University of Parma, Parma, Italy, in 2003 and 2006, respectively.

He worked as a Research and Development Team Leader for a provider of CCTV systems from 2006 to 2015, leading the Computer Vision Research and Development Group of the company. Since July 2015, he has been working with Elettric80, Viano, Italy, as a Research and Development Project Manager.



Marco Martalò (Senior Member, IEEE) received the Ph.D. degree in information technologies from the University of Parma, Parma, Italy, in 2009.

From 2012 to 2017, he was an Assistant Professor with E-Campus University, Novedrate, Italy, and also a Research Associate with the University of Parma until 2020. Since 2020, he has been an Associate Professor of Telecommunications with the University of Cagliari, Cagliari, Italy, where he is part of the Networks for Humans (Net4U) Laboratory. He has coauthored the book *Sensor*

Networks with IEEE 802.15.4 Systems: Distributed Processing, MAC, and Connectivity. His research interests are in the design of communication and signal processing algorithms for wireless systems and networks, as well as security aspects to them.



Simone Perri received the degree in mathematics from the University of Parma, Parma, Italy, in 2014.

He worked from 2014 to 2016 as a Research Fellow with the University of Parma and from 2016 to 2022 with the E80 Research and Development Team, Viano, Italy. Since August 2022, he has been working with Able Tech Srl, Lonato del Garda, Italy.



Gianluigi Ferrari (Senior Member, IEEE) received the Ph.D. degree in information technologies from the University of Parma, Parma, Italy, in 2002.

He is a Full Professor of Telecommunications with the University of Parma. He is currently the Head of the Bachelor Degree in Computer Engineering, Electronics and Telecommunications with the University of Parma (<https://cdliet.unipr.it/>). Since September 2006, he has been the Coordinator of the Internet of Things (IoT) Laboratory (<http://iotlab.unipr.it/>) and, since 2016, he has been the Co-Founder and the President of things2i s.r.l. (<http://www.things2i.com/>), a spin-off company of the University of Parma dedicated to IoT and smart systems. He has published and consulted extensively in these areas, coordinating several technical projects, including EU-funded competitive projects. His research activities revolve around signal processing, communication/networking, and IoT.

Prof. Ferrari is a member of the Scientific Council of the INSIDE Industry Association (<https://www.inside-association.eu/>).