Indoor Localization of Mobile Robots Through QR Code Detection and Dead Reckoning Data Fusion

Payam Nazemzadeh , *Member, IEEE*, Daniele Fontanelli , *Member, IEEE*, David Macii , *Senior Member, IEEE*, and Luigi Palopoli, *Member, IEEE*

Abstract—**Many techniques for robot localization rely on the assumption that both process and measurement noises are uncorrelated, white, and normally distributed. However, if this assumption does not hold, these techniques are no longer optimal and, in addition, the maximum estimation errors can be hardly kept under control. In this paper, this problem is addressed by means of a tailored extended H_∞ filter (EHF) fusing odometry and gyroscope data with position and heading measurements based on quick response (QR) code landmark recognition. In particular, it is shown that, by properly tuning EHF parameters and by using an adaptive mechanism to avoid finite escape time phenomena, it is possible to achieve a higher localization accuracy than using other dynamic estimators, even if QR codes are detected sporadically. Also, the proposed approach ensures a good tradeoff in terms of computational burden, convergence time, and deployment complexity.**

Index Terms—**Estimation, H_∞ filters, localization, position measurements, robotics, sensor fusion.**

I. INTRODUCTION

POSITION estimation and tracking are essential for robots navigation as well as to manage formations of multiple robots [1]. Over the last few years, a variety of sensing solutions has been proposed to determine the absolute position of targets moving indoors. At the moment, none of the existing solutions outperforms the others in all aspects. Most of the techniques based on radio frequency [2], optical [3], or ultrasonic signals [4] require to place fixed reference nodes at known coordinates in a given environment. Moreover, such solutions generally suffer

from scalability issues. Possible alternative techniques solving this problem at its source are those based on light detection and ranging (LiDAR) systems or cameras able to recognize specific landmarks in the environment [5], [6]. However, even if LiDAR systems can assure high ranging accuracy (e.g., in the order of a few cm for distances up to tens of meters) [7], generally they are also quite expensive. Moreover, performance may drop due to interference problems when many devices are used in the same environment or in the presence of mirror reflections [8].

The vision-based egomotion techniques are generally quite heavy from the computational point of view and also suffer from robustness problems due to their sensitivity to changeable light conditions and to the type of landmarks. Nonetheless, such techniques are commonly used for simultaneous localization and mapping [9]. If the landmarks are easy to detect and if their density in the environment is reasonably low, the data fusion of vision-based measurements and dead reckoning could be a viable solution to achieve accurate, scalable, and trustworthy localization [10]–[12]. This is indeed also the basic idea underlying the technique described in this paper. In particular, the proposed approach relies on an extended H_∞ filter (EHF) fusing odometry and gyroscope data with sporadic absolute position and heading values estimated in one shot by measuring distance and angular offsets from quick response (QR) codes stuck regularly on the floor.

Most of Bayesian position estimators of moving targets rely on Kalman filters (KF), extended KF (EKF), unscented KF (UKF), or particle filters (PF). For instance, they can be used either to merge heterogeneous odometer and inertial measurement data [13], or to mitigate the uncertainty growth due to inertial measurement units (IMU) through additional contextual information [14]. As known, KFs and EKFs are optimal when the underlying models are linear or weakly nonlinear, respectively, and they are affected by normally distributed, uncorrelated, and white noise contributions. In the case of strongly nonlinear systems, the UKFs generally provide better results than EKFs, as they usually rely directly on nonlinear models. When instead the process and/or the measurement noise are far from being normally distributed or white, an effective alternative is the PF [15]. Unfortunately, the inherent heuristic nature of this class of algorithms does not allow us to keep the maximum estimation error under control. In addition, PF computational complexity

usually grows with the number of particles. Therefore, a trade-off between convergence time and processing burden has to be found. A possible solution to this problem is offered by the EHF. Similarly to PFs, EHFs do not rely on the upfront knowledge of noise distributions and do not require noises to be white. However, they are purposely designed to minimize the worst-case estimation error with maximum likelihood [16]. For this reason, the use of H_∞ filters for robot localization has become increasingly interesting over the last few years, e.g., in [17] and [18]. However, the system and measurement models used in [17] and [18] are different from those adopted in this paper. Particularly, none of them rely on intermittent and sporadic vision-based measurements. Therefore, no adaptive mechanisms are needed to avoid finite escape time phenomena like in the case at hand.

In the rest of this paper, after a short review of various localization techniques based on vision systems and data fusion (see Section II), first the system and measurement models are described in Sections III and IV, respectively. Then, in Section V, the proposed EHF algorithm is presented. Finally, Sections VI and VII describe the experimental setup and the corresponding results.

II. RELATED WORK

In general, vision-based localization relies on one or more cameras as well as on image processing algorithms recognizing suitable visual landmarks. Such landmarks can provide a variety of information, i.e., images that can be matched with those available in a database of the considered environment [19], features that can be used to estimate the egomotion of the robot [12], [20], or position and orientation data directly [10], [21]. The first approach is completely different from the general idea of this paper, as it requires both a preliminary exploration of the environment and large computational resources for image processing and matching.

The solutions based on image feature extraction typically rely on sequences of images to determine the motion (e.g., the velocity) of a robot. For instance, in [20] an egomotion method fusing optical flow and inertial measurement data through an UKF is proposed. In [12], a monocular camera with the optical axis approximately parallel to the gravitational field is mounted on an IMU. In this case, the landmarks are the ground feature points detected in consecutive images collected by the camera. Again, data fusion is based on an UKF. The main benefit of this solution is that it does not rely on optical flow and it is independent of the underlying vision algorithm. In all the solutions of this kind, the velocity measurement uncertainty is kept bounded. However, the position estimation error usually still tends to grow unboundedly, although at a much slower rate than using the inertial data alone.

The custom indoor adaptive localization algorithm described in [22] tackles these problems, as it exhibits good convergence and a stable steady-state error. It relies on the fusion of vision-based measurements, odometry and inertial data. However, the computational burden is remarkable, as the algorithm requires a graphics processing unit and a parallel implementation to provide real-time performance.

The main advantage of the techniques estimating position and heading directly from the landmarks is the reduced computational effort. In general, serious robustness problems may arise if the information encoded in the landmarks is not easy to detect. In [10] and [11], ad hoc two-dimensional landmarks are placed on the ceiling to mitigate the effect of possible obstacles. However, whereas in [11], localization is based on image processing only, the solution in [10] relies on an EKF that combines the information extracted from the landmarks with the data measured by two encoders and one gyroscope. This approach is similar to the technique described in this paper, but with various differences. First, the system model is different. Second, the EKF relies on the assumption that both the process noise and the measurement uncertainty are normally distributed and white. This is sometimes merely a leap of faith, as it will be shown in Section VI-B. Finally, the landmarks used in [10] are custom and placed on the ceiling (which is not always possible, e.g., in places with high ceilings such as airports or shopping malls), whereas in this paper, standard QR codes are stuck on the floor and are detected by a low-cost camera. The idea of using QR codes as visual landmarks is not completely new [23]–[25], since they can be easily and robustly detected in the environment using standard software tools. For instance, the solution described in [25] relies on a fine-grained grid of QR codes placed on the ceiling. Whereas the method to extract absolute position and orientation from QR codes is similar to the one described in this paper, the overall approach is completely different. In [25], robot localization relies solely on QR code detection. Therefore, at least one landmark must be in the field of view of a high-frame-rate camera at any time. On the contrary, in this paper, robot positioning is based on the fusion of sporadic QR code readings and encoder or gyroscope data. In this way, QR code grid granularity can be traded for positioning accuracy, as it will be shown in Section VII-B.

The technique described in this paper also provides a significant advancement with respect to the localization technique described in [26]. In [26], the planar position of the wheeled device was measured by reading one of the passive radio-frequency identification (RFID) tags deployed in the environment, with the vision system being used just to estimate the heading of the device with respect to the direction pointed by arrow-shaped landmarks. In this paper instead, absolute position and heading are extracted directly from QR codes, thus simplifying the experimental setup and making image processing simpler and faster. Moreover, the proposed estimation algorithm is optimal in a minimax sense, even under the effect of unknown measurement uncertainty distributions and unmodeled dynamic phenomena.

III. SYSTEM MODEL

A qualitative overview of the localization problem considered in this paper is shown in Fig. 1 and it can be potentially applied to any unicycle-like wheeled robot. Two encoders, a gyroscopic platform, and a front monocular camera are used to estimate the position of the robot within a given reference frame $\langle W \rangle = \{X_w, Y_w, Z_w\}$. In the following, the reference point chosen to define the robot's position in $\langle W \rangle$ is the midpoint of
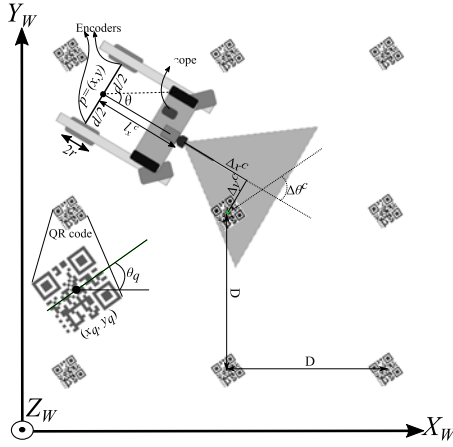
Fig. 1. Overview of mobile robot localization within the fixed reference frame $\langle W \rangle = \{X_w, Y_w, Z_w\}$.

the axle between the rear wheels, where the encoders used for odometry are actually installed. In order to keep into account the systematic errors affecting the odometry-based linear and angular velocity values, the state vector of the system describing the robot's kinematic is defined as $\mathbf{s} = [x, y, \theta, \mu, \delta]^T$ where (x, y) are the planar coordinates of the reference point, θ is the heading of the robot with respect to X_w, and μ and δ are the relative systematic offsets affecting the linear and angular velocities of the robot in the reference point. As a result, if r denotes the wheels radius and d is the length of the wheels axle, the augmented unicycle-like model describing the kinematic of the robot with odometry is

$$\dot{\mathbf{s}} = \mathbf{f}(\mathbf{s})\mathbf{\Omega} \tag{1}$$
$$\mathbf{o} = \mathbf{h}(\mathbf{s})$$

where

$$\mathbf{f}(\mathbf{s}) = \begin{bmatrix} \frac{r}{2}(1+\mu)\cos\theta & \frac{r}{2}(1+\mu)\cos\theta \\ \frac{r}{2}(1+\mu)\sin\theta & \frac{r}{2}(1+\mu)\sin\theta \\ \frac{r}{d}(1+\delta) & -\frac{r}{d}(1+\delta) \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2}$$

where $\mathbf{\Omega} = [\omega_r, \omega_l]^T$ is the input vector including the angular velocities ω_r and ω_l of the right and left wheel, respectively, and finally $\mathbf{h}(\cdot)$ denotes a generic nonlinear output function of the state (see Section IV for details). Observe that in (2), μ and δ are assumed to be constant. To a first approximation, this is a sound assumption, as the systematic deviations affecting the angular displacements measured by the wheels encoders tend to increase linearly with velocity, as it is shown in Section VI-B, and particularly in Table II. It is worth noticing that including μ and δ in the state vector is important to estimate (and to partially compensate for) the drift of odometry-based measurements when dynamic state estimators are used.

If (1) is discretized using the Euler method with period T_s (which guarantees that the induced modeling errors are smaller than the encoders uncertainties in the setup described in

Section VI) and if the process noise (e.g., due to both encoders and robot vibrations) is included in the model, then (1) can be rewritten as

$$\mathbf{s}_{k+1} = \mathbf{s}_k + \mathbf{f}(\mathbf{s}_k)\widehat{\mathbf{\Delta\Phi}}_k \tag{3}$$
$$\mathbf{o}_k = \mathbf{h}(\mathbf{s}_k)$$

where $\mathbf{s}_k$ represents the state vector at time kT_s and $\widehat{\mathbf{\Delta\Phi}}_k = \mathbf{\Delta\Phi}_k + \boldsymbol{\varepsilon}_{\Phi,k}$ is the vector of the noisy input quantities. In particular, the vector $\mathbf{\Delta\Phi}_k = \mathbf{\Omega}_k T_s = [\Delta\Phi_{r_k}, \Delta\Phi_{l_k}]^T$ includes the actual angular displacements of the left and right wheels, respectively, between time kT_s and $(k+1)T_s$, whereas $\boldsymbol{\varepsilon}_{\Phi,k}$ is the corresponding random noise vector.

IV. MEASUREMENT TECHNIQUES AND MODELS

In this paper, two measurement systems are used to mitigate the accumulation of odometry uncertainty, i.e., a vision system consisting of a monocular front camera and a gyroscope-based platform.

The role of the vision system is to detect one of the QR codes stuck on the floor. Without loss of generality, in the following it will be assumed that the QR codes are placed on a regular square-patterned lattice, with the side of each square equal to D, as shown in Fig. 1. Each QR code represents a number which is univocally associated with a triple of values, i.e., the planar coordinates (x_q, y_q) of the point in the center of the landmark, and its orientation angle θ_q with respect to X_w. Consider that only when one of the QR codes is in the field of view of the camera, the encoded information is actually available. Therefore, such measurement data are inherently intermittent and sporadic. Further details on QR code implementation and placement are reported in Section VI-A. With reference to Fig. 1, let Δx^c and Δy^c be the distances between the camera and the detected QR code in the camera frame. Also, let $\Delta\theta^c$ be the angle difference between the camera optical axis and θ_q. The values of such quantities can be measured using standard image processing algorithms, e.g., based on homography [27]. The details of such measurement techniques are out of the scope of this paper. However, it is important to emphasize that if the points of the landmark to be detected are coplanar and if the QR code dimensions are known *a priori*, the homography-based techniques ensure a robust estimation of Δx^c, Δy^c, and $\Delta\theta^c$, regardless of the actual position and orientation of the camera. In the following, the measured quantities will be denoted with "hatted" symbols, i.e., $\widehat{\Delta x}^c = \Delta x^c + \zeta_x^c$, $\widehat{\Delta y}^c = \Delta y^c + \zeta_y^c$, and $\widehat{\Delta\theta}^c = \Delta\theta^c + \zeta_\theta^c$, where ζ_x^c, ζ_y^c, and ζ_θ^c represent the measurement uncertainty terms associated with the respective quantities. An experimental analysis of the distributions of ζ_x^c, ζ_y^c, and ζ_θ^c in the case at hand is reported in Section VI-B.

Given that ensuring high-rate and good heading estimates can greatly improve localization accuracy in the long term [28], an additional gyroscope-based platform is used. To keep the total heading measurement uncertainty bounded, the vision-based angle measurements can be used to adjust the yaw angular

displacements estimated from the integration of gyroscope data (see Section IV-B).

A. Position Measurements

As explained above, the vision system is used to measure the distances Δx^c and Δy^c between the camera and the point (x_q, y_q) associated with the center of a detected QR code. However, if the camera is not co-located in the reference point used to define the position of the robot (as it typically occurs in practice), the coordinates of the reference point are functions of Δx^c and Δy^c through a constant rigid transformation. In particular, if l_x^c and l_y^c denote the constant coordinates of the reference point in the camera frame, then

$$\mathbf{o}_k^c = \mathbf{h}^c(\mathbf{s}_k) = \begin{bmatrix} (x_q - x_k)\cos\theta_k + (y_q - y_k)\sin\theta_k + l_x^c \\ -(x_q - x_k)\sin\theta_k + (y_q - y_k)\cos\theta_k + l_y^c \end{bmatrix}. \tag{4}$$

Moreover, if the elements of $\mathbf{o}_k^c$ are measured, then the corresponding measurement model becomes

$$\hat{\mathbf{o}}_k^c = \begin{bmatrix} \widehat{\Delta x_k^c} \\ \widehat{\Delta y_k^c} \end{bmatrix} = \mathbf{h}^c(\mathbf{s}_k) + \boldsymbol{\zeta}_k^c \tag{5}$$

where $\boldsymbol{\zeta}_k^c = [\zeta_{x,k}^c, \zeta_{y,k}^c]^T$ is the vector of measurement uncertainty contributions at time kT_s.

B. Heading Measurements

Let $\mathbf{q} = [\theta^g, b^g]^T$ be a vector composed by the yaw angle θ^g and the relative angular velocity error b^g associated with the gyroscope-based platform. If b^g is assumed to be approximately constant and is not properly estimated and compensated, then the angular velocity observed by the gyroscope is affected by a systematic deviation proportional to ω. The results in Section VI-B will show that this assumption is supported by experimental evidence. If the sampling period of the gyroscope is T_s, then the discretized heading measurement model is

$$\mathbf{q}_{k+1} = \begin{bmatrix} \theta_{k+1}^g \\ b_{k+1}^g \end{bmatrix} = \begin{bmatrix} \theta_k^g + T_s(1 + b_k^g)\omega_k \\ b_k^g \end{bmatrix} + \varepsilon_k^g$$

$$o_k^\theta = h^\theta(\theta_k^g) \tag{6}$$

where ε^g denotes the noise introduced by the gyroscope-based platform and $h^\theta(\cdot)$ is the output function. In practice, this depends on the yaw angle only, since b^g usually cannot be observed. If the vision system is used to measure $\Delta\theta^c$, then it can be easily shown that $o_k^\theta = h^\theta(\theta_k^g) = \Delta\theta_k^c = \theta_q - \theta_k^g$, as shown in Fig. 1. Therefore, the corresponding measurement equation becomes $\hat{o}_k^\theta = \widehat{\Delta\theta_k^c} = o_k^\theta + \zeta_k^c$, ζ_k^c being the angle measurement error due to the vision system at time kT_s. If the elements of ε^g and ζ^c are uncorrelated and normally distributed with zero mean, the values of $\mathbf{q}$ can be optimally estimated by an EKF based on (6). As known, an EKF relies on two iterative steps (namely, prediction and update) applied to the linearized dynamic [29]. In this case, the equations of the prediction step

are

$$\hat{\mathbf{q}}_{k+1}^+ = \hat{\mathbf{q}}_k + \mathbf{g}(\hat{\mathbf{q}}_k)\hat{\omega}_k$$

$$Q_{k+1}^+ = A_k Q_k A_k^T + B_k \sigma_{\omega,k}^2 B_k^T \tag{7}$$

where $\hat{\mathbf{q}}_k$ and $\hat{\mathbf{q}}_{k+1}^+$ denote the estimated and predicted state, respectively, $\mathbf{g}(\hat{\mathbf{q}}_k) = \begin{bmatrix} T_s(1+\hat{b}_k^g) \\ 0 \end{bmatrix}$ is the input function, $\hat{\omega}_k$ is the angular velocity value given by the gyroscope-based platform at time kT_s, $A_k = \begin{bmatrix} 1 & T_s\hat{\omega}_k \\ 0 & 1 \end{bmatrix}$ and $B_k = \begin{bmatrix} T_s(1+\hat{b}_k^g) \\ 0 \end{bmatrix}$ are the Jacobians of the process model (6) with respect to $\mathbf{q}$ and ω, respectively, computed at $(\hat{\mathbf{q}}_k, \hat{\omega}_k)$; Q_k and Q_{k+1}^+ are the estimated and predicted state covariance matrices and, finally, $\sigma_{\omega,k}^2$ is the variance of the gyroscope noise at time kT_s. By using the same notation, the update step equations of the EKF are

$$K_{k+1}^\theta = Q_{k+1}^+ C^T \left(C Q_{k+1}^+ C^T + \sigma_{\theta^c,k+1}^2 \right)^{-1}$$

$$\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}_{k+1}^+ + K_{k+1}^g (\hat{o}_{k+1}^\theta - C\hat{\mathbf{q}}_{k+1}^+)$$

$$Q_{k+1} = (I_2 - K_{k+1}^\theta C) Q_{k+1}^+ \tag{8}$$

where K_{k+1}^θ is the Kalman gain, $C = [-1, 0]$ is the system output matrix, and $\sigma_{\theta^c,k+1}^2$ is the variance of ζ_θ^c. Note that if no landmark is detected at time kT_s, no update is possible. So the EKF works in open loop. Observe that the first element of $\hat{\mathbf{q}}_k$ is the heading value estimated at time kT_s to be injected in the overall measurement model, whereas the element $(1,1)$ of Q_k is the corresponding heading estimation variance $\sigma_{\theta,k}^2$.

C. Overall Measurement Model

The models described in Sections IV-A and IV-B pave the way to the definition of the overall measurement model equations. Observe that, in the case considered, $\hat{\theta}_k = \hat{\theta}_k^g = \theta_k + \zeta_{\theta,k}^g$, where $\zeta_{\theta,k}^g$ is the heading estimation error resulting from (8) with variance equal to $\sigma_{\theta,k}^2$. Therefore, depending on whether a QR code is detected or not, it follows immediately from (4) and (5) and (7) and (8) that:

$$\hat{\mathbf{o}}_k = \mathbf{h}(\mathbf{s}_k) + \boldsymbol{\zeta}_k$$

$$= \begin{cases} \begin{bmatrix} \mathbf{h^c(s_k)} \\ \theta_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{\zeta}_k^c \\ \zeta_{\theta,k}^g \end{bmatrix} & \text{if QR is detected} \\ \theta_k + \zeta_{\theta,k}^g & \text{if QR is not detected} \end{cases} \tag{9}$$

where $\boldsymbol{\zeta}_k$ is the vector including all the measurement uncertainty contributions, whereas $\mathbf{h}^c(\mathbf{s}_k)$ and $\boldsymbol{\zeta}_k^c$ are defined as in (5).

V. STATE ESTIMATION ALGORITHM

As known, H_∞ filters provide an optimal approach to improve robustness to unmodeled noise and dynamics. In the case considered, the power spectral density of the heading data estimated through (7) and (8) is inherently not white, as the angle values result from the Kalman filter described in Section IV-B.

In general, H_∞ filters stem from game theory and are based on the optimization of a scalar objective function [16]. In the case considered, given the operator $\|\mathbf{v}\|_M^2 \triangleq \mathbf{v}^T M \mathbf{v}$ (with $\mathbf{v}$ and M being a generic vector and square matrix, respectively),

the objective function is

$$J_k \triangleq \frac{\sum_{j=0}^{k} \| L_j \boldsymbol{s}_j - L_j \hat{\boldsymbol{s}}_j \|_{I_j}^2}{\| \mathbf{s}_0 - \hat{\mathbf{s}}_0 \|_{\tilde{P}_0^{-1}}^2 + \sum_{j=0}^{k} \| \boldsymbol{\varepsilon}_{\Phi,j} \|_{\tilde{E}_j^{-1}}^2 + \sum_{j=0}^{k} \| \boldsymbol{\zeta}_j \|_{\tilde{R}_j^{-1}}^2} \tag{10}$$

where matrix

$$L_j = \begin{cases} \begin{bmatrix} I_3 & 0_{3\times 2} \end{bmatrix} & \text{if QR is detected} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} & \text{if QR is not detected} \end{cases} \tag{11}$$

(with I_3 being the identity matrix) allows to include in the optimization of (10) just the state variables that can be observed at the jth sampling time. Since the position and orientation variables are equally relevant for localization purposes, the elements of L_j corresponding to the observed quantities are set equal to 1. On the contrary, the three weighting matrices in the denominator of (10), i.e., $\tilde{P}_0^{-1}$, $\tilde{E}_j^{-1}$, and $\tilde{R}_j^{-1}$, offer various potential degrees of freedom for H_∞ filter design, the only constraint being that they must be symmetric and positive definite.

For the problem described in this paper, the elements of $\tilde{P}_0^{-1}$, $\tilde{E}_j^{-1}$, and $\tilde{R}_j^{-1}$ depend on the experimental uncertainties associated with the individual elements of vectors $\mathbf{s}_0 - \hat{\mathbf{s}}_0$, $\boldsymbol{\varepsilon}_{\Phi,j}$, and $\boldsymbol{\zeta}_j$, respectively. As far as terms $\mathbf{s}_0 - \hat{\mathbf{s}}_0$ and $\boldsymbol{\varepsilon}_{\Phi,j}$ are concerned, they can be reasonably regarded as stationary and normally distributed, since they result from the superimposition of multiple independent contributions, in accordance with the central limit theorem. For $\boldsymbol{\varepsilon}_{\Phi,j}$, this is also confirmed by the experimental results reported in Section VI-B. Thus, $\tilde{P}_0$ and $\tilde{E}_j$ can be set equal to the respective covariance matrices P_0 and E_j, as customary in Bayesian approaches. In particular, $\tilde{E}_j = E_j = \text{diag}(\sigma_{\Phi_r,j}^2, \sigma_{\Phi_l,j}^2)$, where $\sigma_{\Phi_r,j}^2$ and $\sigma_{\Phi_l,j}^2$ are the variances of the independent noise contributions affecting the right and left wheel encoders, respectively, at time jT_s.

The elements of $\boldsymbol{\zeta}_j$ can be also reasonably assumed as uncorrelated. However, as shown in Section VI-B, they are neither Gaussian nor white. In this case, the elements of $\tilde{R}_j$ can be simply and reasonably chosen as proportional to those of the covariance matrix R_j of $\boldsymbol{\zeta}_j$, i.e.,

$$\tilde{R}_j = \begin{cases} \text{diag}\{\alpha_x^2, \alpha_y^2, \alpha_\theta^2\} \cdot R_j & \text{if QR is detected} \\ \alpha_\theta^2 R_j & \text{if QR is not detected} \end{cases} \tag{12}$$

where

$$R_j = \begin{cases} \text{diag}(\sigma_{\Delta x,j}^2, \sigma_{\Delta y,j}^2, \sigma_{\theta,j}^2) & \text{if QR is detected} \\ \sigma_{\theta,j}^2 & \text{if QR is not detected} \end{cases} \tag{13}$$

with $\sigma_{\Delta x,j}^2$, $\sigma_{\Delta y,j}^2$, and $\sigma_{\theta,j}^2$ being the variances of $\zeta_{x,j}^c$, $\zeta_{y,j}^c$, and $\zeta_{\theta,j}^g$ respectively. In both cases, the nonnegative coefficients α_x, α_y, and α_θ can be found numerically, as explained in Section VII-B.

It is worth noticing that (4) is nonlinear. So (10) should be minimized after linearizing the system equations around the state estimated at time kT_s. Unfortunately, the direct minimization of (10) generally is not a tractable problem [16]. A suboptimal solution can be found by computing the values of $\hat{\mathbf{s}}_k$ for which $\sup\{J_k\} < \gamma_k^2$, where γ_k is a time-varying threshold that can be computed adaptively, as explained in Section V-A. This

relaxed problem can be solved by using an array algorithm [30], such as the technique proposed in [31]. In this way, the EHF implementation can be split into a prediction step and an update step, like an EKF. In the case considered, the prediction step equations based on (4) are

$$\hat{\mathbf{s}}_{k+1}^+ = \hat{\mathbf{s}}_k + \mathbf{f}(\hat{\mathbf{s}}_k)\widehat{\boldsymbol{\Delta \Phi}}_k$$
$$P_{k+1}^+ = F_k P_k F_k^T + G_k \tilde{E}_k G_k^T. \tag{14}$$

where $\hat{\mathbf{s}}_k$ and $\hat{\mathbf{s}}_{k+1}^+$ denote the estimated and predicted state, respectively, P_k and P_{k+1}^+ are the corresponding covariances, and F_k and G_k are the Jacobians of the process dynamic (4) with respect to $\mathbf{s}$ and $\boldsymbol{\Delta \Phi}$, respectively, computed at $[\hat{\mathbf{s}}_k, \widehat{\boldsymbol{\Delta \Phi}}_k]$. The equations of the update step instead are [31]

$$\hat{\mathbf{s}}_{k+1} = \hat{\mathbf{s}}_{k+1}^+ + K_{k+1} \left(\mathbf{o}_{k+1} - \mathbf{h}(\hat{\mathbf{s}}_{k+1}^+) \right)$$

$$P_{k+1} = \left(I_5 - P_{k+1}^+ \begin{bmatrix} H_{k+1}^T & L_{k+1}^T \end{bmatrix} U_{k+1}^{-1} \begin{bmatrix} H_{k+1} \\ L_{k+1} \end{bmatrix} \right) P_{k+1}^+ \tag{15}$$

where H_{k+1} is the Jacobian of the output function $\mathbf{h}(\cdot)$ in (9) computed at $\hat{\mathbf{s}}_{k+1}^+$,

$$K_{k+1} = P_{k+1}^+ H_{k+1}^T \left(H_{k+1} P_{k+1}^+ H_{k+1}^T + \tilde{R}_{k+1} \right)^{-1} \tag{16}$$

with $\tilde{R}_{k+1}$ given by (12), and

$$U_{k+1} = \begin{bmatrix} \tilde{R}_{k+1} & 0 \\ 0 & -\gamma_{k+1}^2 I \end{bmatrix} + \begin{bmatrix} H_{k+1} \\ L_{k+1} \end{bmatrix} P_{k+1}^+ \begin{bmatrix} H_{k+1}^T & L_{k+1}^T \end{bmatrix}. \tag{17}$$

Notice that the dimension of the identity matrix I in (17) is equal to the number of rows of L_{k+1}.

A. Selection of Threshold γ_k

The computation of parameter γ_k in (17) deserves special attention, as it is crucially important for correct EHF operation. In principle, if $\gamma_k \to +\infty$, $L_k = I_5$, $\tilde{E}_k = E_k$, and $\tilde{R}_k = R_k$, then the EHF coincides with an EKF based on the same system model. On the contrary, the smaller γ_k, the more the EHF achieves optimality in a minimax sense. Unfortunately, if γ_k is too strict, matrix P_k in (15) might be no longer positive definite [32]. As a result, the solution of the relaxed optimization problem $J_k < \gamma_k^2$ might not exist, and the estimated trajectories could suddenly diverge. If the system is always observable, this problem can be addressed by changing γ_k adaptively, so that P_k (as well as its inverse) is positive definite for any k. In particular, by applying the matrix inversion lemma to (15), the condition of positive definiteness of P_k^{-1} can be expressed as

$$P_k^{-1} = P_k^{+^{-1}} + H_k^T \tilde{R}_k^{-1} H_k - L_k^T \gamma_k^{-2} L_k > 0. \tag{18}$$

Hence, it is shown in [33] that P_k^{-1} is positive definite for

$$\gamma_k^2 > \bar{\lambda} \left(L_k^T L_k (P_k^{+^{-1}} + H_k^T \tilde{R}_k^{-1} H_k)^{-1} \right) \tag{19}$$

where function $\bar{\lambda}(\cdot)$ returns the maximum eigenvalue of the argument matrix. Unfortunately, in the problem at hand, state

variables x_k and y_k are not always observable, unless a QR code is detected at every sampling time. Due to such intermittent observability, the estimated trajectory might occasionally diverge, even if condition (19) holds and even if θ_k is available. To avoid divergence when no QR code is detected, γ_k should be set strictly larger than (19), even if this choice may lead to suboptimal results. In particular, it can be easily shown that if

$$P_k^{+^{-1}} - L_k^T \gamma_k^{-2} L_k > 0 \quad \text{and} \quad H_k^T \tilde{R}_k^{-1} H_k - L_k^T \gamma_k^{-2} L_k \geq 0. \tag{20}$$

Then, P_k^{-1} in (18) is certainly positive definite. Thus, when no QR code is detected, the values of γ_k result, respectively, from

$$\gamma_k^2 > \bar{\lambda}(L_k^T L_k P_k^+) \quad \text{and} \quad \gamma_k^2 \geq \tilde{R}_k = \alpha_\theta^2 \sigma_{\theta,k}^2. \tag{21}$$

Observe that the rightmost expression in (21) results from the fact that $L_k = H_k$ and, consequently, $H_k^T H_k (\tilde{R}_k^{-1} - \gamma_k^{-2}) \geq 0$. Ultimately, in order to avoid finite escape time phenomena while keeping accuracy as high as possible, γ_k can be computed adaptively by merging conditions (19) and (21) as follows:

$$\gamma_k =$$
$$\begin{cases} \xi \bar{\lambda} \left(L_k^T L_k (P_k^{+^{-1}} + H_k^T \tilde{R}_k^{-1} H_k) \right)^{-\frac{1}{2}} & \text{if QR is detected} \\ \max\{\xi \bar{\lambda} \left(L_k^T L_k P_k^+ \right)^{\frac{1}{2}}, \alpha_\theta \sigma_{\theta,k}\} & \text{if QR is not detected} \end{cases} \tag{22}$$

where ξ is an arbitrary constant coefficient to be set slightly larger than 1.

B. Remarks on the Estimation Approach

The proposed estimation algorithm relies on a two-step approach since the heading measurements based on the fusion of gyroscope data and QR code orientation values (as explained in Section IV-B) are injected into the update step of the EHF based on (4). Alternatively, gyroscope and robot dynamics could be merged into a single state-space model, which could be eventually used to define a unique integrated EHF. Whereas this approach looks more efficient, the adopted two-step solution provides a comparable accuracy with a lower computational burden. Indeed, if (6) were combined with (4), not only the state vector, but also the size of all matrices would increase. Since the order of complexity of an EHF is superlinear (mainly because of matrix operations), splitting the estimation problem into two subproblems is computationally more efficient. Moreover, since the heading angle should be estimated in any case from the integral of the angular velocity values of the enlarged state vector, the EKF based on (7) and (8) is certainly optimal as the gyroscope measurement noise is white and Gaussian, as shown in Section VI-B. In conclusion, opting for a single comprehensive EHF would just increase its complexity, thus complicating filter design as well.

VI. Robotic Platform and Setup

The proposed EHF position estimation algorithm has been implemented and tested on the *FriWalk* (see Fig. 2), i.e., a wheeled robotic walking assistant developed within the EU
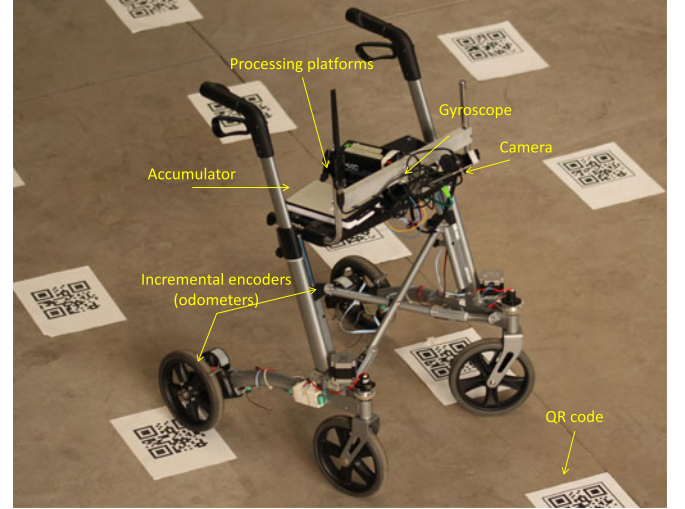

Fig. 2. Snapshot of the *FriWalk* robotic walking assistant.

project "A CyberphysicAl social NeTwOrk using robot friends" (ACANTO).

The *FriWalk* is equipped with two independent front caster wheels and two fixed coaxial rear wheels. Under the hypothesis of pure rolling, the fixed wheels impose a velocity constraint on the robot dynamic, i.e., the device cannot slide along the rear wheel axle direction. As a consequence, the vehicle has a nonholonomic constraint and the dynamic of the point chosen as a reference for localization (namely the midpoint of the rear wheels axle) can be well described by a unicycle-like model, like the one used in Section III. The robot's wheels radius and the axle length are $r = 10$ cm and $d = 59$ cm, respectively. The camera is placed at a height of about 80 and 60 cm ahead of the reference point along the longitudinal axis of the robot. As a result, $l_x^c = -60$ cm and $l_y^c = 0$ cm in (4). The main hardware and software resources of the *FriWalk* are summarized in Table I. The localization algorithm (except for QR code recognition) runs on a Beagle Bone White embedded board collecting encoders and gyroscope data over a controller area network at a rate of 250 Hz (i.e., $T_s = 4$ ms). The position estimation algorithm has been implemented in C++ (without any specific optimization option) by using a Linaro GCC[1] 5.3 toolchain along with Eigen[2] 3.2 and ZeroMQ[3] 4.2 libraries, for linear algebra operations and communication services, respectively. QR code detection and recognition (see Section VI-A) are instead performed by an Intel Nuc mini-PC. This is used for other tasks as well, such as path planning and collision avoidance, which are out of the scope of this paper.

A. QR Code Design, Detection, and Placement

As known, a QR code is a square image containing an encoded binary matrix, as shown in Fig. 1, which can store different kinds of data (e.g., numerical, alphanumerical, or bytes). In the case considered, each QR code stores only an integer number

---

[1]https://releases.linaro.org/
[2]http://eigen.tuxfamily.org/
[3]http://zeromq.org/

TABLE I

MAIN COMPONENTS OF THE *FriWalk* WALKING ASSISTANT USED TO TEST THE LOCALIZATION ALGORITHM

	Item	Features
Sensors	Encoders	CUI Inc. AMT10X, accuracy=± 15 arcmin at 2048 PPR
	Gyroscope	16-bit Invensense IMU-3000, PSD^a = $1.74 \cdot 10^4$ rad/s/$\sqrt{\text{Hz}}$
	Camera	USB PSeye RGB 640 $\times$ 480 webcam
Sensor data processing	Platform	Beagle Bone White embedded board
	Processor	AM335x ARM Cortex-A8 at 720 MHz
	Memory	256-MB DDR2 RAM
	Storage	8-GB Secure Digital (SD)
	OS	Ubuntu Linux 14.04 LTS
Image processing	Platform	Intel Nuc mini-PC
	Processor	Intel I7 5557U at 3.40 GHz
	Memory	8-GB DDR3 RAM
	Storage	256-GB solid state drive (SSD)
	OS	Ubuntu Linux 14.04 LTS

^{a}PSD = angular rate noise spectral density.

TABLE II

MEAN AND STANDARD DEVIATIONS OF THE ERRORS INTRODUCED BY THE ODOMETERS AND BY THE GYROSCOPIC PLATFORM

Sensor	Mean	Std. Deviation
Left odometer	$\eta_l = 0.01\Delta\Phi_l$ rad	$\sigma_{\Phi l} = 0.002$ rad
Right odometer	$\eta_r = 0.01\Delta\Phi_r$ rad	$\sigma_{\Phi r} = 0.002$ rad
Gyroscope platform	$\eta_\omega = 0.15\omega$ rad/s	$\sigma_\omega = 0.07|\omega| + 0.2$ rad/s

q, which is univocally associated with the planar coordinates (x_q, y_q) and the orientation angle θ_q of the QR code in the reference frame $\langle W \rangle$ of the chosen environment, as shown in Fig. 1. This approach is very flexible since the table associating each QR code number to a triple (x_q, y_q, θ_q) can be easily changed and adapted to different environments with no need to reprint the codes. Also, in this way just low-density numeric-only codes can be used. All QR codes were generated by an online tool[4] according to the ISO/IEC Standard 18004:2006 and were printed on regular A4 paper sheets with a resolution of 600 dpi. As a rule of thumb, QR code size should be one order of magnitude smaller than the scanning range. Also, their size should be proportional to a data density factor given by the ratio between the number of columns (or rows) of the chosen QR code and the number of columns (or rows) of a standard version-2 code (i.e., 25). Resilience to data loss is provided by Reed–Salomon error correction coding. Since, in the case considered, the scanning range of the chosen camera is about 1.2 m, version-1 type L (i.e., low level) QR codes with a side of 15 cm and just 21×21 black-and-white cells are used. This choice provides a good tradeoff between detectability (which is never smaller than about 75% if a QR code lies within the field of view of the camera) and error recovery capability (up to 7% of corrupted data). In principle, denser QR codes (e.g., type H) could provide better error recovery [34]. However, they would be also harder to detect unless larger landmarks were used, which would be impractical. QR code detection and recognition are performed using the open source libraries Zbar[5] and OpenCV.[6]

In general, the QR code placement strategy strongly depends on rooms geometry. If no specific constraints exist, a reasonable approach is to deploy the QR codes over a regular lattice. In theory, only three periodic, monohedral, and regular tiling patterns can be designed over the plane, i.e., equilateral triangles, squares, or hexagons. Among them, the square pattern is definitely the easiest to deploy in practice. Thus, this solution is adopted in the following.

B. Sensor Uncertainty Evaluation

In this section, the procedures to evaluate the uncertainty of *FriWalk*'s sensors are described and the corresponding results are reported. This step is essential to justify the use of an EHF.

1) Odometers: The systematic and random uncertainty contributions of odometry-based measurements were evaluated on the field by driving the robot repeatedly at different constant velocities (from 0.1 m/s to 2.5 m/s) over a square path. The statistics of ε_Φ at different velocities confirm that the odometers' uncertainty exhibit a normal distribution. Table II reports the mean and the standard deviation of the odometers' uncertainty distributions (after one sampling period) as a function of the angular velocity of each wheel. Results are basically the same for the right and left wheel, as expected. However, whereas the standard deviations do not depend on the angular velocity (i.e., $\sigma_{\Phi l} \approx \sigma_{\Phi r} = \sigma_\Phi = 2$ mrad $\forall k$), the mean values η_r and η_l tend to grow linearly with $\Delta\Phi_r$ and $\Delta\Phi_l$, respectively. The goodness of the linear fitting model justifies the use of state variables μ and δ in (2).

2) Gyroscopic Platform: The accuracy of the gyroscopic platform was evaluated by means of a calibrated orbital rotator Stuart SB3 in the range [2–40] RPM. The data analysis shows that in all cases, the noise patterns exhibit a white power spectral density and a normal distribution. However, both the mean value η_ω and the standard deviation σ_ω of such distributions tend to grow linearly with the angular velocity ω around axis Z_w. The expressions of η_ω and σ_ω obtained through a linear fitting are reported in Table II. The slope of η_ω is quite large and, to a first approximation, it can be reasonably assumed to be constant. This result confirms that the systematic error introduced by the gyroscope can be successfully modeled by state variable b^g in (6). Observe that, even if the noise distribution is
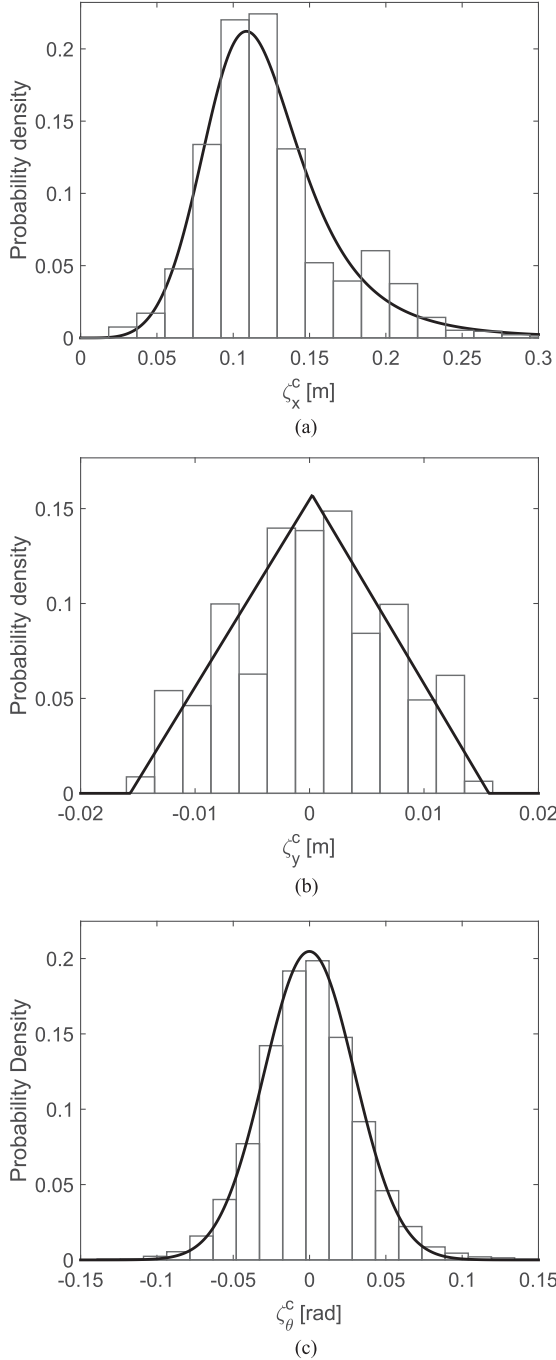
[4]http://goqr.me/

[5]http://zbar.sourceforge.net/
[6]http://opencv.org/

Fig. 3. Estimated probability density functions (PDFs) of camera-based measurement uncertainties ζ_x^c, ζ_y^c, and ζ_θ^c in (a), (b), and (c), respectively, when a QR code is detected and the *FriWalk* robot is in motion.

globally normal, the value of σ_ω is inherently nonstationary due to its dependence on ω. Therefore, the expression of $\sigma_{\omega,k}^2$ to be used in (7) must be computed at every sampling time.

3) Vision System: In order to estimate the PDFs of uncertainty terms ζ_x^c, ζ_y^c, and ζ_θ^c, the robot was repeatedly driven towards a target landmark in different directions and at different speeds till detecting the QR code. Fig. 3(a)–(c) shows the PDFs of ζ_x^c, ζ_y^c, and ζ_θ^c estimated using about 100 experimental data. The black lines correspond to a log-logistic, a triangular, and a

Gaussian distribution for ζ_x^c, ζ_y^c, and ζ_θ^c, respectively. It is worth noticing that the latencies to extract the measures from the collected images are random and can be so large as 150 ms. As a result, the displayed distributions of ζ_x^c, ζ_y^c, and ζ_θ^c are significantly affected by such latencies. In particular, ζ_x^c, ζ_y^c, and ζ_θ^c are quite uncorrelated when the robot is in motion because the total position and heading uncertainty depend not only on the camera, but also on the trajectory of the robot during the random time interval in which the information of a QR code is processed and transferred to the Beagle Bone. This result confirms that matrix R_j in (12) can be reasonably assumed to be diagonal. The difference between the distributions of ζ_x^c and ζ_y^c is mainly due to the adopted setup. Since in the experiments considered (as well as in typical scenarios) the camera field of view is maximum in the direction of its optical axis, the impact of image acquisition and processing latencies is much larger on Δx^c than on Δy^c. Also, Δx^c is overestimated because the robot mainly moves forward. As a result, the distribution of ζ_x^c is skewed with a mean value of about 12 cm, whereas ζ_y^c exhibits a zero mean because the probabilities of detecting a QR code located on the right or on the left of the optical axis are approximately equal. Similarly, the mean value of ζ_θ^c is zero, as the robot can be reasonably assumed to turn left and right with equal probability. Therefore, just the mean value of Δx^c has to be properly compensated (i.e., subtracted from $\widehat{\Delta x}^c$) prior to injecting the measured data into the EHF. Quite importantly, the fact that both ζ_x^c and ζ_y^c are not normally distributed confirms the idea of using an EHF. On the contrary, the seeming Gaussianity of ζ_θ^c justifies the use of the EKF based on (7) and (8) for heading estimation.

4) Position and Heading Measurement Uncertainty: The overall measurement uncertainty vector $\boldsymbol{\zeta} = [\zeta_x, \zeta_y, \zeta_\theta]^T$ in (9) deserves some further explanation. The distributions of ζ_x and ζ_y coincide with those of ζ_x^c and ζ_y^c, respectively. So they can be assumed to be stationary (i.e., independent of k). On the contrary, the distribution of ζ_θ is strongly time varying, as it results from (8). In addition, the distribution of ζ_θ depends on QR code density. As a result, the power spectral density of ζ_θ is definitely not white and it is more similar to a random walk process. However, this is not an issue if an EHF is used.

VII. EXPERIMENTAL RESULTS

To test the behavior of the EHF and to perform a realistic comparison with other position estimation algorithms, the *FriWalk* prototype was driven along multiple random paths in the premises of the University of Trento. The main parameters of the chosen experimental setup are listed in Table III. The position

of the *FriWalk* along every path (i.e., the "ground truth") was reconstructed by using a Sick S300 Expert Laser scanner. The accuracy of this instrument (i.e., 30 mm and 0.5° in polar coordinates for objects up to 30 m far) is indeed much higher than the accuracy achievable with the sensors installed on the robot.

A. EHF Parameters Settings

As known, the parameter values of an EHF have to be chosen on the basis of the specific problem. In the case considered, $\tilde{E}_k = E_k = \text{diag}(4 \cdot 10^{-6} \text{ rad}^2, 4 \cdot 10^{-6} \text{ rad}^2)$ in accordance with the results reported in Section VI-B. The elements of $\tilde{P}_0 = P_0$ can be set very large (e.g., from one to two orders of magnitude larger than the expected values), since no *a priori* information is usually available on the robot's state. The value of parameter γ_k in (17) results from (22), as described in Section V-A. Finally, the elements of $\tilde{R}_k$ are given by (12), with $R_k = \text{diag}(1.6 \cdot 10^{-3} \text{ m}^2, 4.9 \cdot 10^{-5} \text{ m}^2, \sigma_{\theta,k}^2 \text{ rad}^2)$ or $R_k = \sigma_{\theta,k}^2 \text{ rad}^2$, depending on whether a QR code is detected or not, in accordance with the results shown in Fig. 3. It is worth reminding that $\sigma_{\theta,k}^2$ is the variance of the heading angle measured as explained in Section IV-B; therefore it changes over time. Unfortunately, the problem of finding the best values of α_x, α_y, and α_θ coefficients in (12) is intractable analytically. So it was addressed through simulations based on experimental data. To this end, the EHF was repeatedly applied offline to the same set of raw sensor data collected by the *FriWalk*, for different values of α_x, α_y, and α_θ. From the comparison between the estimated results and those obtained with the laser scanner, the following was observed empirically.

1) The best results are obtained when $\alpha_x \approx \alpha_y$. Thus, a single common coefficient (called α_p in the following) can replace both α_x and α_y in (12).
2) Two suboptimal values of coefficients α_p and α_θ can be derived heuristically by finding the pair which minimizes the 99th percentile of the Euclidean distance between actual and estimated positions.

The choice of using the 99th percentile of the position error as a performance index is dictated by the fact that the EHF is optimal in a minimax sense. So the EHF parameters should be chosen to minimize the worst-case errors, while filtering possible outliers that could make the estimated maxima excessively noisy. Fig. 4 shows the 99th percentile curves for different values of α_p and α_θ when $D = 2$ m. The results obtained with a different QR code density exhibit a similar trend and are not reported for the sake of brevity.

To highlight the most accurate configuration, the 99th percentiles of the position errors are plotted as a function of the ratio α_θ / α_p for several values of α_θ. The minima of the curves in Fig. 4 do not change significantly for $\alpha_\theta \geq 100$. Thus, EHF accuracy is optimal for $\alpha_\theta \approx 100$ and $\alpha_p = \alpha_\theta / 10^{1.25} \approx 5.6$.

B. Performance Analysis and Comparison

This section reports a performance analysis as well as a comparison between the proposed EHF (with the optimal settings computed as described in Section VII-A) and an EKF, a UKF and a PF based on the very same models described in
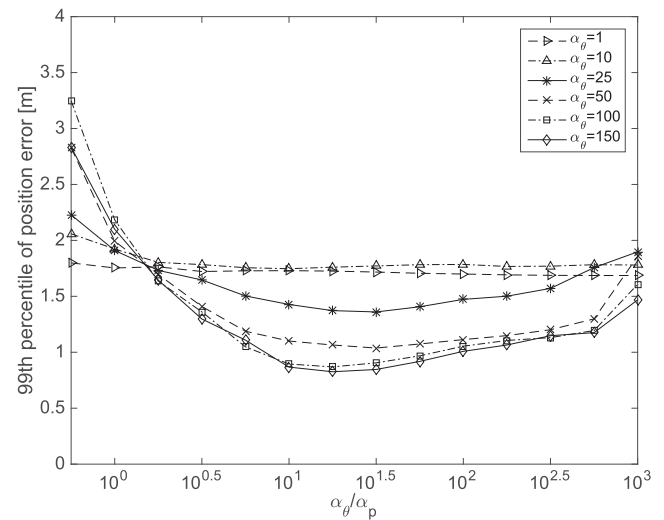

Fig. 4. 99th percentiles of the Euclidean position error as a function of ratio α_θ / α_p. Each curve is obtained with a different value of α_θ.
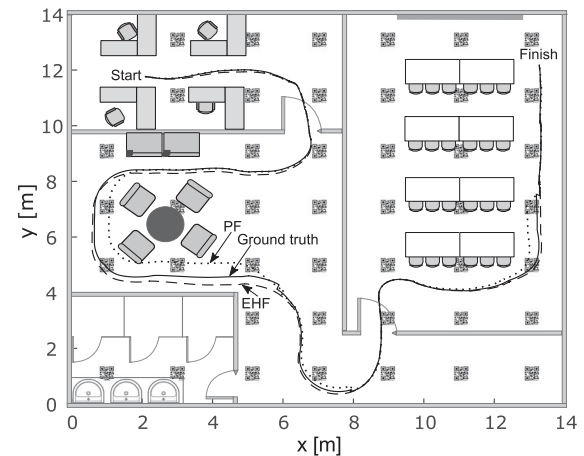

Fig. 5. Sample path in a real-world scenario.

Sections III and IV. EKF, UKF, and PF rely on state-of-the-art approaches as described in [29], [35], and [15], respectively. In the PF case, a sequential importance resampling technique is used. Particles are taken from a normal distribution with a standard deviation about ten times larger than the actual value to avoid the so-called sample depletion problem [15]. In particular, 1000 particles are generated at every time. This number of particles ensures indeed good accuracy, while keeping the computational burden reasonable, although real-time performance can be hardly guaranteed with the chosen processing platform, as it will be shown later.

A first performance comparison in a real-world scenario is shown in Figs. 5 and 6. In the map shown in Fig. 5, the distance D between adjacent QR codes is 2 m, as this choice provides a reasonable tradeoff between performances and deployment complexity. In the case considered, EHF and PF are globally more accurate than EKF and UKF. So, only the trajectories estimated by EHF and PF are plotted in Figs. 5 and 6, not to overcrowd the picture. The corresponding estimation errors of state variables x, y, and θ (solid lines for EHF and dashed lines

TABLE IV
ROOT MEAN SQUARE (RMS) AND 99TH PERCENTILES OF THE x, y, AND θ ESTIMATION ERRORS OBTAINED WITH VARIOUS ALGORITHMS USING SQUARE-PATTERNED GRIDS OF QR CODES OF DIFFERENT DENSITY

|		$D = 1$ m				$D = 2$ m				$D = 3$ m				$D = 4$ m			
		EHF	EKF	UKF	PF	EHF	EKF	UKF	PF	EHF	EKF	UKF	PF	EHF	EKF	UKF	PF
x	RMSE [cm]	10	15	10	10	15	25	25	20	20	35	35	30	25	40	40	35
	99th Perc. [cm]	35	50	45	40	50	105	100	90	60	130	125	95	75	140	135	110
y	RMSE [cm]	10	15	10	10	15	30	25	20	20	40	35	30	20	45	40	35
	99th Perc. [cm]	35	55	45	40	45	100	95	85	55	130	120	100	70	150	140	120
θ	RMSE [crad]	5	10	10	5	5	10	10	10	10	15	15	10	10	15	15	10
	99th Perc. [crad]	20	30	30	25	25	40	40	30	30	40	40	35	30	45	45	40
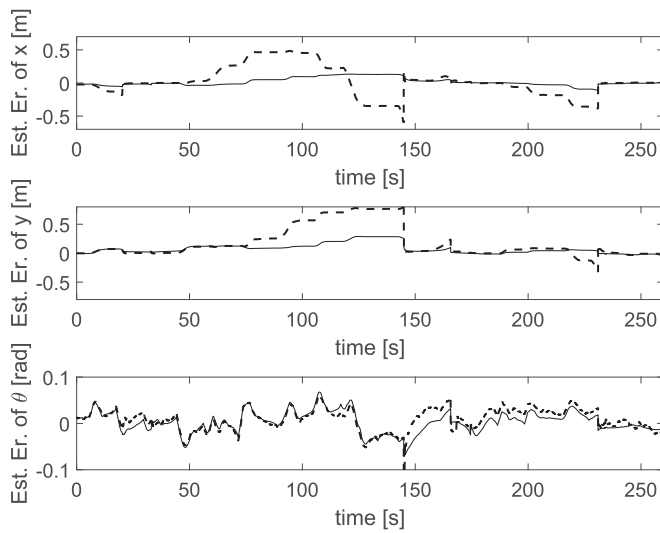
Fig. 6. Estimation errors of state variables x, y, and θ obtained when the EHF (solid lines) and the PF (dashed lines) are used to track the position of the *FriWalk* along the path depicted in Fig. 5.

for PF) are shown in Fig. 6 as a function of time. Clearly, the maximum EHF estimation errors along axes x and y are smaller than those obtained with the PF (i.e., ± 50 cm), whereas the angular errors are generally within ± 0.1 rad in all cases. This behavior is due to the intermittent observability of system (4), which in turn depends on the random availability of QR code measures at a given sampling time. In fact, this depends not only on the field of view of the camera, but also on QR code density and on the robot's actual path. Some Monte Carlo simulations have shown that the average length of the path between two detected landmarks tends to grow quadratically with D, whereas the worst-case values range from about 10 m for $D = 1$ m to about 80 m for $D = 4$ m. This happens, for instance, in the central part of the path shown in Fig. 5 and explains why the estimation error (which depends solely on dead reckoning in this part) tends to grow. Of course, such intermittent observability also affects convergence and explains why the results reported in this paper are quite different from those of other QR-based localization solutions where at least one landmark is always in view [25]. Despite this difference, convergence is always

guaranteed if the robot moves randomly over an area evenly covered by landmarks, as a QR is certainly detected after a finite time and, consequently, position estimation uncertainty is kept bounded. This behavior can be hardly described analytically, but it holds regardless of the chosen position estimation algorithm, although accuracy may differ considerably. In this respect, the proposed EHF is expected to perform better than the other estimators because it is designed to be optimal in a minimax sense.

A more detailed and complete accuracy comparison between EHF, EKF, UKF, and PF is shown in Table IV. This table reports the RMS values and the 99th percentiles of the estimation errors of state variables x, y, and θ computed over more than 40 random paths when different grids of QR codes are used (i.e., with D equal to 1, 2, 3, or 4 m). Both the RMS and 99th percentile estimation errors are computed after the first QR code is detected. This event occurs at a random time, ranging from 2–3 s when $D = 1$ m, up to about 1 min when $D = 4$ m. As a result, the average number of distinct QR codes that are detected over a one meter long path ranges between 0.07 when $D = 4$ m and 0.7 when $D = 1$ m.

From Table IV, it can be clearly recognized that the EHF outperforms both the EKF and the UKF. Also, the EHF is slightly more accurate than the PF with 1000 particles, especially in the worst-case. Observe that when $D = 1$ m, the accuracy of all algorithms becomes comparable. This is due to the fact that in this case localization accuracy is dominated by the performance of the vision system. However, the most interesting results are those obtained when D is larger, i.e., when the landmark density is low. It is worth emphasizing that if state variables μ and δ were not included in the state vector model of (1), the estimation uncertainty of variables x, y, and θ would be larger in all conditions, because the systematic offsets introduced by the odometers could not be estimated and compensated. In particular, the RMS estimation errors would increase by a variable amount (i.e., from about 2% to about 20%), depending on both the specific estimation algorithm and the QR code density adopted.

Table V reports the mean convergence time and the mean computation time (with and without using QR code data in the update step) of all estimation algorithms when $D = 1$ m, i.e., when accuracy is maximum.

TABLE V
CONVERGENCE TIME AND COMPUTATION TIME OF THE EHF, THE EKF, THE
UKF, AND THE PF WHEN $D = 1$ M

	EHF	EKF	UKF	PF
Convergence time [s]	18	9	2	11
Computation time (no QR code update) [ms]	0.09	0.06	0.25	0.80
Computation time (with QR code update) [ms]	0.26	0.16	0.60	6.69

In particular, the convergence time of a single experiment is estimated by computing the average time intervals after which the estimation uncertainties of state variables (x, y, θ) become smaller than the respective 99th percentile reported in Table IV. The results in Table V show that the EHF exhibits the slowest convergence. However, in terms of average computation time, the EHF is just slightly slower than the EKF, but it is much faster than both the UKF and the PF, especially when a QR code is used to update the state of the system. Moreover, with the adopted processing platform, the EKF and the EHF can always return a result in real time (i.e., within $T_s = 4$ ms), whereas the UKF and, above all, the PF may occasionally miss the deadline. This is evident especially in the PF case, since its average computation time is always larger than 4 ms when a QR code is detected.

VIII. CONCLUSION

In this paper, an EHF for indoor localization of unicycle-like robots is described, optimized, implemented, and finally characterized. The proposed approach relies on the following:

1) odometry and gyroscope data that ensure continuous position tracking;
2) QR code recognition to observe and to adjust robot's position and heading;
3) an adaptive mechanism to avoid EHF finite escape time phenomena.

An extensive experimental campaign as well as a performance comparison with alternative estimators based on the same system model (i.e., a classic EKF, an UKF, and a PF) shows that the proposed EHF can provide very good performances with a reasonable computational burden. Indeed, even if the EHF converges more slowly than the others, its accuracy is higher. Also, the EHF computation time is a bit higher than the EKF, but it is lower than both the UKF and the PF. This result is due to the intrinsic nature of the EHF, which is explicitly conceived to minimize the maximum estimation error even if the measurement uncertainty contributions are completely unknown. While estimation accuracy is currently limited by the low-cost sensors of the adopted robotic platform, the proposed approach is general enough to be applied to more sophisticated robots as well.

ACKNOWLEDGMENT

REFERENCES

[1] H. Chen, D. Sun, J. Yang, and J. Chen, "Localization for multirobot formations in indoor environment," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 4, pp. 561–574, Aug. 2010.

[2] P. Chen, Y. B. Xu, L. Chen, and Z. A. Deng, "Survey of WLAN finger-printing positioning system," *Appl. Mech. Mater.*, vol. 380, pp. 2499–2505, Aug. 2013.

[3] R. Mautz and S. Tilch, "Survey of optical indoor positioning systems," in *Proc. Int. Conf. Indoor Positioning Indoor Navig.*, Guimarães, Portugal, Sep. 2011, pp. 1–7.

[4] F. Ijaz, H. K. Yang, A. Ahmad, and C. Lee, "Indoor positioning: A review of indoor ultrasonic positioning systems," in *Proc. 15th Int. Conf. Adv. Commun. Technol.*, Phoenix Park, South Korea, Jan. 2013, pp. 1146–1150.

[5] M. M. Atia, S. Liu, H. Nematallah, T. B. Karamat, and A. Noureldin, "Integrated indoor navigation system for ground vehicles with automatic 3-D alignment and position initialization," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1279–1292, Apr. 2015.

[6] M. J. Gallant and J. A. Marshall, "Two-dimensional axis mapping using LiDAR," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 150–160, Feb. 2016.

[7] H. Song, W. Choi, and H. Kim, "Robust vision-based relative-localization approach using an RGB-depth camera and LiDAR sensor fusion," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3725–3736, Jun. 2016.

[8] S. W. Yang and C. C. Wang, "On solving mirror reflection in LIDAR sensing," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 255–265, Apr. 2011.

[9] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman, "A framework for vision based bearing only 3D SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, May 2006, pp. 1944–1950.

[10] A. Rusdinar, J. Kim, J. Lee, and S. Kim, "Implementation of real-time positioning system using extended Kalman filter and artificial land-mark on ceiling," *J. Mech. Sci. Technol.*, vol. 26, no. 3, pp. 949–958, 2012.

[11] B. Dzodzo, L. Han, X. Chen, H. Qian, and Y. Xu, "Realtime 2D code based localization for indoor robot navigation," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Shenzhen, China, Dec. 2013, pp. 486–492.

[12] G. Panahandeh and M. Jansson, "Vision-Aided inertial navigation based on ground plane feature detection," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1206–1215, Aug. 2014.

[13] J. Simanek, M. Reinstein, and V. Kubelka, "Evaluation of the EKF-based estimation architectures for data fusion in mobile robots," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 985–990, Apr. 2015.

[14] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: Introduction of contextual aspects," *Inf. Fusion*, vol. 7, no. 2, pp. 221–230, Jun. 2006.

[15] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 53–82, Jul. 2010.

[16] D. Simon, *Optimal State Estimation: Kalman, H∞, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[17] F. Wang, Z. Wang, S. Lauria, and X. Liu, "Mobile robot localization using robust extended H∞ filtering," *Proc. Inst. Mech. Eng., Part I: J. Syst. Control Eng.*, vol. 223, no. 8, pp. 1067–1080, Dec. 2009.

[18] H. Hur and H.-S. Ahn, "Unknown input h∞ observer-based localization of a mobile robot with sensor failure," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 6, pp. 1830–1838, Dec. 2014.

[19] F. Fraundorfer, C. Engels, and D. Nister, "Topological mapping, localization and navigation using image collections," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, USA, Oct. 2007, pp. 3872–3877.

[20] M. Bloesch *et al.*, "Fusion of optical flow and inertial measurements for robust egomotion estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL, USA, Sep. 2014, pp. 3102–3107.

[21] D. Fontanelli, A. Danesi, F. A. Belo, P. Salaris, and A. Bicchi, "Visual servoing in the large," *Intl. J. Robot. Res.*, vol. 28, no. 6, pp. 802–814, 2009.

[22] K. Wang, Y. H. Liu, and L. Li, "A simple and parallel algorithm for real-time robot localization by fusing monocular vision and odometry/AHRS sensors," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1447–1457, Aug. 2014.

[23] C. Gionata, F. Francesco, F. Alessandro, I. Sabrina, and M. Andrea, "An inertial and QR code landmarks-based navigation system for impaired wheelchair users," in *Ambient Assisted Living*, S. Longhi, P. Siciliano, M. Germani, and A. Monteri, Eds. New York, NY, USA: Springer, 2014, pp. 205–214.

[24] A. Basiri, P. Amirian, and A. Winstanley, "The use of quick response (QR) codes in landmark-based pedestrian navigation," *Int. J. Navig. Observ.*, vol. 2014, pp. 1–7, Apr. 2014.

[25] H. Zhang, C. Zhang, W. Yang, and C. Y. Chen, "Localization and navigation using QR code for mobile robot in indoor environment," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Zhuhai, China, Dec. 2015, pp. 2501–2506.

[26] P. Nazemzadeh, F. Moro, D. Fontanelli, D. Macii, and L. Palopoli, "Indoor positioning of a robotic walking assistant for large public environments," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 11, pp. 2965–2976, Nov. 2015.

[27] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, vol. 26. New York, NY, USA: Springer, 2012.

[28] P. Nazemzadeh, D. Fontanelli, and D. Macii, "An indoor position tracking technique based on data fusion for ambient assisted living," in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl.*, Milan, Italy, Jul. 2013, pp. 7–12.

[29] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation With Application to Tracking and Navigation—Theory, Algorithm and Software*. Hoboken, NJ, USA: Wiley, 2001.

[30] B. Hassibi, T. Kailath, and A. Sayed, "Array algorithms for H_∞ estimation," *IEEE Trans. Automat. Control*, vol. 45, no. 4, pp. 702–706, Apr. 2000.

[31] W. Li and Y. Jia, "H-infinity filtering for a class of nonlinear discrete-time systems based on unscented transform," *Signal Process.*, vol. 90, no. 12, pp. 3301–3307, 2010.

[32] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Trans. Automat. Control*, vol. 34, no. 8, pp. 831–847, Aug. 1989.

[33] X. Shen and L. Deng, "A dynamic system approach to speech enhancement using the H_∞ filtering algorithm," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 4, pp. 391–399, Jul. 1999.

[34] K. H. Pandya and H. J. Galiyawala, "A survey on QR codes: In context of research and application," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 7, no. 3, pp. 258–262, Mar. 2014.

[35] E. A. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Lake Louise, AB, Canada, Oct. 2000, pp. 153–158.

Daniele Fontanelli (M'09) received the M.S. degree in information engineering and the Ph.D. degree in automation, robotics, and bioengineering from the University of Pisa, Pisa, Italy, in 2001 and 2006, respectively.

From 2006 to 2007, he was a Visiting Scientist in the Vision Lab, University of California at Los Angeles, Los Angeles, CA, USA. From 2007 to 2008, he was an Associate Researcher in the Interdepartmental Research Cente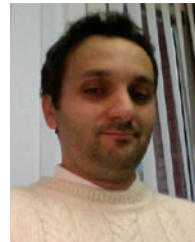r "E. Piaggio," University of Pisa. He is currently an Assistant Professor with the Department of Industrial Engineering, University of Trento, Trento, Italy. His research interests include real-time control and estimation, localization algorithms, wheeled mobile robots control, and service robotics.

David Macii (M'06–SM'14) received the M.S. degree in electronic engineering and the Ph.D. degree in information engineering from the University of Perugia, Perugia, Italy, in 2003.

He was a Visiting Researcher at the Berkeley Wireless Research Center, University of California at Berkeley, Berkeley, CA, USA, as a Fulbright Research Scholar, between 2009 and 2010. He is currently an Associate Professor with the Department of Industrial Engineering, University of Trento, Trento, Italy. He is author and co-author of more than 100 papers published in books, scientific journals, and international conference proceedings. His research interests include measurement and estimation techniques based on digital signal processing.

Payam Nazemzadeh (S'13–M'17) received the M.S. degree in mechatronics engineering from the University of Semnan, Semnan, Iran, and the Ph.D. degree in information and communication technology from the University of Trento, Trento, Italy, in 2011 and 2016, respectively.

He was a Visiting Researcher in the Department of Signal Processing, KTH, Royal Institute of Technology, Stockholm, Sweden, in 2015. He is currently a Senior Researcher with the Research Institute San Raffaele, Milan, Italy. His research interests include estimation theory, indoor localization, sensor fusion, and wearable devices.

Luigi Palopoli (M'16) received the graduate degree in computer engineering from the University of Pisa, Pisa, Italy, and the Ph.D. degree in computer engineering from Scuola Superiore Sant'Anna, Pisa, Italy, in 1998 and 2002, respectively.

He is currently an Associate Professor with the Department of Information Engineering and Computer Science, University of Trento, Trento, Italy. He has been the coordinator of EU research projects projects DALi (FP7) and ACANTO (Horizon 2020). His research interests include embedded system design, with a special focus on real-time embedded control, formal methods, and stochastic analysis of real-time systems, especially for service robotics.