Indoor Positioning using efficient Map Matching, RSS measurements and an improved motion model

Francisco Zampella, Antonio R. Jiménez, Fernando Seco

Abstract—Unlike outdoor positioning, there is not a unique solution to obtain the position of a person inside a building or in GNSS denied areas. Typical implementations indoor rely on dead reckoning or beacon based positioning, but a robust estimation must combine several techniques to overcome their own drawbacks. In this paper, we present an indoor positioning system based on foot mounted Pedestrian Dead Reckoning (PDR) with an efficient Map Matching, Received Signal Strength (RSS) measurements and an improved motion model that includes the estimation of the turn rate bias. The system was implemented using a two levels structure with a low level PDR-filter and a high level particle filter to include all the measurements.

After studying the effect of the step displacement on the particle filters proposed in the literature, we concluded that a new state with the turn rate bias (a non observable state in PDR) is needed to correctly estimate the error growth and in the long term, correct the position and heading estimation. Additionally, the wall crossing detection of Map Matching was optimized as matrix operations, and a room grouping algorithm was proposed as a way to accelerate the process, achieving real time execution with more than 100000 particles in a building with more than 600 wall segments. We also include a basic path loss model to use RSS measurements that allows a better initialization of the filter, less particles, and a faster convergence, without the need of an extensive calibration. The inclusion of the Map Matching algorithm lowers the error level of the RSS-PDR positioning from 1.9 m to 0.75 m, 90% of the time. The system is tested in several trajectories to show the improved estimation in the improved positioning, the time to convergence and the required number of particles.

Index Terms—Indoor Positioning, foot mounted Pedestrian Dead Reckoning, Map Matching, particle filter, RSS.

I. INTRODUCTION

The position of a person is a valuable information used in many applications ranging from vehicle navigation, ambient assisted living, location dependant advertisement, first responders navigation, among many others. Outdoors, Global Navigation Satellite Systems (GNSS) like GPS, GLONASS, BEIDOU or GALILEO are able to provide a reasonably good positioning [1], but indoors, the attenuations and reflections of the satellite signals limit their use.

Indoors (or in GNSS denied areas), many alternative systems have been presented, usually based on triangulation or trilateration of Radio Frequency (RF) signals [2], ultrasound signals (US) [3], camera images [4], etc. These Systems offer absolute positioning with a limited error level, but in non line of sight conditions, they generate significant errors in the localization. Another possibility is the reconstruction of the trajectory of a pedestrian using inertial measurements and the detection of the steps, usually known as Pedestrian Dead Reckoning (PDR) [5], [6]. These systems offer a good relative positioning/motion model, but suffer from an increasing drift due to the bias and noise in the measurements, and require an initial position/orientation estimate. Each technology alone does not provide a final solution to the positioning problem, but fusing different systems, it is possible to significantly improve the localization.

The bias of the gyroscope is the main source of error in PDR estimations and since the first implementations in [5] it has been included in the estimation states. However, with zero velocity measurements during the stance phases, the yaw and the vertical component of the turn rate are non observables, and this creates an increasing error in those states that after some time affects the whole estimation. In [7] the author proposed the use of magnetic measurements, but that correction is only feasible when the magnetic disturbance in the area are low, while in [8] the correct magnetic field is detected using an array of magnetometers. As a way to avoid the magnetic perturbations, in [9] and [10], the authors use the change of the observed rotations of the magnetic fields to estimate the turn rate biases. In these cases the estimation requires a magnetometer to estimate the bias.

The fusion of a PDR method and triangulation or trilateration methods offers a good solution to the positioning problem. In particular, RF signals as proposed in [11], [7] and [12] are preferred due to the fact that the signals are usually available even without a line of sight, although with higher measurement errors. This combination is used due to the movement model provided by the inertial measurements and the bounded error level of the RF signals, able to limit the drift of PDR. These systems are able to provide an error bellow 2 m, 90% of the time (in [7] the author uses PDR and RSS path-loss measurements in an EKF with 50 RFID tags in a 2800 m² area), but the estimation is usually deviated indoors due to the non line of sight effects in the RF signals.

The use of the information about the walls in the building, also allows to limit the drift and obtain the initial position and heading of a PDR estimation. These wall corrections generates highly non linear measurements associated with the position of the person and the authors in [13], [14], [15] and [16] proposed an algorithm based on wall crossing detections in a two level scheme, a particle filter (PF, high level filter) using
the step information obtained from PDR (low level filter).

The two level scheme has been widely used to include many non-linear measurements like maps, WiFi-RSS positioning and off-line path smoother [17], angle of arrival, time difference of arrivals and RFID-RSS measurements [18], WiFi-RSS fingerprinting and maps [19], etc. Other authors use the scheme to implement Simultaneous Location and Mapping (SLAM) using as a reference the paths of a user [20], the WiFi signals [21] or even the user’s recognized actions [22].

This paper will focus on positioning of first responders using an advanced data fusion scheme to include map information, active RFID devices acting as beacons and mobility models for tracking, achieving position dependant parameter estimation and a robust performance. Three incremental improvements and new results over the state of the art are proposed: an advance step motion model, a vectorized wall crossing detection and a room grouping algorithm, each of then explained next.

We have observed that the step information used in particle filters might generate wrong estimations of the possible positions of the person and we will introduce a step model that includes the turn rate bias as a better representation of the step displacements in a particle filter. Additionally, in buildings with a significant number of walls, if the number of particles is too big, the filter takes too much time detecting the wall crossings and can not be executed in real time. We propose a vectorization of the wall crossing detection operations and an algorithm that groups the particles according to the rooms to accelerate the process.

The remainder of this article is organized as follows: Section II explains the structure of the two level filter used. Section III explains the proposed improvement in the motion model while Section IV proposes improvements in the Map Matching algorithm. Section V describe the proposed experiments and the obtained results, and finally Section VI offers the conclusions and future work.

II. SENSOR FUSION WITH A TWO LEVELS FILTER

In this section we will explain the general fusion scheme used, based on a two level structure as previously proposed in [13], [14], [15]. The lower level is based on a PDR estimation, using measurements received at 100 Hz. The upper level is based on a particle filter divided into a prediction phase (propagate the particles using the PDR displacements after each step, approximately 1 Hz) and a correction phase (modify the weights of the particles according to any measurement observed like RF signals or map information). The proposed fusion scheme can be observed in figure 1.

A. Lower level, Pedestrian Dead Reckoning

The inertial measurements can be used to determine the relative position of a pedestrian without the need of external sensors. The most common techniques used are: 1) the estimate of the step length and orientation change of each stride observing the characteristics of the IMU readings during the steps [23], [24] and 2) if the IMU is located in the foot, the use of an Inertial Navigation System (INS) [25] corrected with Zero velocity UPdaTes (ZUPT) during the stance phases of the step as proposed in [5], [6]. The first option offers a flexible method but with a higher increase in the error after each step, while the second option offers a better positioning at the cost of limiting the possible positions of the IMU. In the present work we will use the second option due to the lower error, but it can be adapted to use the first technique. The low level PDR is represented as the gray box in figure 1.

In the INS-ZUPT technique, the different navigation states can be propagated from an initial position \( \mathbf{r}^o \), velocity \( \dot{\mathbf{r}}^o \) and orientation (all in the navigation reference frame, superindex \( n \)) using the following equations:

\[
\mathbf{r}^o(k + 1) = \mathbf{r}^o(k) + (\dot{\mathbf{r}}^o(k) + \dot{\mathbf{r}}^o(k + 1)) \cdot \Delta T/2,
\]

\[
\dot{\mathbf{r}}^o(k + 1) = \dot{\mathbf{r}}^o(k) + \left( C^o_n(k) \cdot \mathbf{a}^b(k) - \mathbf{g} \right) \cdot \Delta T,
\]

\[
C^o_n(k + 1) = C^o_n(k) \cdot \exp([\omega^b(k) \times] \cdot \Delta T),
\]

where the estimates are represented with a hat (˘), \( \Delta T \) is the sampling interval, the orientation is represented using the Direction Cosine Matrix (DCM) \( C^o_n \) that rotates the measurements from the sensor reference frame (values with superindex \( b \)) to the navigation reference frame, \( \mathbf{a}^b \) and \( \omega^b \) are the acceleration and turn rate measured in the sensor’s reference frame, \( \mathbf{g} \) is the gravity vector, and \([\omega^b \times]\) is the skew symmetrical matrix of \( \omega^b \):

\[
[\omega^b \times] = \begin{bmatrix}
0 & -\omega^b_z & \omega^b_y \\
\omega^b_z & 0 & -\omega^b_x \\
-\omega^b_y & \omega^b_x & 0
\end{bmatrix}.
\]

A typical INS mechanization usually has an error that increases as the cube of the time, something that with a consumer grade IMU can generate significant positioning errors after less than a minute. The use of ZUPT resets the velocity and part of the orientation error after each step, lowering the error increment to a percentage of the total traveled distance (usually between 0.2 % and 5 %).

The ZUPT measurement is usually implemented using an Extended Kalman Filter (EKF) to estimate the errors in the navigation states (position, velocity, orientation and the bias in the accelerometer and gyroscope). In this implementation we used as the states \( \mathbf{x} = [\delta \Psi; \mathbf{r}; \dot{\mathbf{r}}] \) (we did not estimate the biases in the IMU because most of them are not observable), where \( \delta \Psi \) represents the errors in the orientation, such that:

\[
C^o_n(k) = \exp([\delta \Psi(k) \times]) \cdot C^o_n(k).
\]

The attitude or orientation is estimated using (3). The stance phase of a normal walk is detected evaluating the ranges of the norm of the acceleration and turn rate, in [26] the author evaluates different techniques, concluding that the turn rate provides the best information about the phases of the walk. The state update uses (1) to (3). It is represented as:

\[
\dot{\mathbf{x}}(k) = f(\mathbf{x}(k - 1), \mathbf{a}^b(k - 1), \omega^b(k - 1)) + \mathbf{w},
\]

where \( \mathbf{w} \) is the process noise, such that \( \mathbf{Q} = E(\mathbf{w} \cdot \mathbf{w}^T) \), and \( f() \) is the propagation function based on (1), (2) and \( \Delta \Psi(k) = \Delta \Psi(k - 1) \). The estimate of the covariance matrix \( \hat{\mathbf{P}} \) is propagated using:

\[
\hat{\mathbf{P}}(k) = \Phi(k - 1) \cdot \mathbf{P}(k - 1) \cdot \Phi^T(k - 1) + \mathbf{Q},
\]
where \( \Phi \) is the Jacobian of the propagation function:

\[
\Phi = \begin{bmatrix}
I_3 & 0_3 & 0_3 \\
0_3 & I_3 & I_3 \cdot \Delta T \\
- [C^T \theta \times] \cdot \Delta T & 0_3 & I_3
\end{bmatrix},
\]

(8)

\( I_n \) and \( O_n \) are \( n \) by \( n \) identity and zero matrices, respectively.

When a stance is detected, a pseudo measurement of a zero velocity observation is implemented as a state correction. This measurement can be modeled as:

\[
m = H \cdot x + v,
\]

(9)

where the measurement is \( m = [0, 0, 0]^T \) (zero velocity), the observation matrix is \( H = [0_3, 0_3, I_3] \) and the measurement noise is \( v \), such that \( R = E(v \cdot v^T) \) (we use a value of \( R = 0.02(m/s)^2 \cdot I_3 \)).

Using this information, the states and covariance are corrected as:

\[
x(k) = \dot{x}(k) + K \cdot (m - H \cdot \dot{x}(k)),
\]

(10)

\[
P(k) = (I - K \cdot H) \cdot \dot{P}(k),
\]

(11)

where the Kalman gain \( K \) can be calculated as:

\[
K = \dot{P}(k) \cdot H^T \cdot (H \cdot \dot{P}(k) \cdot H^T + R)^{-1}.
\]

Finally, the attitude is corrected using (5), and therefore the attitude errors are reset (in the next state update we assume \( \delta \Psi = [0, 0, 0]^T \)). INS based PDR offers a good relative positioning but requires an initial position/orientation and will accumulate errors with distance. This is usually handled using range or angle measurements to provide a robust system, but their non-linearities might affect the estimation process.

B. Upper level, particle filter

The inclusion of an upper level particle filter offers the flexibility of including non-linear measurements and does not require the initial states, but carries the cost of requiring more computing time than an EKF. Several authors like [13], [14] and [15] proposed to update the filter only after each step, tracking only the non-observable states of the lower filter, the position \( r \) and orientation \( \theta \) of the person in the navigation frame (the superindex \( n \) is eliminated for an easier representation). In our model, the states \( X^{(i)} \) for the \( i \)-th particle are represented as \( X^{(i)} = [r^{(i)}, \dot{r}^{(i)}, \dot{r}^{(i)}, \dot{\psi}^{(i)}] \) and they are propagated from the stance \( j \) to the stance \( j + 1 \) with the information from the step \( j \). Using the adaptive error model from [12], the particles are propagated as:

\[
X^{(i)}(j + 1) = X^{(i)}(j) + C_z(\theta^{(i)}(j)) \begin{bmatrix}
0_{3 \times 1} \\
1
\end{bmatrix} \left( \Delta X^{(i)}(j) + \eta_{\Delta x} \right),
\]

(13)

where The step displacement is defined as \( \Delta X(j) = [\Delta r_x(j), \Delta r_y(j), \Delta r_z(j), \Delta \theta(j)]^T \), measured in the navigation frame rotated according to the yaw of the stance \( j \). \( \eta_{\Delta x} \) is the error model of the step displacement generated using a Gaussian distribution with a covariance \( P_{\text{step}} \), adjusted according to the orientation of the step \( \theta_s \), step length \( SL \), swing time \( t_{sw} \) and total step time \( t_{st} \). The model as proposed in [12] is:

\[
\sqrt{P_{\text{step}}(\theta_s, SL, t_{sw}, t_{st})} = C_z(\theta_s)(A \cdot SL + B \cdot t_{sw} + C \cdot \sqrt{t_{st}} + D),
\]

(14)

obtained from a linear approximation to the Cholesky decompositions of the observed covariances (\( \sqrt{P} \)) in a Monte Carlo evaluation (the values of the Matrices are presented in the Appendix, and the rotation matrix \( C_z(\theta^{(i)}(j)) \) is a rotation over the \( Z \) axis, i. e.:

\[
C_z(\theta^{(i)}(j)) = \begin{bmatrix}
\cos(\theta^{(i)}(j)) & -\sin(\theta^{(i)}(j)) & 0 \\
\sin(\theta^{(i)}(j)) & \cos(\theta^{(i)}(j)) & 0 \\
0 & 0 & 1
\end{bmatrix}.
\]

(15)

We have observed that this model does not take into account the non-observable bias in the vertical component of the turn rate, therefore in Section III we propose an extended motion model as a way to improve the particle displacements.
Once the estimation have been propagated, any measurement \( z(j) \) updates the weights of the particles \( (w^{(i)}) \), according to:
\[
 w^{(i)}(j) = \frac{w^{(i)}(j-1) \cdot p(z(j)|X^{(i)}(j))}{p(z(j))},
\]
where \( p(z(j)) = \sum_i p(z(j)|X^{(i)}(j)) \) acts as a normalization factor. Using this equation we propose to include the following information:

1) RF measurements: A RF signal transmitted in free space is altered according to several expected attenuations and delays that can be used to determine the distance or angle of the source of the signal [2]. The most common measures used to determine the distance or angle between emitter and receiver are the time of arrival (TOA), the time difference of arrival (TDOA), the received signal strength (RSS) and the angle of arrival (AOA). Those distances, angles and signal strength measurements can be translated to positions using several methods ranging from trilateration/triangulation, error function minimizations, fingerprinting and Bayesian methods [27].

Any measurement \( z_m \) from a beacon \( m \) can be represented as a function of the position of the object to locate \( r = [r_x, r_y, r_z]^T \) as:
\[
 z_m = h_m(r) + \eta_z,
\]
where \( h_m(r) \) is the observation function from the \( m \)-th beacon and \( \eta_z \) represents the error of the measurement. Time and angle measurements are uncommon in typical RF beacon installations on a building (WiFi, Bluetooth, RFID, etc) and the only measurement available in most RF devices is the RSS of the signal.

The measurement \( RSS_m \) in dB from the beacon \( m \) is usually estimated in indoor environments with a generic path loss model [28] as:
\[
 RSS_m = \alpha - 10 \cdot \beta \cdot \log_{10}(d_m/d_0) + \eta_{RSS},
\]
where \( \alpha \) is the received RSS at a reference distance \( d_0 \) (usually 1 m), \( \beta \) is the path loss exponent (2 in free space), \( d_m = \sqrt{|r - r_m|^2} \) is the distance to the beacon at \( r_m \) and \( \eta_{RSS} \) is the measurement noise with variance \( \sigma_{RSS}^2 \). Indoors, the different multipaths and obstructions of the signals can amplify or attenuate the signals, affecting the path loss model. This is usually modeled calibrating the values of \( \alpha \) and \( \beta \) for the area of study [29].

Once a large enough set of RSS measurements have been determined, the position can be computed by converting the trilateration equations into linear form [30], or by direct minimization of a cost function constructed using the path loss law [2], [17]. These methods require a simple calibration procedure to estimate the path loss model for at least one emitter-receiver pair, but is affected by the obstacles and building characteristics in the real propagation paths. Another commonly used technique is fingerprinting [31], where the received signals are compared to previously recorded measurements in dense grids of points and the position is obtained using methods that range from minimization of the Euclidean distances, Neural Networks, Bayesian modeling, etc. This method offers a lower error, but it is severely affected by changes in the environment and requires an extensive calibration.

RSS measurements have been widely used to provide absolute positions to inertial estimations, usually, fusing both estimations on a Bayesian filter. Using an EKF, the author in [32] implements a loose coupling between ZUPT based PDR and the position obtained from WiFi RSS and the position obtained from the navigation sensor. While in [33] the author uses a tight coupling of the linearized measurement model of the RSS path loss of RFID tags to improve a ZUPT based PDR. Using a particle filter, [17], [34], [35] implemented loose couplings of PDR based on step length and orientation change, with the position obtained from WiFi RSS, while [19] uses a tight coupling of ZUPT based PDR and the fingerprints resemblance to the real WiFi RSS measurements. In this paper we will use a tight coupling of the RSS measurements with a generic path loss model, updating the particle weights according to (16) and:
\[
 p(z(j)|X^{(i)}(j)) = \frac{1}{\sigma_{RSS} \sqrt{2\pi}} \exp \left( -\frac{(\alpha - 10 \cdot \beta \log_{10}(\hat{d}_m/d_0) - RSS_m)^2}{2\sigma_{RSS}^2} \right),
\]
where \( \hat{d}_m \) is calculated using \( X^{(i)}(j) \), an estimate of the position according to the time of reception of the of the RSS measurement [12].

2) Map Matching: The map layout of a building is commonly used to depict the obtained trajectories using any indoor positioning system, but overlaying any obtained trajectory on the map can allow us to get a better idea of the real path based on the simple principle that an estimate should not pass through a wall.

The localization of vehicles using GNSS systems motivated the first attempts to use the map information to enhance the positioning, providing multiple methods to include the information of the map in the estimation. This method can significantly improve the accuracy of an estimation, but can only be used if the person follows a particular expected path and a correct map of the area is available. Following [36] the algorithms can be classified as:

**Geometrical analysis:** The position estimate is approximated to known paths, fitting to the closest point, curve or polygon.

**Topological analysis:** The estimates are approximated to the closest known points but the displacements between them are limited according to their connectivity.

**Probabilistic map-matching algorithms:** The method defines an elliptical confidence region around a position fix obtained from the navigation sensor. The error region is superimposed on the possible paths of the user to identify the path segment in which the user is walking. If the error region contains more than one segment, the candidates are evaluated using a heading, connectivity and closeness criteria.

**Advance map-matching algorithms:** Any method that treats the map matching using sensor fusion techniques to include multiple measurements from different sensors.
Most methods use non-parametrical filters due to the non-linear, non-Gaussian measurements involved. The most common approaches track the position and orientation of the person, using the step length and orientation change obtained from PDR to propagate the estimations, and updating the probabilities according to the wall crossings detected or additional measurements received.

Due to the fact that the first three classes alter the walking pattern of the person, focus on single hypothesis or requires an initial estimate of the position of the person, we will center our research in the advanced map-matching algorithms implemented in particle filters.

In [13], [16], [37], each author independently proposed the use of a particle filter to estimate the position and orientation of a person. They used the step length and orientation change (obtained from PDR) to propagate the particles, and updated the weights of the particles if they crossed any wall (assign a weight of 0) or not (weight remains the same). Finally during the resampling the hypotheses with a weight of 0 are eliminated. In [38] the author is also able to use map matching implementing this approach to guide a person in multiple floor scenarios using a simplified map.

The information of the map allows the filter to focus on the particles that pass in the correct path. The number of operations required for wall crossing detections usually grows with the number of particles and walls. The methods proposed in [16], [37], [38] check the wall crossing of each particle to all the walls in the building, something only feasible when the number of walls and particles is small.

A less detailed map reduces the time required for map matching, but also reduces the accuracy of the method and might affect the estimated walking pattern of the person. A less accurate map can generate a particle deprivation due to crossings with badly mapped walls. As a way to avoid the elimination of the particles it might be necessary to reduce the weight of the particles that pass a wall instead of killing them (set weight to 0), however this might generate hypotheses in wrong positions that consume a significant part of the computing power. In this paper we will assume that the map is accurate enough (our map errors are lower than 5 cm) and to accelerate the process, we propose in Section IV-A a vectorized version of the wall crossing detection. Future papers will deal with the inaccuracy of the map.

The author in [13] also proposed to include in a new state, the room (or polygon) of the particle, and to evaluate only the walls in the rooms the particle is in. This allows to only check for approximately 4 to 20 segments for each particle, accelerating the computation significantly (it was implemented in Java in a Linux machine). The algorithm differentiates between walls and doors (or crossable segments with a connection to another room) and checks for interceptions between the particle’s movement $\vec{AB}$ (from $\mathbf{r}^{(i)}(j)$ to $\mathbf{r}^{(i)}(j + 1)$) and the walls and doors of the room of the particle. The three possible outcomes are:

1) The movement does not intercept any wall nor door. The particle keeps its weight and remain in the same room.
2) The movement intercepts a wall. The particle is assigned a weight of 0.
3) The particles first interception is with a door. The particle changes its room to the one connected through that door and $\vec{AB}$ is changed to $\vec{IB}$, where $I$ is the interception between $\vec{AB}$ and the door. The wall checking must be executed again with the new movement and with the walls and doors of the new room.

The process is repeated until the outcome is one of the first two options.

This algorithm needs to check each particle for its crossings, but most of the time there are more than one particle in the same room with similar conditions. As a way to accelerate this process with matrix operations, in Section IV-B, we propose to group the particle within the same room and perform the wall crossing detection efficiently using vectorized operations.

### III. EXTENDED MOTION MODEL/PREDICTION

As previously discussed the behavior of PDR can be modeled propagating particles using the step displacements between stances and (13). We propose to evaluate how accurate this assumption is, using a Monte Carlo approach with the synthetic foot-mounted IMU signal from [39] (10 closed loops, approximately 1000 m of total traveled distance) and recorded IMU noise. We will compare the evolution of the root mean squared error (rmse) in the horizontal position obtained with PDR (the real error growth of the estimation, obtained from the Monte Carlo experiment) and the dispersion of the particle distributions obtained using the particle displacement approach previously proposed (our model of the error growth, obtained from the covariance of the particles).

In figure 2 we can observe in red the evolution of the real error growth with time for a 9 states INS-EKF PDR approach, and in green the model of the error growth with the originally proposed PF with step displacements. As previously observed in [12], the rmse in the PF grows as a random walk, this is due to the random noise model used, but as we can see, the real error (the error of the original EKF based PDR) grows approximately linearly with time, this means that the particles

![Fig. 2. Evolution of the Root Mean Squared Error (in the horizontal position) with time for several methods, obtained using a Monte Carlo analysis of 1000 simulations.](image-url)
in the PF after some time (around 400 s) will have a lower covariance than expected, underestimating the real error of PDR, something that might generate a particle deprivation. This problem is mainly due to the effect of the unobservable bias in the heading’s turn rate (vertical component of the gyroscope’s bias) that generates errors in the heading and position that grow linearly with time.

In [40], the author proposes to increase the error level of the components of the step error according to time, but that approach only affects the distribution of the particles, but it cannot correct the effect of the bias. We propose to include as an additional state $X_5$, the derivative of the heading (heading’s turn rate bias $\delta \theta$), modeling the bias of the gyroscope [41] as a random walk in the bias. The new state vector will be $X^{(i)} = [r_x^{(i)}, r_y^{(i)}, r_z^{(i)}, \theta^{(i)}, \delta \theta^{(i)}]^T$, and the particles can be propagated as:

$$X^{(i)}(j+1) = F \cdot X^{(i)}(j) + U \cdot (\Delta X^{(i)}(j) + \eta_{\Delta x}) + N \cdot \eta_{\delta \theta}, \tag{20}$$

where the transition matrix $F$ is:

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & t_{\text{step}} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{21}$$

the time from the previous step is $t_{\text{step}}$, the displacement matrix $U$ is:

$$U = \begin{bmatrix} C_2(\theta^{(i)}(j)) & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \\ 0_{1 \times 3} & 0 \end{bmatrix}, \tag{22}$$

the noise matrix is $N = [0, 0, 0, 0, 1]^T$ (we do not include process noise in the first four states, as the step error model already does it), and $\eta_{\delta \theta}$ is the bias error model, that will be dependent on the time since the last step and a bias rate random walk coefficient $K$ (the standard deviation in the change of the bias after 1 second, it can be obtained using an Allan Variance analysis, in our case $1 \cdot 10^{-3} \cdot \sqrt{s^2}$). The standard deviation of the change in the gyroscope bias with respect to the previous stance is:

$$\sigma_{\delta \theta} = K \cdot \sqrt{t_{\text{step}}} \tag{23}.$$

With the proposed 5 states model, we repeated the simulations of the evolution of the rmse obtaining the error growth displayed in the blue line of the figure 2. The new model has an error growth closer to the real PDR error growth, providing a better step propagation model thanks to the inclusion of the heading turn rate bias modeling and avoiding an underestimation of the real error growth.

IV. MAP MATCHING

After the propagation of the particles, the hypotheses can be adapted to the map of the building discarding those particles that intercept a wall and rewarding those that survive. In [13] the author shows that PDR with map matching based in wall crossing detection is able to position a person indoor if the trajectory is singular enough, but requires a significant amount of particles to converge. It also propose the use of the WiFi RSS to lower the amount of particles required and remove symmetrical trajectories, but the calibration of the RSS measurements requires a significant amount of time and might fail if the environment is modified. We propose to optimize the wall crossing detections using vectorized math and group the operations according to the room the particles are in, minimizing the computation time and allowing parallel computing.

A. Vectorized wall crossing detection

As previously discussed in Section II-B2 the inclusion of the map layout information is possible using a wall crossing detection, but the typical number of walls in a building and the significant amount of particles used in a PF require that the wall crossing detection must be executed as matrix operations for a faster computing.

The basic wall detection used in our case is based on the line segment intersection detection shown in figure 3, proposed in [42] and adapted to the two dimensional case. It is based on obtaining the coefficients $\gamma_{i,s}$ and $\lambda_{i,s}$ for the line segments $\mathbf{p}_1$ and $\mathbf{p}_2$, and the interception point $I_{i,s}$. The points and coefficients are related according to:

$$I_{i,s} = \mathbf{p}_1 + \gamma_{i,s} (\mathbf{p}_2 - \mathbf{p}_1) = I_1 + \lambda_{i,s} (I_2 - I_1), \tag{24}$$

and if both $\gamma_{i,s}$ and $\lambda_{i,s}$ are greater than zero and lower than one, the segments intercept. The coefficients can be obtained as:

$$\gamma_{i,s} = \frac{(I_2 - I_1)_y (I_1 - I_2)_y - (I_1 - I_2)_x (I_1 - I_2)_y (I_2 - I_1)_y - (I_2 - I_1)_x (I_1 - I_2)_y)}{(I_1 - I_2)_y (p_2 - p_1)_y - (I_1 - I_2)_x (p_2 - p_1)_y), \tag{25}$$

and

$$\lambda_{i,s} = \frac{(p_2 - p_1)_x (I_1 - I_2)_y - (p_2 - I_1)_y (I_1 - I_2)_x (p_2 - p_1)_y - (p_2 - I_1)_y (I_1 - I_2)_x (p_2 - p_1)_y)}{(I_1 - I_2)_y (p_2 - p_1)_y - (I_1 - I_2)_x (p_2 - p_1)_y). \tag{26}$$

Associating the points to our Map matching scheme, the line segments are defined from $\mathbf{p}_1 = \mathbf{r}^{(i)}(j)$ to $\mathbf{p}_2 = \mathbf{r}^{(i)}(j + 1)$ (the step displacement) and from $I_1$ to $I_2$ (line segment from the wall $s$). We tested a vectorized implementation of this algorithm [43], but it was not fast enough to execute 100000
We propose to check all the wall crossings as matrix operations. For an easier sorting of the particles according to their room we will use an additional state $X_6 = g$ as proposed in [13], that will specify which room the particle is in. The new state vector is $X^{(r)} = [x, y, v_X, \theta, \dot{\theta}, g]^T$. If we want to evaluate the particles in the room $g_0$, we define a subset of particles $V = \{v_1, \ldots, v_{N_v}\}$, such that $X^{(v_p)} = g_0$, with $p = 1, \ldots, N_v$. The subsets associated with the differences $(p_2 - p_1)x$ and $(p_2 - p_1)y$, for the particles $V$, can be represented as the rows

$$
\Delta P_x = \begin{bmatrix} r^{1\times} + 1 \end{bmatrix} - r^{1\times} \\
\vdots \\
\Delta P_x \\
\Delta P_y = \begin{bmatrix} r^{1\times} + 1 \end{bmatrix} - r^{1\times} \\
\vdots \\
\Delta P_y \\
\Delta L_x = \begin{bmatrix} \Delta L \end{bmatrix} \\
\Delta L_y = \begin{bmatrix} \Delta L \end{bmatrix}
$$

where the element $\text{Cross}_s$ indicates if a wall crossing was detected between the particle $v$ and the line segment $s$. Finally, the values of the column $v$ of $\Gamma$ can be used to detect the order of the intersections of the particle $v$ with the different lines. The first interception is obtained looking for the lowest value of the column $v$ of $\Gamma$ among the ones with $\text{Cross}_s = 1$. These operations can be executed efficiently using Linear Algebra Package (LAPACK) libraries and efficient element-wise operations (both implemented in MATLAB operations) accelerating the processing time of the wall crossing detection.

### B. Room groups and wall structures

As previously proposed by [13], the particles only need to check the crossing with a limited number of walls, and we propose to group the particles according to their room to optimize the use of the vectorized operations of Section IV-A and to avoid having to access the wall segments with every particle. With the particle groups, we will like to access the wall segments in a hierarchical structure (RoomStructure), based on an array of rooms as observed in figure 4. Each room element will include a room number, the line segments of the room (walls, doors, etc.), the connections to other rooms and a subset of room elements for each connecting room with a reduced number of segments (not all the walls of the connecting room are at reach from the original room), therefore this structure will follow a tree data type. The segments of the room is based on an array of line segments $l^r$, where each is defined by 2 points, $l^r_1$ and $l^r_2$. As a rule the concatenation of the segments of a room should generate a closed polygon. The lines defining a room can be separated into non crossable segments (usually walls or any obstruction in the walking path) and crossable segments (openable doors or any walkable division between the declared rooms), and we will define this as a connections array in the same order of the line segments, the non crossable elements have a value of 0, while the crossable elements value is the room it connects to.

Each connectable segment will have a subRoom, a (reduced) room element, following the structure observed in figure 4 to indicate the wall crossings to check when the particle passes by that crossable segment. This subset will only include the reachable line segments of the connecting room, from the connecting segment of the original room and with a pre defined reach distance. We have observed that a typical distance between two consecutive stances of the same foot in PDR is approximately 1.2 m, assuming two consecutive long steps with a failed stance detection in the middle we will define a maximum reach of 3 m.

The reachable segments are obtained offline drawing multiple maximum reach steps from each connecting segments of a room (worst case scenario), varying the position in the segment and the orientation of the displacement, and observing the first crossings of the simulated steps with the segments of the next room. The segments array will not include the crossable segments already passed (the connecting door or room division). If a connecting segment in the new room is reachable from the original room, the subRoom element will
have a subRoom element with the segments reachable from
the original connecting segment. Given the fact the step
length is limited we have seen using the above mentioned method,
that a room element may contain up to 2 levels of additional
subRooms.

With these operations and the room structure in mind we
propose the following pseudo-code to verify the wall crossing
in a building:

```plaintext
newUsedRooms=void;
FOR room1= all the roomsWithParticles
    subset1=particles with (X_6==room1);
    selectedRoom1=RoomStructure(room1);
    segments1=selectedRoom1.segments;
    connections1=selectedRoom1.connections;
    crossings1=EvaluateCrossings(subset1,segments1);
    IF at least a particle did not cross a segment
        newUsedRooms ← room1;
    ENDIF
particlesToKill=particles crossing a wall;
set X_6 of particlesToKill to 0;
FOR room2=possible connections in connections1
    subset2=particles that crossed to room2
    selectedRoom2=selectedRoom1.subRoom(room2);
    segments2=selectedRoom2.segments;
    connections2=selectedRoom2.connections;
    crossings2=EvaluateCrossings(subset2,segments2);
    IF at least a particle did not cross a new segment
        newUsedRooms ← room2;
        set X_6 of particles not crossing to room2
    ENDIF
particlesToKill=particles crossing a wall;
set X_6 of particlesToKill to 0;
FOR room3=possible connections in connections2
    subset3=particles that crossed to room3
    selectedRoom3=selectedRoom2.subRoom(room3);
    segments3=selectedRoom3.segments;
    connections3=selectedRoom3.connections;
    crossings3=EvaluateCrossings(subset3,segments3);
    IF at least a particle did not cross a segment
        newUsedRooms ← room3;
        set X_6 of particles not crossing to room3
    ENDIF
particlesToKill=particles crossing a wall;
set X_6 of particlesToKill to 0;
ENDFOR
ENDFOR
roomsWithParticles=newUsedRooms;
set weight of particles with (X_6==0) to 0;
```

The code is defined for a 3 level room crossing (go through
3 rooms in 1 step), but can be easily modified to include more
levels.

Map matching offers an important source of information
and an excellent way to improve the estimations, especially
in narrow corridors, but it fails when the building presents
symmetries that generates multiple hypotheses. Map matching
also requires a significant amount of particles (and therefore
processing) to initialize the position and orientation, but as
the author in [13] proposes, the number can be lowered
using additional sensors like magnetometers, altimeters or any
measurement related to a beacon in a known position.

V. EXPERIMENTS AND ANALYSIS

In this Section we will discuss the proposed experiments
and their results. The proposed algorithms were evaluated in
the Building A of the Centre for Automation and Robotics. It
is a non symmetrical structure with offices of different sizes
and can be seen in figure 5. The offices are located in a
66 m × 42.5 m area with 90 RFID tags (M100 from RF Code)
installed in the building, transmitting their ID every 1 sec in
the 433 MHz Band. This tag was selected due to their low
price and high range. In all the tests we carried a RFID reader
(M220 from RFCode) at the waist level and a MTi IMU from
XSens mounted in the instep of the right foot of the person,
recording at 100 samples/sec the accelerations and turn rates
used for Pedestrian Dead Reckoning.

As a way to test the different improvements proposed in this
paper we will show four experiments. The first experiment
is designed to validate the map matching algorithm and to
prove that it is possible to locate a person when the initial
position and orientation is unknown. The second experiment
deals with the effect of the inclusion of the turn rate bias state
in the particles while the third one will treat the use of RSS measurement in the positioning. Finally the execution time of each algorithms will be analyzed.

A. Positioning with PDR and Map Matching without initial position/orientation

The map matching algorithm based on the PDR step displacements was tested in our building, recording a counterclockwise walk, entering several offices and returning to the original starting point after two laps. The particles were initially distributed uniformly around the bounding box area with an uniformly distributed orientation between $-\pi$ and $\pi$ rad. The followed path is observed in figure 5.

It is clear that many particle’s paths outside of the building will not intercept any wall, therefore those particles will not be eliminated and will consume a part of the processing time. As a way to avoid those hypotheses several strategies can be implemented, but as in this first experiment we only want to validate the map matching algorithm, we will use a non crossable bounding box around the building.

As we can observe in figure 5, for this particular path, the algorithm is able to estimate the position of a person without initial position nor orientation after 100 foot steps. The number of steps required to obtain a single estimate of the position (one particle cloud) will vary according to the walking path and the symmetries of the building, but as we mentioned before this is just an example of the capabilities of the algorithm. Map matching offers a positioning algorithm capable of obtaining the initial position and orientation of a PDR estimation when the trajectory and the building are non symmetrical enough, but as we will see in Section V-C, a robust positioning will require additional measurements to avoid multiple hypotheses and to accelerate the convergence.

B. The effect of the turn rate’s bias in the state propagation

The turn rate’s bias has a long term effect in the propagation of the particles, in tight corridors the distributions tend to eliminate the biases effects and correct the position of the estimation without a problem, but if the person enters an area without wall correction (large halls, external gardens or paths, etc.) or just stays during a large period of time wandering in the same room (e.g. visiting an art gallery), the lack of lateral corrections will alter the orientation and position estimation. As a way to evaluate the effect of including the turn rate’s bias in the state estimation, we tested the algorithm in a path with indoor and outdoor trajectories, the path can be observed in figure 6. The particles were initialized uniformly in a known office but with an unknown orientation, but at the moment the person exits the building, the particles have converged to a single cloud and during the outside walk the estimation will depend only in the state propagation without map corrections.

The evolution of the clouds in different points of the external path can be observed in figure 6. In the first external path (figure 6.a and b) the evolution seems similar (slightly bigger in the estimation with bias), but when the path reenters the building many particles are eliminated due to crossings with walls, and in the new proposed model, it is able to correctly estimate the bias because only the particles with the correct bias will survive. In the following external loops, the estimations with bias modeling are able to correct the effect induced by the not observable turn rate’s biases and the particle clouds are centered in the correct positions. In the case without bias estimation, the particle clouds are affected by the non observable turn rate’s biases deviating the cloud in the same way that the first loop and, if the external loops prolong too much, this might generate a particle deprivation when reentering the building through a narrow door.

C. PDR and Map matching with RSS measurements

As previously mentioned by Woodman in [13] the number of particles required for positioning will grow with the size of the building and a way to limit that amount is the use of RF signals. We propose to use the RFID measurements from 24 beacons in our building (we typically receive signals from 75% of the tags closer than 25 m from the receiver), to limit the initial distribution of the particles and to avoid the symmetries (and outside paths) in the walking paths. The initial particle distribution was estimated using the RSS measurements before the first step, after that, the rest of the RSS measurements were included in the particle correction phase of the filter, using the path loss model from [33]. We used a fixed amount of particles.

For this experiments the pedestrian made a L shaped walking path as observed in the figure 7. If only map matching were used, this shape might generate an additional positioning hypothesis in the bounded area defined for the first experiment (light blue line in the figure), but if the outer area is not bounded, there are plenty of additional hypotheses. As observed in the figure, distributing the initial particles centered in the position obtained from the RSS measurements, the particles already eliminates the false hypothesis and centers the computing in the real position. Figure 8 shows the evolution of the number of hypotheses obtained using a clustering algorithm on the position and orientation of the particles, it can be seen that the use of RSS measurements in Map Matching, greatly accelerates the convergence of the filter from 80 foot steps (to obtain 2 hypotheses) to 12 foot steps (to obtain a single hypothesis).

The use of map matching and RSS measurements significantly lowers the positioning error obtained with only one of the previous methods as it can be observed in the red line of the cumulative distribution function in figure 9 (less than 0.75 m of error, during 90% of the path) and table I (the method has a mean error of 0.51 m). Without the use of RSS measurements the estimation has at least 2 hypotheses (after

<table>
<thead>
<tr>
<th>Algorithm \ error</th>
<th>Average</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>only RSS</td>
<td>1.12 m</td>
<td>0.63 m</td>
</tr>
<tr>
<td>only Map matching, best hypothesis</td>
<td>1.97 m</td>
<td>3.92 m</td>
</tr>
<tr>
<td>Map matching and RSS</td>
<td>0.51 m</td>
<td>0.42 m</td>
</tr>
</tbody>
</table>

TABLE I MEAN AND STANDARD DEVIATION OF THE ERROR USING MAP MATCHING AND/OR RSS MEASUREMENTS FOR THE TRAJECTORY OBSERVED IN FIGURE 7.
Fig. 5. Estimation of the position using PDR and Map Matching with 100000 particles. Axis in meters. For an easier view only 300 randomly selected particles are shown.

Fig. 6. Evolution of the particle clouds after different number of steps, comparing the step propagation with (right, green) and without (left, red) turn bias estimation. Using a particle filter with 100000 particles. a. and b. First external loop, the bias affects the particles deviating the estimation. c. and d. e. and f. Second and third external loops, the estimation with bias is able to correct the effect while the estimation without deviates from the correct path. Axis in meters. For an easier view only 300 randomly selected particles are shown.
Fig. 7. Evolution of the particle clouds after different number of steps using RSS measurements and map matching using 100000 particles. The magenta circles centered in the beacons show the distance associated with the received RSS measurements, while the relative probability of each particle is represented with the color (according to the color scale) and size. Axis are in meters. For an easier view only 1000 randomly selected particles are shown.

Fig. 8. Evolution of number of hypotheses using map matching with and without RSS measurements. The number of hypotheses is obtained using a clustering algorithm.

Fig. 9. Cumulative distribution function of the positioning error obtained with the particle filter using map matching and/or RSS measurements. The method Map matching uses an externally selected best cluster for comparison, using the 2º possible hypothesis generates errors greater than 30 m and without external measurements the real position can not be distinguished.
80 steps), one of those hypotheses has an error comparable to the method using map matching and RSS measurements, but without external measurements it is not possible to identify the correct position, additionally, due to the slow convergence, approximately 15% of the points have more than 2.5 m of error and as observed in the table I this increases the mean error level to 1.97 m. Using only RSS measurements it is possible to locate the user with a fast convergence and without multiple hypotheses, but with a higher error (less than 1.9 m of error, during 90% of the path, mean error of 1.12 m).

D. Computing time

We analyzed the time required to execute each algorithm in MATLAB, using a PC with an Intel Core 2 processor (2.83 GHz) in Windows 7. Although the processor has several cores, the functions implemented uses only 1 core at a time and we expect it to improve if better parallel processing libraries are used. The evaluated algorithms were:

- **rMM**: room based Map Matching as proposed by Woodman in [13] evaluating only the segments of the particle’s room. The implementation was not vectorized (in a loop) and we expect it to be slower than any other method due to the way MATLAB handles the loops (in [13] the programming language used is fast executing loops, but we needed to use MATLAB). We used a fixed amount of particles.

- **mMM**: matrix based Map Matching, detecting the wall crossings with all the segments of the building using the vectorized operations (33) to (35).

- **eMM**: efficient Map Matching, evaluating only the segments of the particle’s room, but grouping the particles according to the room and data structure of figure 4 and using the vectorized operations (33) to (35).

- **RSS**: using RSS measurements with a path loss model in the correction phase as proposed in [18].

The algorithms were tested in the 732.34 seconds signal from figure 5, with the signals from 24 RFID beacons distributed along the whole building, when using RSS measurements.

Table II shows the execution time of the previous algorithms. It shows that the use of matrix operations significantly lowers the time required to use Map Matching compared with a non vectorized version. The use of room based grouping, in addition to the matrix operations (the efficient Map Matching) is the fastest method, over 100 times faster than the implementation in a for loop and 13 times faster than using the walls in matrix operations. Extrapolating the times, it might be possible to execute up to approximately 300,000 particles at the limit of real time processing (in our 2.83 GHz processor), equivalent to the maximum of 274,000 particles obtained in [13] with a 2.6 GHz processor in Java using Linux. Additionally, due to the use of elementwise operations, it can be easily implemented in parallel processors, increasing the efficiency of the method.

The inclusion of RSS measurements also decreases the required number of particles for convergence, mainly because it helps to focus the initial particles in the area of interest and adjust their weights for a better resampling. With a small number of particles (1,000), using map matching in the estimation with RSS measurements might generate a particle deprivation as indicated by the superindex 1, this is mainly because RSS estimations usually have a position dependant error that can make small particle clouds to collide with the walls. The best way to avoid this is to increment the initial RSS measurement variance or increase the number of particles.

VI. CONCLUSION

In this paper we have presented an indoor positioning system based on foot-mounted Pedestrian Dead Reckoning with Map Matching and RSS measurements. On that scheme we have proposed the following improvements over the state of the art: an improved step motion model that includes the estimation of the turn rate bias and the correction of its effect on the estimation, a vectorized version of the wall crossing detection algorithm capable of quickly determining the first wall crossing of a step, and a room grouping algorithm to allow vectorized wall crossing detections with a minimum number of walls to evaluate.

We observed that the typical motion model of particle filters used in Map Matching suffers from an orientation deprivation due to unmodeled states. We proposed the inclusion of a turn rate bias as a new state of the PF, to correctly model the increment of the orientation and positioning errors. The turn rate bias state is also able to correct the positioning especially when the person passes large periods of time without orientation corrections, i.e. large hall rooms, external areas, etc.

We proposed and explained a wall crossing detection algorithm based on matrix operations that accelerates the time required to detect the intersections of groups of displacements over groups of wall segments. This kind of matrix operations can be fast in a high level programming language, and offer a significant advantage over non vectorized implementation. We have observed that the matrix operation is over 100 times faster than our loop implementation (in MATLAB) with the advantage of working in a high level language, allowing future implementations in GPUs, multiple cores, etc.

As a way to improve the wall crossings detection comparing only with the reachable walls [13] we proposed a particle grouping scheme according to the room the particle is in. This scheme allows the use of vectorized operations (not only wall

<table>
<thead>
<tr>
<th>Algorithm \ particles</th>
<th>100,000</th>
<th>10,000</th>
<th>1,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>only RSS</td>
<td>259.45 s</td>
<td>28.36 s</td>
<td>14.33 s</td>
</tr>
<tr>
<td>rMM</td>
<td>362.26 s</td>
<td>3034.76 s</td>
<td>does not converge</td>
</tr>
<tr>
<td>rMM + RSS</td>
<td>30561.56 s</td>
<td>3102.59 s</td>
<td>520.90 s</td>
</tr>
<tr>
<td>mMM</td>
<td>1400.57 s</td>
<td>447.76 s</td>
<td>does not converge</td>
</tr>
<tr>
<td>mMM + RSS</td>
<td>3564.00 s</td>
<td>360.27 s</td>
<td>3400.57 s</td>
</tr>
<tr>
<td>eMM</td>
<td>246.73 s</td>
<td>35.64 s</td>
<td>does not converge</td>
</tr>
<tr>
<td>eMM + RSS</td>
<td>446.49 s</td>
<td>52.59 s</td>
<td>19.45 s</td>
</tr>
</tbody>
</table>

1 The number of particles used might be close to the minimum required and the estimation can have a particle deprivation
crossing detection, but any position dependant measurement) to accelerate the process.

This paper also studies the inclusion of RSS measurements with a simple path loss, and remarks the effect on the number of particles required for positioning and the time requires for the filter to converge. These measurements do not require an additional calibration of the model. The inclusion of the Map Matching in the estimation with RSS measurements and PDR lower the error level from 1.9 m to 0.75 m 90% of the time, improving significantly the estimation.

Future works in this subject might include the use of multiple core calculation libraries, implementation in smartphones, the inclusion of additional measurements like GPS, magnetic fields [44], light sources matching [45], etc.

ACKNOWLEDGMENT
This work was supported by the LEMUR project (TIN2009-14114-C04-02), LORIS project (TIN2012-38080-C04-04), SMARTLOC project (CSIC-PIE Ref.201450E011) and the JAE PREDoc program.

REFERENCES
The Matrices used in the adaptive error model from (14)

\[
\sqrt{P_{\text{step}}(\theta_s, SL, t_{sw}, t_{st})} = C_z(\theta_s)(A \cdot SL + B \cdot t_{sw} + C \cdot \sqrt{t_{st}} + D)
\]

were:

\[
A = \begin{bmatrix}
-6.8 \cdot 10^{-5} & 0 & 0 & 0 \\
0 & 9.8 \cdot 10^{-4} & 0 & 0 \\
9.2 \cdot 10^{-5} & 0 & 2.2 \cdot 10^{-4} & 0 \\
0 & 2.3 \cdot 10^{-4} & 0 & -1.2 \cdot 10^{-3}
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
2.8 \cdot 10^{-3} & 0 & 0 & 0 \\
0 & 1.2 \cdot 10^{-3} & 0 & 0 \\
-7.3 \cdot 10^{-5} & 0 & 5.1 \cdot 10^{-5} & 0 \\
0 & -6.5 \cdot 10^{-4} & 0 & 3.4 \cdot 10^{-3}
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
-8.6 \cdot 10^{-4} & 0 & 0 & 0 \\
0 & -2.6 \cdot 10^{-4} & 0 & 0 \\
-1.5 \cdot 10^{-4} & 0 & 1.3 \cdot 10^{-5} & 0 \\
0 & 5.5 \cdot 10^{-4} & 0 & 5.5 \cdot 10^{-3}
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
8.6 \cdot 10^{-4} & 0 & 0 & 0 \\
0 & 7.1 \cdot 10^{-4} & 0 & 0 \\
2.7 \cdot 10^{-4} & 0 & 1.3 \cdot 10^{-4} & 0 \\
0 & 4.1 \cdot 10^{-5} & 0 & -3.6 \cdot 10^{-3}
\end{bmatrix}
\]

**APPENDIX**

**Francisco Zampella** received his Electronic Engineering and Master in Electronic engineering degree from the Universidad Simón Bolívar, Caracas, Venezuela, in 2008 and 2012, respectively and his Master in Electronic Systems degree from the Universidad de Alcalá, Spain in 2012. He worked within the Centro de Automation y Robotics (CAR) that belongs to CSIC (Spanish Council for Scientific Research), Madrid, Spain, holding a PhD scholarship and working toward his doctoral degree. His research expertise is in the areas of Indoor Positioning in GPS denied areas, Pedestrian Dead Reckoning, Ultra Wide Band Positioning, Bayesian Estimation, Sensor Fusion and Robotics.

**Antonio Ramón Jiménez Ruiz** was born in Santander, Spain, in 1968. He received the degree in physics and computer science and the Ph.D. degree in physics from the Universidad Complutense de Madrid, Spain, in 1991 and 1998, respectively. Since 1993 he has been within the Centro de Automation y Robotics (CAR) that belongs to CSIC (Spanish Council for Scientific Research), Madrid, Spain, where he holds a research position. His research expertise is in the areas of local positioning solutions for indoor/GPS-denied localization and navigation of persons and robots, signal processing, Bayesian estimation and inertial-ultrasonic-RFID sensor fusion. He has published more than 100 articles in journals and conference proceedings, and he is reviewer of many international journals and projects in the field.

**Fernando Seco Granja** was born in Madrid, Spain. He holds a degree in Physics (Universidad Complutense of Madrid, 1996) and a PhD in Physics (UNED, 2002), with a dissertation about the magnetostrictive generation of ultrasonic waves applied to a linear position sensor. Since 1997 he has been working at the Centre for Automation and Robotics (CAR) of the CSIC (Spanish Council for Scientific Research), in Arganda del Rey, Madrid, where he holds a research position. His main research interest lies in the design and development of indoor Local Positioning Systems (LPS), especially those based on ultrasonic and radiofrequency technologies, in signal processing for CDMA-based localization systems, and in Bayesian estimation.