

Entity Localization and Tracking: A Sensor Fusion-based Mechanism in WSNs

Stefano Tennina, Marco Valleta, Fortunato Santucci, Marco Di Renzo, Fabio Graziosi, Riccardo Minutolo

▶ **To cite this version:**

Stefano Tennina, Marco Valleta, Fortunato Santucci, Marco Di Renzo, Fabio Graziosi, et al.. Entity Localization and Tracking: A Sensor Fusion-based Mechanism in WSNs. IEEE MELT-11, Sep 2011, Banff, Canada. pp.1-6. hal-00663057

HAL Id: hal-00663057

https://centralesupelec.hal.science/hal-00663057

Submitted on 25 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Entity Localization and Tracking: A Sensor Fusion-based Mechanism in WSNs

Stefano Tennina*, Marco Valletta†, Fortunato Santucci†, Marco Di Renzo‡, Fabio Graziosi†, Riccardo Minutolo§
*CISTER Research Center, Polytechnic Institute of Porto (ISEP/IPP), Porto, Portugal
†Center of Excellence DEWS, University of L'Aquila, Italy
‡Laboratory of Signals and Systems (L2S), UMR 8506 CNRS - SUPELEC Univ Paris-Sud (Paris), France
§THALES ITALIA S.p.A., Chieti Scalo (CH), Italy
Emails: sota@isep.ipp.pt, marcovalletta@gmail.com, fortunato.santucci@univaq.it, marco.direnzo@lss.supelec.fr, fabio.graziosi@univaq.it, riccardo.minutolo@it.thalesgroup.com

***Abstract*—Knowing exactly where a mobile entity is and monitoring its trajectory in real-time has recently attracted a lot of interests from both academia and industrial communities, due to the large number of applications it enables; nevertheless, it is nowadays one of the most challenging problems from scientific and technological standpoints. In this work we propose a tracking system based on the fusion of position estimations provided by different sources, that are combined together to get a final estimation that aims at providing improved accuracy with respect to those generated by each system individually. In particular, exploiting the availability of a Wireless Sensor Network as an infrastructure, a mobile entity equipped with an inertial system first gets the position estimation using both a Kalman Filter and a fully distributed positioning algorithm (the Enhanced Steepest Descent, we recently proposed), then combines the results using the Simple Convex Combination algorithm. Simulation results clearly show good performance in terms of the final accuracy achieved. Finally, the proposed technique is validated against real data taken from an inertial sensor provided by THALES ITALIA.**

***Keywords*-WSN; Localization and Tracking; Inertial Systems; Sensor Fusion.**

I. Introduction

Wireless Sensor Networks (WSNs) are distributed networked embedded systems where each node combines sensing, computing, communication, and storage capabilities [1]. Due to their unprecedented design challenges and potentially large revenues, in recent years WSNs have witnessed a tremendous upsurge in interest and activities in both academia and industry [2]. In particular, they have become increasingly popular in military and civilian sectors, and have been proposed for a wide range of application domains, e.g., control and automation, logistics and transportation, environmental monitoring, health-care and surveillance.

In general, WSNs are required to possess self–organizing capabilities, so that little or no human intervention for network deployment and setup is required. A fundamental component of self–organization is the ability of sensor nodes to "sense" their location in space, i.e., determining where a given node is physically located in a network [3], [4]. In particular, node localization is a key enabling capability to support a rich set of geographically aware protocols for distributed and self–organizing WSNs [5], and for achieving context–awareness.

Several interesting applications are triggered by the possibility offered by the network to allow tracking of a mobile entity (ME). In this case it is important to differentiate among two distinct scenarios: (i) the ME is a node belonging to the WSN, and as a consequence it is equipped with the needed instruments, (i.e., collaborative localization) or (ii) the ME does not belong to the WSN, (i.e., not collaborative localization) as in "surveillance" applications [6] [7].

In this work we focus on the former scenario: a WSN where a limited number of *anchor* nodes knows *a-priori* its own position and a mobile node needs to first estimate its starting position then track its own movement in an environment. To do so, we address a combination (fusion) of positioning techniques such as our recently proposed distributed Enhanced Steepest Descent (ESD) algorithm [8] and an Inertial Navigation System (INS) composed by an Inertial Measurements Unit (IMU) such as accelerometer and gyroscope.

Although the idea of combining information from multiple types of positioning sensors to improve the final estimation is not new [9], the key to obtain an efficient position estimation, and, thus, a good tracking, is the implementation of valid (WSN-based) algorithms together with the use of sensors able to provide very accurate measurements.

The remainder of this paper is as follows. Section II overviews on the inertial navigation systems while Section III briefly summarizes the ESD algorithm. Both serve as background for Section IV where the fusion technique to integrate INS and ESD estimations is shown. Section V presents (i) the simulation model and the performance results of the proposed technique, and (ii) the preliminary experiments using a real IMU platform, provided to the Center of Excellence DEWS in L'Aquila (Italy) by Thales Italia. Finally, Section VI provides concluding remarks and future on-going work.

II. Inertial Navigation System

The fundamental idea of the Inertial (or *Newton*) Navigation derives from the physics: *"The second integral of acceleration is position"* [10]. Inertial navigation systems are based on the dead–reckoning and include a set of sensors, called IMU, and a navigation computer as shown in Fig. 1.

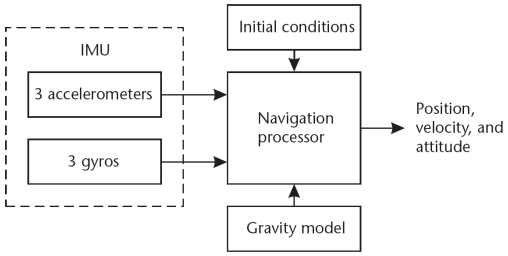


Fig. 1. Basic INS scheme.

The sensing part is often composed by an accelerometer (to measure changes in position) and a gyroscope (to maintain absolute angular references). Generally, there is at least one sensor for each of the three axes (mutually orthogonal to each other) to capture pitch (up and down), yaw (left and right) and roll (clockwise or counter–clockwise rotation) movements. The navigation processor gets IMU's measurements to output position, speed and attitude. In particular, the gyroscopes provide the angular speed, needed to estimate the attitude and the accelerometers provide acceleration measurements that are first integrated to compute the speed and then integrated again to obtain the position. However, since the position is obtained through a double integration, the processor needs to know its initial position. Moreover, to compute the necessary corrections due to the effects of the Earth gravity on the acceleration measurements, a gravity model is also included.

These systems have been well investigated in literature and in synthesis there are several advantages of using them [10]: (i) they use fully *autonomous* systems, which can operate without the need of the human control and do not need connectivity to external elements; (ii) they can be integrated with other navigation systems (as e.g., the GPS) to obtain the initial position estimation and re-calibrate it periodically; (iii) they are insensitive to *jamming* or attacks from external entities, since they do not transmit nor receive data to perform their computations, and (iv) since new sensors are recently being produced using Micro Electro Mechanical Systems (MEMS)-based technology, their weight and size are extremely reduced. On the other hand, there are also some drawbacks, such as [10]: (i) since they use integration process, their estimation errors usually tend to grow over the time; (ii) an initial calibration of the system is often needed, and that might require time, and (iii) their energy consumptions are usually high, especially if the sensors' sampling frequency is high.

Summarizing, the INS has two fundamental characteristics: the *autonomy* and *accuracy* of the estimations over the short periods. On the other hand, its main drawback is that the estimation errors grow fast over the time. For this reason, in real applications, where long term navigation is needed, an auxiliary navigation system is used to reset the INS estimations. Usually, this system is the GPS that, however, suffers from several limitations, like e.g., the need to constantly have visibility with the satellites. In this work, we show that the ESD positioning algorithm over WSN can be efficiently used as an auxiliary system, instead of the GPS, to improve the final accuracy of the navigation.

III. Enhanced Steepest Descent

In this section, the ESD distributed positioning algorithm is briefly introduced as an enhancement of the well-known Steepest Descent (SD) method. The following notation will be used here: (i) bold symbols denote vectors and matrices, (ii) $(\cdot)^T$ denotes transpose operation, (iii) $\nabla(\cdot)$ is the gradient operator, (iv) $\|\cdot\|$ is the Euclidean distance, (v) $\angle(\cdot,\cdot)$ is the phase angle between two vectors, (vi) $(\cdot)^{-1}$ denotes matrix inversion, (vii) $\hat{\mathbf{u}}_j = [\hat{u}_{j,x}, \hat{u}_{j,y}, \hat{u}_{j,z}]^T$ is the estimated position of the mobile node $\{U_j\}_{j=1}^{N_U}$, (viii) $\mathbf{u}_j = [u_{j,x}, u_{j,y}, u_{j,z}]^T$ is the trial solution of the positioning algorithm, (ix) $\bar{\mathbf{u}}_i = [x_i, y_i, z_i]^T$ denote the positions of the reference nodes $\{A_i\}_{i=1}^{N_A}$, and (x) $\hat{d}_{j,i}$ is the estimated (via ranging measurement) distance between reference node $\{A_i\}_{i=1}^{N_A}$ and blind node $\{U_j\}_{j=1}^{N_U}$.

Both SD and ESD are gradient descent methods [11]. This means that the position of a node U_1 is obtained by minimizing the error cost function $F(\cdot)$ defined as follows:

$$F(\mathbf{u}_1) = \sum_{i=1}^{N_A} \left(\hat{d}_{1,i} - \|\mathbf{u}_1 - \bar{\mathbf{u}}_i\|\right)^2 \tag{1}$$

such that $\hat{\mathbf{u}}_1 = \arg\min_{\mathbf{u}_1} \{F(\mathbf{u}_1)\}$. The minimization of (1) can be done using a variety of numerical optimization techniques, each one having its own advantages and disadvantages in terms of accuracy, robustness, convergence speed, complexity, and storage requirements [11].

1) Classical Steepest Descent: The classical Steepest Descent (SD) is an iterative line search method that allows to find the (local) minimum of the cost function in (1) at step $k+1$ as follows [11, pp. 22, sec. 2.2]:

$$\mathbf{u}_1(k+1) = \mathbf{u}_1(k) + \alpha_k \mathbf{p}(k) \tag{2}$$

where α_k is a step length factor, which regulates the convergence speed [11, pp. 36, ch. 3] and $\mathbf{p}(k) = -\nabla F(\mathbf{u}_1(k))$ is the search direction of the algorithm. In particular, since the optimization problem is non-linear, a fixed and small value of α_k is in general preferred in order to reduce the oscillatory effect when the algorithm approaches the solution.

2) Enhanced Steepest Descent: The SD method provides, in general, good accuracies in estimating the final solution. However, it may require a large number of iterations, which may result in a convergence speed too slow, especially when mobility is considered. In order to improve such convergence speed, we proposed an enhanced version, we called ESD.

The basic idea behind the ESD algorithm is to continuously adjust the step length value α_k as a function of the current and previous search directions $\mathbf{p}(k)$ and $\mathbf{p}(k-1)$, respectively. In particular, α_k is adjusted as follows:

$$\begin{cases} \alpha_k = \alpha_{k-1} + \gamma & \text{if} \quad \theta_k < \theta_{\min} \\ \alpha_k = \alpha_{k-1}/\delta & \text{if} \quad \theta_k > \theta_{\max} \\ \alpha_k = \alpha_{k-1} & \text{otherwise} \end{cases} \tag{3}$$

where $\theta_k = \angle(\mathbf{p}(k), \mathbf{p}(k-1))$, $0 < \gamma < 1$ is a linear increment factor, $\delta > 1$ is a multiplicative decrement factor, and θ_{min} and θ_{max} are two angular threshold values that control the step length update.

By using the four degrees of freedom γ, δ, θ_{min} and θ_{max}, we can simultaneously control the convergence rate of the algorithm and the oscillatory phenomenon when approaching the final solution in a simple way, and without appreciably increasing the complexity of SD. Basically, the main advantage of the ESD algorithm is the adaptive optimization of the step length factor α_k at run time, which allows to dynamically either accelerate or decelerate the convergence speed of the algorithm as a function of the actual value of the function (1).

IV. Distributed Sensor Fusion

As mentioned earlier, the INS can give very accurate position estimations, but only over short periods. On the other hands, the ESD is able to provide position estimations that are more stable over time, but less accurate. As a consequence, the two systems are complementary to each other and can be fused to improve the final system performance.

Basically, we have two estimations $\hat{x}_i$ $(i = 1, 2)$, i.e., the positions computed by the two systems, and the problem is to generate a single "optimal" estimation $\hat{x}_{1e2}$ as a fusion of $\hat{x}_1$ and $\hat{x}_2$. To do so, we need also to compute the two error covariance matrices [12]-[14] $P_{ij} = E[(x - \hat{x}_i)(x - \hat{x}_j)']$, $i = 1, 2; j = 1, 2$. The main problem for these systems is that the values of the cross–correlation matrices $P_{12} = P'_{21}$ are not known a-priori. However, if the two systems generating the estimations $\hat{x}_1$ and $\hat{x}_2$ are independent from each other, then the cross-correlation matrices P_{ij} (for $i \neq j$) can be assumed to be negligible and the *Simple Convex Combination* (SCC) [15] fusion algorithm can be efficiently used. In general, the SCC algorithm is often used due to its ease of implementation. It provides optimal estimates if the cross–covariance matrix P_{ij} is close to zero, otherwise, the estimates are sub–optimal.

Hence, the new estimation is computed as follows[1]:

$$\begin{aligned} \hat{x}_{1e2} &= P_2 (P_1 + P_2)^{-1} \hat{x}_1 + P_1 (P_1 + P_2)^{-1} \hat{x}_2 \\ &= P_{1e2} \left(P_1^{-1} \hat{x}_1 + P_2^{-1} \hat{x}_2 \right) \end{aligned} \quad (4)$$

and the error covariance matrix is $P_{1e2} = \left(P_1^{-1} + P_2^{-1} \right)^{-1}$.

V. Navigation System

In this section we show how the sensor fusion can be applied in tracking scenarios. The system performance is assessed by first relying on a MATLAB simulation model, then on preliminary experimental activities with a real IMU hardware.

To simulate the motion of a ME over a given area, we implemented a kinematic model. From this model, acceleration, speed and position of the ME are derived. Then, the INS system has been simulated (in a bi-dimensional space) as two acceleration sensors that feed a Kalman filter to track the movements of the ME. In parallel, the measurements of distances of the ME from four anchor nodes in the field have been simulated, using an additive zero-mean Gaussian ranging error model, and feed the ESD algorithm to produce the position estimation. These two estimations finally feed the SCC fusion, that provides a global position estimation.

[1]To simplify the notation, we assume $P_1 = P_{11}$ and $P_2 = P_{22}$

A. Kinematic Model

Among the plethora of solutions presented in literature to model the movement of an entity [9], [12], in this work the Discrete White Noise Acceleration (DWNA) model has been adopted. DWNA is also called as *piecewise constant white acceleration*, since it considers the acceleration as constant during the sampling intervals. The model is described by the following system of equations:

$$\begin{cases} x_{k+1} = x_k + \dot{x}_k \Delta T + \frac{1}{2} \ddot{x}_k \Delta T^2 \\ y_{k+1} = y_k + \dot{y}_k \Delta T + \frac{1}{2} \ddot{y}_k \Delta T^2 \\ \dot{x}_{k+1} = \dot{x}_k + \ddot{x}_k \Delta T \\ \dot{y}_{k+1} = \dot{y}_k + \ddot{y}_k \Delta T \\ \ddot{x}_{k+1} = n_{\ddot{x}} \\ \ddot{y}_{k+1} = n_{\ddot{y}} \end{cases} \quad (5)$$

where: (i) k and $k+1$ are the sampling instants; (ii) $\Delta T = T_{k+1} - T_k$, is the sampling period; (iii) x and y are the 2D coordinates of ME; (iv) $\dot{x}, \dot{y}, \ddot{x}, \ddot{y}$ are the coordinates of the speed ($\frac{m}{s}$) and acceleration ($\frac{m}{s^2}$) vectors, respectively, and (v) $n_{\ddot{x}}, n_{\ddot{y}}$ are the random values of acceleration ($\frac{m}{s^2}$), assumed as constants within each ΔT, and having a 0-mean Gaussian distribution with standard deviation $\sigma_{n_{\ddot{x}}}$ and $\sigma_{n_{\ddot{y}}}$.

These equations can be written using matrices as:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \tilde{F}\mathbf{N}'_k \quad (6)$$

where $\mathbf{x}_k$ is the state vector at the step k, A is the state transition matrix, $\tilde{F}$ is a gain matrix and $\mathbf{N}'_k$ is the noise vector.

For the sake of simplicity, in this model we implicitly assumed that rotations of the ME are negligible with respect to the movements along the axes: in other words, the ME is as a *point* that moves on the *xy-plane*. However, in order to make the DWNA more realistic, an additional noise term has been included to model the interaction of the ME with obstacles. Basically, the interaction is modeled as a random acceleration vector computed at time k added to the acceleration provided by the model: this causes the ME to move on a different direction. Moreover, this additional vector is applied or not based on a Bernoulli random decision variable (with probability μ).

After having simulated the *true* trajectory of the ME, a Kalman Filter has been implemented as the INS' navigation processing core to estimate the coordinates *(x,y)* of the ME from its acceleration measurements. In parallel, the gradient descent methods have been implemented as described in Section III to estimate the position from the measurements of distances w.r.t. the anchors.

1) Kalman Filter: In realistic conditions, the acceleration measurements given by the IMU are noisy. As a consequence, the *observation* vector $\mathbf{z}$, i.e., the measurements from the sensors, is modeled as:

$$\mathbf{z}_k = C\mathbf{x}_k + \tilde{G}\mathbf{N}''_k \quad (7)$$

where C is the state observation matrix, $\tilde{G}$ is the noise matrix and $\mathbf{N}''_k$ is the observation noise vector, which is supposed to have a zero-mean Gaussian distribution. Similarly to the state noise, it is possible to compute the covariance matrix for $\boldsymbol{N}''_k = [\epsilon_{ax}, \epsilon_{ay}]^T_k$ as:

$$E[\mathbf{N}''_k \mathbf{N}''^T_j] = \begin{bmatrix} \sigma^2_{\epsilon_{ax}} & 0 \\ 0 & \sigma^2_{\epsilon_{ay}} \end{bmatrix} \tag{8}$$

where $\epsilon_{ax}, \epsilon_{ay}$ are the noise of the acceleration measurements over the two axes.

Thus, the final system is:

$$\begin{cases} \mathbf{x}_{k+1} = A\mathbf{x}_k + \tilde{F}\mathbf{N}'_k \\ \mathbf{z}_k = C\mathbf{x}_k + \tilde{G}\mathbf{N}''_k \end{cases} \tag{9}$$

Since the state noise ($\mathbf{N}'_k$) and the measurement noise ($\mathbf{N}''_k$) are zero-mean Gaussian random vectors and independent from each other, they can be grouped into a single noise vector $\mathbf{N}_k = [\mathbf{N}'_k, \mathbf{N}''_k]^T$, with the associated covariance matrix:

$$E[\mathbf{N}_k \mathbf{N}^T_j] = \begin{bmatrix} \sigma^2_{n_{\ddot{x}}} & 0 & 0 & 0 \\ 0 & \sigma^2_{n_{\ddot{y}}} & 0 & 0 \\ 0 & 0 & \sigma^2_{\epsilon_{ax}} & 0 \\ 0 & 0 & 0 & \sigma^2_{\epsilon_{ay}} \end{bmatrix} \tag{10}$$

2) Gradient Descent Methods: In order to apply the SCC, the covariance matrices must be computed. While the Kalman Filter provides already such error covariance matrix together with the position estimation, with the SD and ESD the covariance matrices have to be computed from scratch. This computation is done through a time-based technique, considering the estimated position by the SD and the ESD at each time k and the measured distances from the reference nodes. In particular, since the error function to be minimized is defined as in (1), the variance of the error can be computed as the variance of a vector $Error\left(\tilde{u}^h_j\right)$ whose elements are:

$$Error\left(\tilde{u}^h_j\right) = \sum_{i=1}^{N_A} \left(e^h_{i,j}\right)^2 \tag{11}$$

where $e^h_{i,j} = \hat{d}_{i,j} - \left\|\mathbf{u}_i - \bar{\mathbf{u}}^h_j\right\|$ is the error w.r.t. the anchor $\mathbf{u}_i$, $\bar{\mathbf{u}}^h_j$ is the position estimation at the step h and $i = 1, \ldots, N_A$, $j = 1, \ldots, N_U$. Then the variance of this error must be computed considering in such a vector the estimations obtained at steps h, with $h = k - m, \ldots, k$ (i.e., considering a moving window of the last m estimations).

B. Simulation Analysis

The simulated scenario is similar to Fig. 2, where a ME is supposed to move from an environment equipped with a WSN (and in particular with four anchor nodes) to another similar one. In the middle, the ME traverses an area where no anchor nodes are present, but we suppose the ME still maintain the radio connectivity with the old anchors. This is a typical scenario mixed indoor–outdoor. The goal is to show that the fusion is able to choose the optimal position estimation along the trajectory.

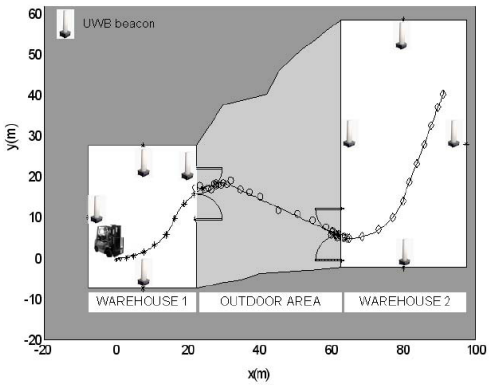


Fig. 2. Mixed Indoor–Outdoor Scenario [16].

The simulation setup is as follows[2]: (i) simulation duration $n = 100$ s; (ii) $\Delta T = 1$ s; (iii) standard deviations of the state noise, distances[3] and accelerations equal to $\sigma_{n_{\ddot{x}}} = \sigma_{n_{\ddot{y}}} = 0.5\ \frac{m}{s^2}$, $\sigma_{d_{i,j}} = 0.1$ m and $\sigma_{\epsilon_{ax}} = \sigma_{\epsilon_{ay}} = 0.25\ \frac{m}{s^2}$, respectively; (iv) the maximum number of iterations for both SD and ESD algorithms is 100, (v) the error tolerance for the minimization is $\varepsilon = 0.001$ (i.e., when for a given position estimation the error function falls below ε the gradient-based algorithms stop, before reaching the maximum number of iterations); (vi) the initial position estimation for both SD and ESD at every run (i.e., every ΔT) is reset into the origin of the system's coordinates; (vii) the learning speed[4] is $\alpha_0 = 0.5$; (viii) the degrees of freedom for the ESD algorithm are: $\gamma = 0.1$, $\delta = 1.75$, $\theta_{min} = 5°$ and $\theta_{max} = 30°$; (ix) the Bernoulli probability of the ME to change direction due to obstacles is $\mu = 0.1$, and (x) the dimension of the mobile window to compute both SD and ESD error covariances is $m = 20$ s, according to (11).

We suppose that every ΔT the estimations from the INS and the gradient methods are available and synchronous and, since the ME motion is computed using the kinematic model described in Section V-A, to simplify the simulation, we assume that at the step $k = 80$ (i.e., after 80 seconds) both SD and ESD instantaneously switch to the second set of anchor nodes. Finally, in order to compare the performance of the solutions proposed, even if the traces are synchronous, the Dynamic Time Warping tool [17] was chosen as similarity metric. This tool generally measures the similarity[5] between two traces, by automatically correcting any timing discrepancy.

Given the above setup and assumptions, Fig. 3 reports the

[2]Due to space constraints, the results are shown w.r.t. a single setup.

[3]In [8] a RSS-based ranging technique has been adopted with the ESD algorithm. Here $\sigma_{d_{i,j}}$ accounts for the statistics of the ranging errors, independently from which (RSS-, Angular- or Time-based) model is actually used.

[4]Initial value for the ESD and fixed value for the SD

[5]Similarity is measured as a distance between the two curves; thus lesser the DTW value, more similar the two curves are.

results of a simulation run composed by 100 trials, in each of them a ME travels from an initial position close to anchors A1-A4 to one close to A5-A8, similarly to the reference scenario in Fig. 2, following a random walk, as described in Section V-A. Fig. 3 shows the cumulative distribution of the error measured by the DTW for all the tracking algorithms available: INS, SD, ESD and two SCC fusions, one that considers INS and SD and the other between INS and ESD.

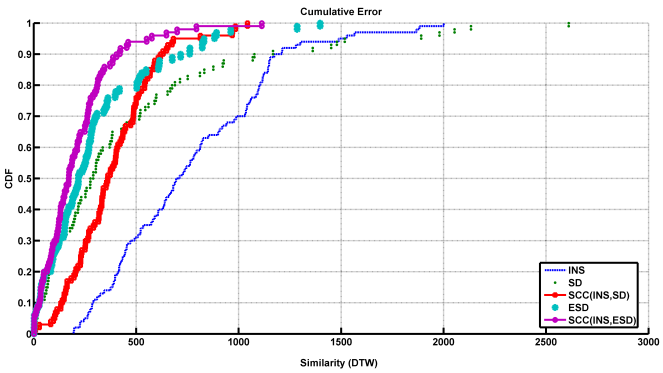


Fig. 3. Cumulative error distributions of the considered tracking methods.

From Fig. 3 the following conclusions can be drawn: (i) the INS as a tracking solution alone suddenly shows bad performance being really sensible to the high value of σ_1, σ_2 and the presence of obstacles; (ii) the ESD and the SD as independent solutions show better performance than INS and this is mostly due to the choice of $\sigma_{d_{i,j}}$ with respect to $\sigma_{\epsilon_{ax}}, \sigma_{\epsilon_{ay}}$; (iii) as evidenced in previous work [8], it is confirmed that ESD outperforms SD in terms of accuracy of the final estimation, and (iv) overall, the sensor fusion algorithm improves the performance of the tracking system.

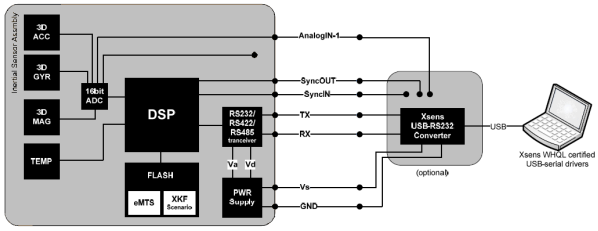
In general, the comparison of SCC(INS,SD) with SCC(INS,ESD) confirms that the key to obtain good tracking performance is the implementation of efficient algorithms together with the use of sensors able to provide very accurate measurements. With the help of this MATLAB tool, future work will include the analysis of the system performance as a function of parameters, such as $\sigma_{n_{\ddot{x}}}, \sigma_{n_{\ddot{y}}}, \sigma_{d_{i,j}}, \sigma_{\epsilon_{ax}}$ and $\sigma_{\epsilon_{ay}}$.

C. Experimental Analysis

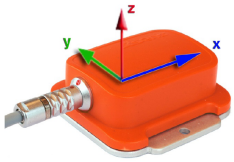
The encouraging simulation results have been validated against preliminary real experiments using the *XSENS MTi* Motion Tracker [18]. It is a small-size IMU with a 3D compass and an embedded processor able to compute in real-time roll, pitch and yaw, as well as the linear 3D accelerations, angular speed and Earth magnetic field.

1) XSENS MTi *Motion Tracker:* The MTi orientation is estimated using a XSENS internal Kalman filter (XKF3) which fuses the signals from the gyroscopes, accelerometers and magnetometers to compute an optimal statistical 3D orientation with high accuracy. By the XKF3 algorithm, the Earth magnetic field and gravity measurements enable an *attitude and heading* drift errors compensation [18].

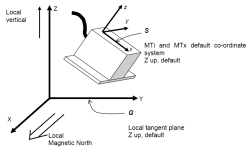
Fig. 4(a) shows how the MTi can be connected to a PC through the USB-serial adapter, where the sensor outputs are collected. Every sensor measurement is referred to the Cartesian system as shown in Fig. 4(b). This reference system is called *S* and is *body-fixed* to the device. The orientation of the sensors are then computed between *S* and the *Earth-fixed* system (called *G*), as shown in Fig. 4(c).



(a) Scheme and interface with the PC.



(b) Coordinates system.



(c) Reference system.

Fig. 4. Details of the *MTi* IMU.

2) 2D tracking: The Kalman filter and DWNA kinematic model have been applied to estimate the trajectory of the MTi sensor, starting from its acceleration measurements. However, in order to use the Kalman filter, the covariance matrices of the state and the output noises are needed. While the former was computed using the whole set of the acquired measurements, the latter was inferred from the nominal noise density values reported in the MTi datasheet [18].

To simplify the experiment, the MTi device was moved on a table, like a PC's mouse. By this way, Roll and Pitch contributions are close to zero, then the Earth's gravity contribution can be filtered out by simply subtracting the average value of the measured accelerations for the *x* and *y* axes.

Fig. 5 shows the reconstructed trajectory of a clockwise movement of the MTi device along a rectangular shape[6] of *10 cm* x *8 cm*. Interestingly, the system shows very good performance in estimating the trajectory for quick movements of the device, while, as confirmed in [19], for low acceleration values the reconstructed trajectory is not satisfactory, due to the effects of the noise in the measurements.

VI. Conclusions and Perspectives

In this work we proposed a positioning and tracking system based on the fusion of position estimations provided by two different and independent techniques. Thanks to the MATLAB

[6]An experimental campaign is currently on-going at the University of L'Aquila using the platform in more realistic and wider scenarios.

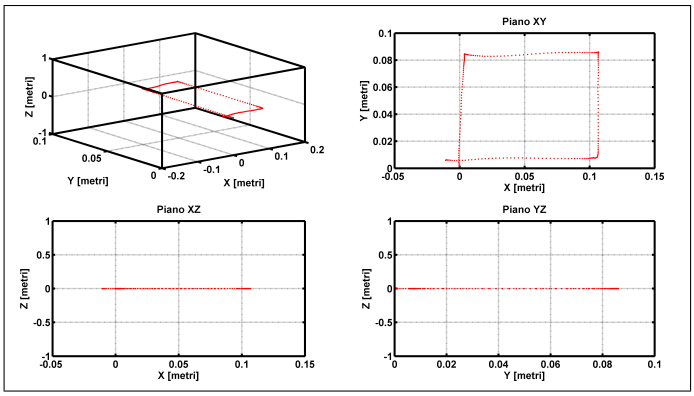


Fig. 5. Example of estimated trajectory.

simulator we developed, we are able to assess the performance limits of the algorithms (i.e., INS, SD, ESD and SCC) when they are used to estimate the positions of a mobile entity. Simulation results clearly show the advantage of the integration of different systems: sensor fusion provides better accuracies with respect to those provided by each individual algorithmic component, but the best joint performance are dependent on the accuracy of each individual component.

The Kalman Filter built on top of the DWNA kinematic model has been tested also using measurement traces acquired by a real platform, the XSENS Motion Tracker MTi, that Thales Italia made available to the Center of Excellence DEWS in L'Aquila (Italy). Preliminary results confirm the good performance when the movement traces have large acceleration impulses, while for slow movements, i.e. small acceleration, the noise level becomes dominant.

In the current version of the simulator, the Kalman filter needs to know the covariance statistic matrices, thus the acceleration measurements must be already available. This limits the applicability of the proposed solution in scenarios requiring hard real-time constraints. On-going work includes the substitution the INS of the Kalman filter with a faster, but sub-optimal time-based algorithm (such as a trajectory prediction). By this way, the INS system can reset its initial position estimation, as soon as the SCC detects that the ESD is going to have a better accuracy.

Longer term future work will include the development of realistic signal propagation models [20] in the MATLAB tool, in order to enable a smoother transition for the ESD among different WSN-equipped spaces (as in Fig. 2). This will foster new studies about anchor deployment planning for WSNs, to cope with tracking requirements but with knowledge of the new performance limits provided by the sensor fusion technology.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments.

The research activity of Stefano Tennina is supported by the EMMON project, funded by National Funds, through the FCT - Portuguese Foundation for Science and Technology, under grant ref. ARTEMIS/0003/2008, as well as by the ARTEMIS Joint Undertaking, under grant agreement n. 100036. The work done by Fortunato Santucci and Fabio Graziosi is supported by the FP7 NoE HYCON2, project number 257462 and project IRMA2 (Ministry of Defense of Italy).

REFERENCES

[1] A. Goldsmith and S. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *Wireless Communications, IEEE*, vol. 9, no. 4, pp. 8 – 27, aug. 2002.

[2] M. Dohler, "Wireless sensor networks: the biggest cross-community design exercise to-date," *Bentham Recent Patents on Computer Science*, vol. 1, no. 1, pp. 9–25, 2008.

[3] J. Bachrach and C. Taylor, *Handbook of sensor networks: algorithms and architectures - Localization in sensor networks*, I. Stojmenovic, Ed. Wiley Series on Parallel and Distributed Computing, 2005.

[4] C. Wang and L. Xiao, "Sensor localization under limited measurement capabilities," *Network, IEEE*, vol. 21, no. 3, pp. 16 –23, may-june 2007.

[5] M. Mauve, A. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *Network, IEEE*, vol. 15, no. 6, pp. 30 –39, nov/dec 2001.

[6] N. Patwari and J. Wilson, "RF sensor networs for device-free localization: Measurements, models, algorithms," *Proceedings of the IEEE*, 2010.

[7] ——, "See through walls: Motion tracking using variance-based radio tomography networks," *IEEE on transaction on mobile computing*, 2010.

[8] S. Tennina, M. Di Renzo, F. Graziosi, and F. Santucci, "Locating zigbee nodes using the ti's cc2431 location engine: a testbed platform and new solutions for positioning estimation of wsns in dynamic indoor environments," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, ser. MELT '08. New York, NY, USA: ACM, 2008, pp. 37–42. [Online]. Available: http://doi.acm.org/10.1145/1410012.1410022

[9] J. Hightower and G. Borriello, "Particle filters for location estimation in ubiquitous computing: A case study," in *In Proceedings of International Conference on Ubiquitous Computing (UbiComp*, 2004, pp. 88–106.

[10] Grewal, Weill, and Andrews, *Global Positioning System and Inertial Navigation and Integration*. Jhon Wiley, 2007.

[11] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer, Ed. Springer, 2006.

[12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation. Theory Algorithms and Software*. John Wiley and Sons, 2001.

[13] D. Silva, Brancalion, and Fernandes, "Data fusion techniques applied to scenarios including ADS-B and radar sensors for air traffic control," *12th International Conference on Information Fusion, USA*, 2009.

[14] "Comparison of two measurement fusion methods for kalman-filter-based multisensor data fusion," *IEEE transaction on aerospace and electronic system*, 2001.

[15] S. J. Julier and J. K. Uhlmann, "A non divergent estimation algorithm in the presence of unknown correlations," *American Control Conference*, 1997.

[16] J. Fernandez-Madrigal, E. Cruz-Martin, J. Gonzalez, C. Galindo, and J. Blanco, "Application of uwb and gps technologies for vehicle localization in combined indoor-outdoor environments," in *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, feb. 2007, pp. 1 –4.

[17] G. Li, Y. Wang, M. Li, and Z. Wu, "Similarity match in time series streams under dynamic time warping distance," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 4, dec. 2008, pp. 399 –402.

[18] *MT Software Development Kit Documentation*, Rev. j ed., Xsens Technologies, May 2009. [Online]. Available: http://www.xsens.com/en/general/mti

[19] A. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," *6th IEEE International Symposium on Intelligent Signal Processing*, 2009.

[20] O. Ozdemir, R. Niu, and P. Varshney, "Tracking in wireless sensor networks using particle filtering: Physical layer considerations," *Signal Processing, IEEE Transactions on*, vol. 57, no. 5, pp. 1987 –1999, may 2009.