Abstract
This article focuses on human navigation, by proposing a system for mapping and self-localization based on wearable sensors, i.e., a laser scanner and a 6 Degree of Freedom Inertial Measurement Unit (6DOF IMU) fixed on a helmet worn by the user. The sensor data are fed to a Simultaneous Localization And Mapping (SLAM) algorithm based on particle filtering, an approach commonly used for mapping and self-localization in mobile robotics. Given the specific scenario considered, some operational hypotheses are introduced in order to reduce the effect of a well-known problem in IMU-based localization, i.e., position drift. Experimental results show that the proposed solution leads to improvements in the quality of the generated map with respect to existing approaches.

Keywords: human navigation and mapping, IMU-based dead reckoning, laser-based SLAM

1. INTRODUCTION
This article focuses on human navigation and is especially oriented towards guaranteeing the safety and efficiency of a group of first-responders involved in indoor search and rescue operations, such as during earthquakes, fires or floods. The objective is to propose a system for mapping and self-localization based on wearable sensors. The information returned by the system could be used, for instance, to explore the area in an optimal way or to find a safe evacuation route (i.e., providing guidance similarly to a GPS navigation system).

Human navigation has started to receive attention from the scientific community only recently [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. In this domain, an Inertial Measurement Unit (IMU) is a very common source of information to detect human motion. Since IMUs are often noisy and, even worse, it is necessary to integrate acceleration data twice to estimate the position, most works in the field focus on solutions to reduce the resulting position drift, which is a consequence of errors in the estimated velocities. For example, the authors of [2, 3, 4, 5, 14] propose to use a technique called Zero Velocity Update: they fix the accelerometers to the user’s foot and then, analysing the pattern of accelerations, identify a time interval in which the foot stands on the ground and therefore velocities can be reliably estimated to be zero.

1The work has been partially funded by University of Genova and Regione Liguria / SIIT.

Email address: marco.baglietto@unige.it, antonio.sgorbissa@unige.it, damiano.verda@unige.it, renato.zaccaria@unige.it (Marco Baglietto, Antonio Sgorbissa, Damiano Verda, Renato Zaccaria)
The present project exhibits some major differences with respect to the works above. The main idea of this work is to take inspiration from technologies exploited in mobile robotics (e.g., in rescue robotics [15, 16]) not only to estimate the user position, but also to generate a map of the environment: this concept is also referred to as Simultaneous Localization And Mapping (SLAM) [17].

In order to perform SLAM, it is assumed that the user is equipped not only with inertial sensors, but also with a laser scanner which returns proximity measurements about the environment. In order for such measurements to be significant, the laser scanner itself must be raised to an height allowing it to detect obstacles which play an important role for updating the map (like walls and furniture) and to ignore, for example, small objects on the floor. The solution adopted here is to couple a laser scanner and a 6DOF IMU on a helmet (Figure 1). Since the Zero Velocity Update is not applicable in this case, considering that the inertial sensors are not fixed to the user’s foot, the proposed approach requires to deal with the well–known problems deriving from the double integration of noisy data.

The major contribution of this work is to propose a novel solution for Human Navigation and Mapping which relies on an original approach to face position drift: specifically, we show that the output of a standard SLAM algorithm for 2D mapping and some additional heuristics can be used to improve the position estimate provided by the IMU.

The possibility of adapting SLAM technique for human navigation has been considered also in other works, e.g., [18, 19]. The work described in [18] relies on a Particle Filter (PF) [20], but the approach is different in that the data provided by the IMU are not integrated to compute an estimate of the user position, but they are simply used to distinguish if the user is moving or not. This information is then used to update position by assuming that the user is walking forward with an almost constant velocity. The approach proposed by [19] has a completely different focus. It relies on a different technique based on the Extended Kalman Filter (EKF) [20], and puts strong constraints on the environment, which is modelled as constituted of orthogonal structural planes.
Section 2 describes related work. Section 3 describes the system architecture, the approach adopted for Human Localization and Mapping, and for drift reduction. Section 4 describes experimental results. Conclusions follow.

2. RELATED WORK

Existing solutions to the problem of human navigation can be roughly divided into two classes: approaches which deal only with path-reconstruction (also referred to as Pedestrian Dead-Reckoning), and approaches which build also a map of the environment as the user moves (i.e., performing SLAM).

2.1. Dead-reckoning

Solutions belonging to the first class often rely on models of the human gait [21] in order to deal with the drift problem which follows from the double integration of noisy data. The authors of [1] propose to fix three accelerometers to the user’s foot, under the assumption that, at each step, it is possible to recognize a characteristic acceleration pattern [22]. The direction of each step is estimated using magnetometers. Then, the user position can be computed by considering that, as shown in [23, 24], the length of a human step is not constant, but it varies around a stable pivotal value. The system achieves good results, but it is subject to a position error which grows with the number of steps that the user takes. To reduce this effect, the authors of [2, 3] propose a technique called Zero Velocity Update. By analysing the acceleration pattern during a step, they observe that it is possible to identify a time interval during which the foot stands on the ground. Since velocity can be assumed to be equal to zero in this time interval (unless the user is dragging his foot), this information can be used to limit the error introduced by the integration of noisy acceleration data. Experimental results show that this technique reduces the position error. The authors of [7] propose a model of human motion based on multiple motion modalities, which correspond to different ways to update the user position. In [25] the authors focus on the development of an algorithm for adaptive step length estimation. Other works propose innovative pedestrian dead-reckoning systems: for instance by considering the situation that the user is keeping a mobile phone (equipped with inertial sensors) in her hands [11], or in the pocket of her trousers [12], i.e., without making a priori assumptions on the position and orientation of the sensor itself. Finally, the idea of assisting first responders with this kind of technologies is exploited in the WearIT@Work project\(^3\) presented in [14]. Our approach differs from the works above in that it does not assume any model of the human gait. This is also a consequence of the fact that we couple a 6DOF IMU and a laser scanner on a helmet, which turns out to be a good design solution to reduce occlusions and correctly interpret range data while performing SLAM, but not for detecting human steps. Then, to reduce positioning drift, we propose a different approach which relies on heuristics and on the output of SLAM itself.

Some approaches in the literature propose to integrate a number of sensors to achieve a higher positioning accuracy. This is the case, for instance, of [6], which relies on a IMU, a GPS, an electronic compass, as well as a pedometer based on heel and toe contact switches. The approach proposes an adaptive knowledge system based on Artificial Neural Networks (ANN) and Fuzzy Logic, which is trained during the GPS signal reception and used to maintain navigation under GPS-denied conditions. In [9] a pedestrian inertial navigation device called NavShoe is

\(^3\)http://www.wearitatwork.com/
presented. This device can work in arbitrary unstructured indoor/outdoor environments and it consists of a miniature low-power inertial/magnetometer package tucked into the shoelaces of one foot, wirelessly coupled to a PDA that runs software to merge inertial, geomagnetic, and optional GPS measurements to derive an optimal position estimate. In [4, 5] the user is provided with ultrasonic sensors to be deployed in the environment during exploration, and to be used as landmarks to reduce the estimation error. This approach produces good experimental results, ensuring also the possibility to manage a high level of electromagnetic noise (which deteriorates the accuracy of data taken by magnetometers) in a better way. Our approach differs from the works above in that it does not integrate GPS, compasses, magnetometers, or environmental sensors as a support to estimate the absolute position or heading of the user.

2.2. Map-building

Solutions belonging to the second class usually refer to SLAM approaches in the literature, and adapt them to human navigation. SLAM itself is a broad field of research, and an extensive review is outside the scope of this article [20]: therefore, only works related to humanoid robots and human navigation are discussed in the following.

The authors of [26, 8, 27, 28] deal with the problem of performing SLAM with humanoid robots: with respect to mobile robotics, this field of research deserves a greater attention since it shares more similarities with human navigation. In [26] the authors present a technique designed to perform 3D EKF-based SLAM for a humanoid robot equipped with a single camera. Experiments with a human-size robot show accurate positioning, but only when the robot is moving within an area smaller than 2×2 m². Moreover, the system heavily relies on the fact that the walking-motion pattern generator is known in advance and can be fed to the EKF: this information is not available in our approach (and in human navigation in general). In [28] the authors propose a system for autonomous navigation and mapping with odometry data and two laser scanners mounted on the feet of a human-size robot. As in our approach, the authors choose DP-SLAM [29, 30], and produce maps which are sufficiently accurate for autonomous navigation in a typical office environment. However, the position of laser scanners makes the approach infeasible for a real-world application in a cluttered environment. A similar approach is proposed in [8], by mounting the laser scanner on the robot head: once again, the approach relies on accurately modelling odometry to produce a more accurate particle distribution for the particle filter which performs SLAM. In [27] the problem of learning grid maps with a 1-meter tall humanoid robot is solved using a Rao-Blackwellized particle filter. The robot is equipped with a laser scanner and inertial sensors for computing the chest attitude, and exhibits accurate positioning and map building in a 20×20 m² area without relying on odometry. The approach shares some similarities with the present work: however, the motion velocity of humanoid robots is necessarily smaller than that of humans, and therefore comparing performance is difficult.

Few works [18, 19, 31, 13] deal explicitly with human navigation and SLAM. Specifically, the HeadSLAM proposed in [18] shares many similarities with the present work, in that it couples a laser scanner and an IMU on a helmet in order to estimate motion and correct scan data. Differently from our approach, which integrates acceleration data to produce an estimate of the position, the authors use vertical accelerations as a “boolean sensor” to distinguish between two opposite states, i.e., the user is either walking or not. In the former case, an average linear speed of 1.35 m/s is assumed, and used to provide an initial odometry guess to be fed to SLAM. Gyroscopes are used to provide heading information: nevertheless, the approach looks incapable of distinguishing those situations where the user is moving backward or laterally, or even when the
head is not aligned with the direction of motion of the body. The authors of [19] present a Laser-Aided Inertial Navigation System (L-INS) especially thought for visually impaired people: the IMU measurements are integrated to obtain pose estimates, which are subsequently corrected through an EKF using line-to-plane correspondences between linear segments in the laser-scan data and structural planes of the building (e.g., walls). Experiments shows that the system is able to build reliable maps: however, differently from our work, it assumes that indoor structural planes are orthogonal to each other, thus working only in simpler indoor environments. It is worth noticing that neither of the approaches above rely on an accurate model of the human gait: the work described in [19] switches to Zero Velocity Update techniques only when range data are not available for some time. Finally, [31, 13] propose FootSLAM and PlaceSLAM, two systems for human navigation and mapping which do not rely on the observations provided by range or visual data. Instead, FootSLAM relies only on the information provided by wearable accelerometers to detect steps: since human motion is constrained by the topology of the environment, this information can be integrated to produce a map of the environment itself. PlaceSLAM further elaborates this concept by integrating on-line features observed by the user herself to label the map, thus improving performances. These latter approaches are very different from the present work both in their general spirit and in the sensors and techniques adopted for map-building.

3. HUMAN NAVIGATION AND MAPPING

An overview of the system architecture is shown in Figure 2. Two main blocks are visible: 2D SLAM and Velocity & Attitude Estimation.

![Figure 2: Block diagram which shows the main components of the system and information flow.](image)

The 2D SLAM block represents a typical laser-based SLAM algorithm returning a 2D map...
of the environment. In the current implementation we have chosen Distributed Particle SLAM (DP-SLAM)\cite{29, 30}, which has the interesting property that it does not make any assumptions about landmarks: the problem of data association is solved by storing not only the hypothetical position of significant environmental features, but whole maps. Even if a map is a much more complex type of data with respect to a landmark, DP-SLAM can still guarantee acceptable execution time using an accurate technique, called DP-Mapping, to manage the usage of memory and computational resources. Every iteration, 2D SLAM requires in input an estimate of the user motion in the 2D Workspace corresponding to a plane parallel to the floor, as well as a vector of observations projected on the same plane.

The Velocity & Attitude Estimation block, which is the focus of the present work, has three main subcomponents:

1. **Prediction** integrates the data provided by the 6DOF IMU to estimate the user’s velocity and attitude in the 3D Workspace;
2. **Correction**, together with **Prediction**, implements an EKF-inspired algorithm to correct the estimated velocity and attitude, on the basis of the information returned by 2D SLAM and additional heuristics, to the end of reducing position drift;
3. **Projection** projects the user’s velocity as well as laser range data on a plane parallel to the floor (i.e., to be fed to 2D SLAM).

In the following, all these components are described in greater details.

### 3.1. Velocity & attitude estimation

Figure 3 on the right shows the 6DOF IMU mounted on the helmet. The body frame \( \mathbf{b} \), also referred to as IMU-frame, is aligned with the axes of the IMU, with the \( b_x \) axis pointing forward, the \( b_y \) axis pointing leftward, and the \( b_z \) axis pointing upward. The navigation frame \( \mathbf{n} \), represented by the orthogonal axes North, West, and Up (NWU), is a fixed coordinate frame with respect to which the location of the IMU-frame must be computed (with the NW plane parallel to the floor, and the \( U \) axis pointing upward).

We define a state vector

\[
\mathbf{x} = \begin{bmatrix} x & y & z & v_x & v_y & v_z & \phi & \theta & \psi \end{bmatrix}^T,
\]

where the position and velocity of the IMU-frame with respect to the navigation frame are represented by \( x, y, z, v_x, v_y, v_z \), and its orientation is represented by the three Euler angles \( \phi \) (roll), \( \theta \) (pitch), and \( \psi \) (yaw), where the order of rotation is about \( b_z \), followed by \( b_y \), and then \( b_x \).

The control vector is defined as

\[
\mathbf{u} = \begin{bmatrix} a_{b_x} & a_{b_y} & a_{b_z} & \omega_{b_x} & \omega_{b_y} & \omega_{b_z} \end{bmatrix}^T,
\]

where \( a_{b_x}, a_{b_y} \) and \( a_{b_z} \) indicate the linear accelerations along the three axis of the IMU-frame,
and \( \omega_{bx}, \omega_{by} \) and \( \omega_{bz} \) indicate the angular velocities. Then, the state equations of the system are

\[
\begin{align*}
\dot{x} &= v_x, \\
\dot{y} &= v_y, \\
\dot{z} &= v_z, \\
\dot{v}_x &= a_{bx} \cos \theta \cos \psi + a_{by} (\sin \varphi \sin \theta \cos \psi - \cos \varphi \sin \psi) + a_{bc} (\sin \varphi \sin \psi + \cos \varphi \sin \theta \cos \psi), \\
\dot{v}_y &= a_{bx} \cos \theta \sin \psi + a_{by} (\cos \varphi \cos \psi + \sin \varphi \sin \theta \sin \psi) + a_{bc} (\cos \varphi \sin \theta \sin \psi - \sin \varphi \cos \psi), \\
\dot{v}_z &= -a_{bx} \sin \theta + a_{by} \sin \varphi \tan \theta + a_{bc} \cos \varphi \cos \theta + g, \\
\dot{\varphi} &= \omega_{bx} + \omega_{by} \sin \varphi \tan \theta + \omega_{bc} \cos \varphi \tan \theta, \\
\dot{\theta} &= \omega_{by} \cos \varphi - \omega_{bc} \sin \varphi, \\
\dot{\psi} &= \omega_{by} \sin \varphi \cos \theta + \omega_{bc} \frac{\cos \varphi}{\cos \theta}.
\end{align*}
\]

Given that it is impossible to know the actual components of \( u \), these quantities are measured through the IMU, which returns an approximation of the control vector, i.e.,

\[
\hat{u} = \begin{bmatrix} a_{bx} & a_{by} & a_{bc} & \omega_{bx} & \omega_{by} & \omega_{bc} \end{bmatrix}^T.
\]

Using equations (3)–(11), the data returned by the IMU can be integrated to produce a prediction of the user attitude \( \varphi, \theta, \psi \), velocity \( v_x, v_y, v_z \), and hence position \( x, y, z \). This leads to discrete time state equations in the form

\[
x_k = f(x_{k-1}, u_k).
\]

Specifically, \textit{Prediction} integrates (6)–(11) to produce an estimate of the user velocity \( \hat{v}_x, \hat{v}_y, \hat{v}_z \) and attitude \( \hat{\varphi}, \hat{\theta}, \hat{\psi} \) in the 3D Workspace; \textit{Correction} updates this estimate when observations are available (described in the next Section); \textit{Projection} refers to the estimated values \( \hat{v}_x, \hat{v}_y, \hat{\varphi}, \hat{\theta} \), together with the control inputs \( \hat{\omega}_{by}, \hat{\omega}_{by}, \hat{\omega}_{bc} \), to estimate the user motion on the \( NW \) plane of the navigation frame according to (3) (4), (11), i.e.,
\( u_x = v_x \) \hspace{1cm} (14)  
\( u_y = v_y \) \hspace{1cm} (15)  
\( u_\psi = \omega_{by} \sin \phi \cos \theta + \omega_{bz} \cos \phi \cos \theta \) \hspace{1cm} (16)

2D SLAM uses \( \dot{u}_x, \dot{u}_y, \dot{u}_\psi \) to update particles, and returns the most likely hypothesis about the position \( x_S, y_S \) and heading \( \psi_S \) of the user by taking into account the additional laser range data.

Experiments are performed with a low-cost IMU composed of a 3–axes accelerometer\(^5\) and three 1–axis gyroscopes\(^6\). By naively integrating (9)–(11) with a sampling time \( \Delta T \) equal to 10 ms, an acceptable accuracy in angular measurements is observed. In Figure 4 the angular velocity and position are shown, while performing a rotation of \( 2\pi \) radians around one axis. The final angle is approximately equal to \( 2\pi \), with an error in the order of \( 10^{-2} \) radians.

On the other hand, the linear accelerations need to be integrated twice to get a position estimate: first, accelerations are projected on the navigation frame and integrated to produce linear velocities (6)–(8), then velocities must be integrated again to produce an estimate of the position (3)–(5). In absence of any strategy for drift reduction, errors in the estimated velocities introduce an increasing positioning error which deteriorates more and more the position estimate as time passes. Figure 5 refers to an experiment where the IMU is initially not in motion, next it is moved for a given distance along one axis, and then stopped. The estimated velocity increases (and so does position) when the user starts moving, but at the end of the experiment the velocity does not decrease to 0, thus producing a drift in the position estimate.

The presence of 2D SLAM helps in reducing this effect, since it relies on laser range data to provide an estimate of the user position \( x_S, y_S \) and heading \( \psi_S \) which is more reliable than computed by naively integrating (3)–(11). However, SLAM itself is affected by errors in \( \dot{v}_x, \dot{v}_y \) and

\(^5\)MMA7260Q model.  
\(^6\)LISY300AL model.
\( \dot{a}, \dot{\theta}, \dot{\phi} \), since these components of the state are required by Projection to compute the user motion on the NW plane, and to consequently propagate particles. This problem will be discussed in the next Section, where the role played by Correction is described in details.

Finally notice that Projection implements also a policy to correctly interpret laser data. In fact, when the user neck is inclined forward/backward or left/right (i.e., \( \theta \neq 0 \) or \( \varphi \neq 0 \)), not only laser range data must be correctly projected on the NW plane (which can be done through simple trigonometric transformations), but it can even happen that the laser scanner detects objects which should not be part of the map, i.e., small removable objects, or the floor itself. Specifically, given that \(|\theta| \) and \(|\varphi|\) are below a given threshold, laser range data corresponding to walls, furniture, or even collapsed structures (in case of an earthquake) are straightforwardly integrated in a flat 2D model of the environment. If, otherwise, \(|\theta|\) or \(|\varphi|\) exceed the given threshold, the data returned by the laser are simply discarded. Projecting all environmental features on a flat 2D model of the environment yields some limitations, e.g., it needs a manual intervention to label areas of the map requiring the user to crawl or to climb an object: however, it is considered as an acceptable compromise to keep the approach simple and computationally efficient.

### 3.2. Drift reduction

The objective of Correction is to correct the estimates of \( v_x, v_y, \varphi, \theta, \psi \) provided by Prediction through observations, which partly derive from the outputs \((O.1, O.2)\) of 2D SLAM, and partly from operational hypotheses that are used as heuristics \((H.1, H.2)\) for data filtering:

- **O.1** the angle \( \psi_s \) returned by 2D SLAM can be used to update \( \dot{\psi} \);
- **O.2** the planar velocities \( v_x^s, v_y^s \) can be estimated by numerically differentiating \( x_s, y_s \) (returned by 2D SLAM) with respect to time, and used to update \( \dot{v}_x, \dot{v}_y \);
- **H.1** the expected acceleration \( E(\dot{v}_z) \) along the \( U \) axis of the navigation frame is equal to 0, with a given variance;
- **H.2** the expected neck rotational velocities \( E(\dot{\varphi}) \) and \( E(\dot{\theta}) \) are equal to 0, with a given variance.
The estimate of the user velocity provided by SLAM (O.2) proves to be particularly useful when the user is not moving: in this case, laser data are “very similar” in subsequent scans and data association is easier, therefore allowing to estimate \( v_x \approx v_y \approx 0 \). Asserting that the acceleration along the \( U \) axis of the navigation frame is equal to 0 (H.1) corresponds to saying that there cannot be a persistent movement along the vertical axis. If it were not so, we would have to deduce from (8) that the user is going off the ground. To avoid this, it is assumed that the average value is 0 with a given variance. The situation in which the user suddenly falls or starts crawling does not pose any problem from this point of view: this kind of action produces a sudden change in the acceleration along the \( U \) axis and, after that, the average value tends to 0 again (laser range data are possibly discarded during the transition). Similarly, it is assumed that the neck angular velocities are equal to 0 on average, with a given variance (H.2). If it were not so, it could be necessary to accept that, at some point, the user is walking upside down.

Let the reduced state vector \( x'_k \) include all components of the state to be corrected, i.e.,

\[
\begin{bmatrix}
  v_{x,k} \\
  v_{y,k} \\
  \varphi_k \\
  \theta_k \\
  \psi_k
\end{bmatrix}^T
\]  

(17)

The discrete state equations for \( x'_k \) (used in the prediction phase of the EKF) are not dependent on the remaining components of \( x_k \), and hence directly follow from the integration of (6), (7), (9)–(11), i.e., \( x'_k = \Gamma(x'_{k-1}, u_k) \).

The expected measurement at time step \( k \), i.e., the measurement model \( h(x'_k, u_k) \), is

\[
\begin{align*}
  h_1 &= -a_{bx,k} \sin \theta_k + a_{by,k} \sin \varphi_k \cos \theta_k + a_{bz,k} \cos \varphi_k \cos \theta_k + g, \\
  h_2 &= \omega_{bx,k} + \omega_{by,k} \sin \varphi_k \tan \theta_k + \omega_{bz,k} \cos \varphi_k \tan \theta_k, \\
  h_3 &= \omega_{bx,k} \cos \varphi_k - \omega_{bz,k} \sin \varphi_k, \\
  h_4 &= v_{x,k}, \\
  h_5 &= v_{y,k}, \\
  h_6 &= \psi_k
\end{align*}
\]

(18)–(23)

whereas, according to O.1, O.2, H.1, H.2, the actual measurement vector \( z_k \) at time step \( k \) is

\[
\begin{bmatrix}
  0 \\
  0 \\
  v_{x,k} \\
  v_{y,k} \\
  \psi_{z,k}
\end{bmatrix}^T
\]

(24)

where \( z_{k,1} \) represents the average acceleration along the \( U \) axis, \( z_{k,2} \) and \( z_{k,3} \) represent the average derivatives of the roll and the pitch, \( z_{k,4} \) and \( z_{k,5} \) represent the linear velocities along the \( N \) and \( W \) axes, \( z_{k,6} \) represents the yaw.

Strictly speaking, the components \( z_{k,1}, z_{k,2}, z_{k,3} \) in (24) are not “measurements,” since they do not depend on the state at time step \( k \). Then, concerning those components, the EKF has simply the effect of a High Pass Filter, which suppresses the low frequencies of \( \dot{v}_r, \dot{\varphi}, \dot{\theta} \) (forcing derivatives to have zero mean), and attenuates or leaves high frequencies unaltered depending on the value of the Kalman gain. This reflects on the corresponding components of the state, finally producing a dynamic behaviour which is coherent with the motion constraints that have been hypothesized in H.1, H.2.

The covariance matrices \( Q \) and \( R \), associated respectively with the noise which affects the motion model and the measurement, are evaluated through experiments. Roughly speaking, the covariance on \( z_{k,1}, z_{k,2}, z_{k,3} \) must be sufficiently “high” to take into account that the corresponding components in (24) refer to expected values instead of actual measurements. On the opposite,
the covariance on $z_{4,k}, z_{5,k}, z_{6,k}$ must be sufficiently “small” to guarantee that the components of the state observed by 2D SLAM are weighted more than those predicted on the basis of the IMU alone. Finally notice that the prediction and the measurement model depend on the control vector $u_k$: this is not a problem because, while computing the Jacobian matrices to linearise the model, the controls can be treated as constant values for every sampling instant $k$.

In principle, other estimators could be used in place of the EKF: the EKF has been chosen since it provides a framework which can be easily extended to consider other measurements, provided by other sensors or deduced from other hypotheses. Notice also that, differently from the approach adopted here, the EKF could ideally be implemented in a single logical block with the SLAM algorithm. By adopting a loosely coupled approach, it is possible to integrate in the system different SLAM algorithms with greater ease.

3.3. Observability analysis

It is necessary to verify that all the estimated components of the state, i.e., $v_x, v_y, \varphi, \theta, \psi$, are locally corrected by the EKF. To verify this, we compute the Jacobian matrices $F^\prime_k$ and $H_k$ at time step $k$, where

$$F^\prime_{k,ij} = \frac{\partial f^\prime_i}{\partial x^\prime_j}(x_k, u_k), H_{k,ij} = \frac{\partial h_i}{\partial x^\prime_j}(x_k, u_k).$$

(25)

Then, we compute the Kalman observability matrix

$$O = \left[ H_k^T \ (H_k F_k^\prime)^T \ \cdots \ (H_k F_k^\prime 4)^T \right]^T$$

(26)

and analyse how $\text{rank}(O)$ varies depending on the linearisation performed on $f^\prime(x^\prime_{k-1}, u_k)$ and $h(x^\prime_k, u_k)$. Since $O \in \mathbb{R}^{30 \times 5}$ we initially perform this analysis with the support of the MatLab Symbolic Math Toolbox. The analysis returns that, under some conditions, the system is fully observable with $\text{rank}(O) = 5$. In particular, by analysing (21)–(23), it is straightforward to verify that $v_x, v_y, \psi_k$ are always observable (regardless of the values assumed by the other components of $x^\prime_k$ and by $u_k$). Then we check under which conditions the remaining components $\varphi_k, \theta_k$ are observable.

The equations in (9)–(10) and (18)–(20) show that a reduced state vector

$$x^\prime_k = \left[ \begin{array}{c} \varphi_k \\ \theta_k \\ \end{array} \right]$$

(27)

can be decoupled from the other components of $x^\prime_k$. In fact (9)–(10) depend only on $\varphi_k, \theta_k$, and the measurement model can be analogously split into two parts, a set of equations

$$h^\prime(x^\prime_k, u_k) = \left[ \begin{array}{c} h_1 \\ h_2 \\ h_3 \\ \end{array} \right]$$

(28)

which depends only on $\varphi_k, \theta_k$ (18)–(20), and a second set of equations depending only on $v_x, \psi$, $v_y, \psi, \theta$ (21)–(23).

Considering the reduced state $x^\prime\prime_k$ and the corresponding measurement vector $h^\prime(x^\prime_k, u_k)$, let the corresponding Jacobian matrices be

$$F_{k}^{\prime\prime} = \begin{bmatrix} a & c \\ b & 0 \end{bmatrix}$$

(29)

$$H_{k}^{\prime\prime} = \begin{bmatrix} d & e \\ a & c \\ b & 0 \end{bmatrix}$$

(30)
where

\[ a = \omega_{by,k} \cos \varphi_k \tan \theta_k - \omega_{bz,k} \sin \varphi_k \tan \theta_k \]
\[ b = -\omega_{by,k} \sin \varphi_k + \omega_{bz,k} \cos \varphi_k \]
\[ c = -b(\tan^2 \theta_k + 1) \]
\[ d = a_{by,k} \cos \varphi_k \cos \theta_k - a_{bz,k} \sin \varphi_k \cos \theta_k \]
\[ e = -a_{by,k} \cos \varphi_k - a_{bz,k} \sin \varphi_k \sin \theta_k - a_{bc,k} \cos \varphi_k \sin \theta_k \]

and finally

\[ O' = \begin{bmatrix} H'F & (H'F')' \end{bmatrix}^T = \begin{bmatrix} d & a & b & ad + be & a^2 + bc & ab \\ e & c & 0 & cd & ac & bc \end{bmatrix}^T. \] (32)

The maximum possible rank of \( O' \) is 2. By computing the determinant of all \( 2 \times 2 \) submatrices of \( O' \), it can be inferred that \( O' \) loses rank if and only if \( ae = 0 \) and \( b = 0 \), since in this case \( c = 0 \) as well (31). For instance, the system is not fully observable if both the angular speeds \( \omega_{by,k} \) and \( \omega_{bz,k} \) are equal to 0, and consequently \( a = b = 0 \). In this case, which corresponds to the IMU-frame moving on a straight line with a fixed yaw and pitch, the heuristic \( \mathcal{H}_2 \) gives no contribution, and the heuristic \( \mathcal{H}_1 \), taken alone, is only able to correct a linear combination of \( \varphi, \theta \). However, it is sufficient that the user performs a curve, or temporarily rotates her neck to the left or to the right, to cause \( \omega_{bc} \neq 0 \), and to consequently correct all components of the state.

### 3.4. Calibration

A calibration procedure is available to detect the initial attitude of the IMU before operations, or whenever deemed necessary to re-calibrate the system. The IMU is required to stand still during the calibration procedure: then, an acceleration equal to 0 is expected to be measured along all axes of the navigation frame, and the measurement vector \( \mathbf{z}'_k \) at time step \( k \) can be assumed to be

\[ \mathbf{z}'_k = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T, \] (33)

where \( \mathbf{z}'_{k,1}, \mathbf{z}'_{k,2}, \mathbf{z}'_{k,3} \) represent the accelerations along \( N \), \( W \), and \( U \).

The state vector to be estimated during calibration is

\[ \mathbf{x}''_k = \begin{bmatrix} \varphi_k \\ \theta_k \\ \psi_k \end{bmatrix}^T \] (34)

and the corresponding measurement model \( \mathbf{h}''(\mathbf{x}''_k, \mathbf{u}_k) \) is

\begin{align*}
\mathbf{h}_1 &= a_{by,k} \cos \theta_k \cos \psi_k + \\
& a_{by,k}(\sin \varphi_k \sin \theta_k + \cos \varphi_k \sin \psi_k) + \\
a_{bz,k}(\sin \varphi_k \cos \psi_k + \cos \varphi_k \sin \theta_k \sin \psi_k) \\
\mathbf{h}_2 &= a_{by,k} \cos \theta_k \sin \psi_k + \\
& a_{by,k}(\cos \varphi_k \cos \psi_k + \sin \varphi_k \sin \theta_k \sin \psi_k) + \\
a_{bz,k}(\cos \varphi_k \sin \theta_k \sin \psi_k - \sin \varphi_k \cos \psi_k) \\
\mathbf{h}_3 &= -a_{by,k} \sin \theta_k + a_{bz,k} \sin \varphi_k \cos \theta_k + \\
a_{bc,k} \cos \varphi_k \cos \theta_k + g
\end{align*} (35) (36) (37)
The calibration phase corresponds to computing the Euler angles of a vector $a_b \mathbf{b}_x + a_y \mathbf{b}_y + a_z \mathbf{b}_z$, with norm $g$ by measuring its projection $\dot{v}_x$, $\dot{v}_y$, $\dot{v}_z$ on the axes of the navigation frame, and therefore this procedure necessarily yields a solution which is not unique in the sense of the Euler angle representation.

### 4. EXPERIMENTAL RESULTS

In the current set-up, the following sensors have been selected: i) the low–cost Atomic 6DOF IMU - XBee Ready, which includes a 3–axis accelerometer (MMA7260Q model) and three 1–axis gyroscopes (LISY300AL model); ii) the Hokuyo URG–04LX laser scanner, with a maximum sensing range of about 4 m, a scanning area of 240 deg with angular resolution 0.36 deg, and a scanning time of 100 ms for a full scan; iii) the more performing Hokuyo UTM-30LX Laser (used only for some experiments) with a sensing range of about 30 m, a scanning area of 270 deg with angular resolution 0.25 deg, and a scanning time of 25 ms for a full scan. As it is shown in Figure 1, the sensors are mounted on a helmet, in order to detect the obstacles that are significant for mapping, e.g., walls or furniture: obstacles which are slightly above or below the head are simply projected on the map on the basis of the current estimate of the helmet attitude (as described in Section 3.1). The sensors are chosen by considering also their weight, so that the whole helmet weighs no more than a helmet with a lamp mounted on it (the Hokuyo UTM-30LX exhibits higher performance, but it is also bigger and heavier than the Hokuyo URG–04LX, i.e., 0.23 Kg vs. 0.14 Kg).

Figure 6 shows the accuracy of the calibration procedure by placing the helmet on the floor, and by performing a rotation of about 45 deg around the $N$ axis; the final values are very close to the expected ones ($\phi = 0.785$, $\theta = 0$, $\psi = 0$ rad) with an error in the order of $10^{-2}$ rad. It is also possible to observe that, in this case, only 20 measurements are needed to have a good estimate; in all the executed tests, the number of measurements needed for calibration never exceeds 100. Considering that the IMU sampling time is 10 ms, the maximum time needed for calibration is about 1 second. This is particularly important whenever the user wishes to perform a full re-calibration of the system during operations.
In order to validate the approach during motion, experiments have been performed in two different environments: the second floor of the DIST research department and the interiors of S.I.I.T. (Integrated Intelligent Systems and Technologies), a large research facility in Genoa where the University and industrial companies run joint laboratories to cooperate on research projects related to security and automation. The approach described in the present article have been compared with HeadSLAM [18]. As discussed in Section 2, HeadSLAM uses accelerometers only to classify the motion state of the user. In order to avoid classification errors in this phase (i.e., to compare our algorithm with an “ideal implementation” of HeadSLAM), we have re-implemented HeadSLAM by manually specifying when the user is walking or not: in the former case, as prescribed by the algorithm, an average linear velocity of 1.35 m/s is assumed in the forward direction.

Figure 7 corresponds to an experiment performed at the second floor of DIST. The user starts in the corridor at the bottom of the map, performs some movements in place (to purposely produce an error in the estimated velocities and, consequently, a position drift), and finally walks straight toward the hall at the top of the map with an approximately constant velocity. The map built during operations is shown in gray and the plan of the building is superimposed in red, therefore allowing for a comparison between ground truth and actual data. IMU and laser data are acquired during operations, and then processed off-line in different ways in order to validate the solutions proposed throughout the article. The Figures correspond to: 7.a) our algorithm; 7.b) HeadSLAM; 7.c) our algorithm, without using accelerometers to estimate linear velocities, i.e., by assuming $\dot{v}_x = \dot{v}_y = \dot{v}_z = 0$ in (6)–(8); 7.d) our algorithm, without correcting $v_x, v_y$ with the output of SLAM through the EKF. In all the Figures 100 particles have been used for DP-SLAM, and data are processed with an Intel Core 2 1.66 GHz processor and 1 GB of RAM: processing time is from 1.2 to 2.2 times greater than actual operation time, and this would be a problem for a real-time approach (the work described in [18] do not exhibit real-time results as well).

The corridor in Figure 7 is a critical case: in fact, the difference between consecutive measurements of the laser, whether the user moves or not, are very similar, and therefore scan matching turns out to be not very helpful. Under these conditions the experiments in Figures 7.a and 7.b yield acceptable results, with some differences: our approach initially tends to underestimate the length of the path travelled by the user, but then it improves and finally provides a map which matches very closely the walls and furniture of the building; HeadSLAM provides a better estimate of the corridor length (since in this case the user is walking looking forward as prescribed by the model), but it behaves worse when estimating the orientation of the hall at the top of the map. Figure 7.c shows that, without the contribution of accelerometers, the SLAM algorithm tends to severely underestimate the length of the corridor: this is due to the low sensing range of the laser and the small number of features to disambiguate the user position. Finally, Figure 7.d shows the dramatic effect of position drift, when integrating accelerometers in absence of a method to correct the resulting linear velocities, thus validating the drift reduction approach based on $v_{xS}, v_{yS}$.

Figure 8 shows results when processing the same data as above, by reducing the number of particles used for SLAM to increase real-time responsiveness. The Figures correspond to: 8.a) our algorithm with 50 particles; 8.b) our algorithm with 30 particles. Under these conditions, the map tends to be less accurate: in 8.a the length of the corridor is underestimated, whereas in 8.b the orientation of walls is affected by errors. In the experiment corresponding to Figure 8.a the processing time is from 1.1 to 1.6 times greater than actual operation time. In the case in Figure 8.b the processing time is approximately equal to operation time, thus allowing for a real-time implementation of the algorithm at the price of a lower accuracy in mapping and localization.
Figure 7: Corridor of the second floor of DIST. a) our algorithm with 100 particles; b) HeadSLAM; c) our algorithm without accelerometers; d) our algorithm without EKF correction on $v_x, v_y$.
Figure 8: Corridor of the second floor of DIST. a) Our algorithm with 50 particles; a) our algorithm with 30 particles.

Figure 9: Office at the second floor of DIST. a) Our algorithm; b) HeadSLAM; c) our algorithm without EKF correction on $\phi$, $\theta$. 

16
Figure 9 shows results of a different experiment performed in the same environment. The user starts in the corridor at the bottom of the map, moves forward for a short walk, and finally enters the office on the left. The Figures correspond to: 9.a) our algorithm; 9.b) HeadSLAM; 9.c) our algorithm, but without correcting $\varphi$, $\theta$ through the EKF according to $\mathcal{H}_1$, $\mathcal{H}_2$. Under these conditions only the algorithm in Figure 9.a produces acceptable results, whereas HeadSLAM in Figure 9.b fails completely: this is likely due to the fact that the user does not look in the direction of motion when entering the office, but instead it rotates the head before the body, which produces a wrong estimate of her position and attitude. The result in Figure 9.c is not acceptable as well, thus validating the contribution of the heuristics introduced.

Figure 10.a shows the system behaviour while dealing with a short loop. The user starts in the laboratory, then she goes out through one door and gets back in through the other one, moving along a cyclic path. After that, she gets out the room and moves along the corridor. It can be noticed that, as in previous experiments, in Figure 10.a the length of the corridor tends to be underestimated in absence of features (the behaviour in presence of longer loops have not been tested because the SLAM algorithm per se is not the focus of this article). Figures 10.b and 10.c shows the behaviour of the system in different environments, i.e., a small apartment and a laboratory at the first floor of DIST.

Figure 11 shows experiments in a larger environment performed at SIIT. The correspond to: 11.a) our algorithm, using the URG–04LX laser scanner with a maximum range of 4 m; 11.b) our algorithm, using the UTM–30LX laser scanner with a maximum range of 30 m; 11.c) same dataset as 11.b, but processed with HeadSLAM; 11.d) same dataset as 11.d, but without using IMU data; 11.e) same dataset as 11.b, but ignoring all laser range measurements returning a distance higher than 4 m.

Figure 11.a shows that, in larger environments, the UTM–30LX laser scanner does not allow
Figure 11: S.I.I.T. a) our algorithm, URG–04LX laser scanner; b) our algorithm, UTM–30LX laser scanner; c) Head-SLAM, UTM–30LX laser scanner; d) our algorithm, without IMU; e) our algorithm, discarding laser range measurements > 4 m.
the system to produce accurate maps: the topology of the environment is preserved to a certain extent, but its geometry is deformed. Accurate results are achieved in Figure 11.b by using the UTM–30LX laser scanner, in which case our algorithm produces a map which matches walls and furniture very closely, whereas HeadSLAM in Figure 11.c fails almost completely. Figure 11.d confirms that the contribution of the IMU is fundamental for achieving the results above. Finally, Figure 11.e, compared with Figure 11.a, offers unexpected insights: in fact, it shows that the superior performance of the UTM–30LX laser scanner is not only due to the higher sensing range, but more likely to the higher scanning frequency, resolution, and field of view.

5. Conclusions

In this paper, a method for human navigation and mapping has been proposed which is based on wearable sensor (a 6DOF IMU and a laser scanner) to be mounted on a helmet. The system is especially targeted to search and rescue operations, where it is important to have a decision support in real-time to optimize exploration and to find evacuation routes.

The approach is different from the other approaches in the literature in that

- it does not require a model of the human gait, since all data returned by the IMU are integrated to produce an estimate of the sensor position and attitude, and merged with laser data through a standard 2D SLAM algorithm;

- it proposes an original solution, which is partly based on the output of the SLAM algorithm and partly on heuristics, to reduce the position drift which follows from the double integration of noisy data.

The approach has been tested in real-world experiments in different environments, validating the solutions proposed throughout the article and offering insights on how to improve the system. We are currently extending the approach to work in more complex scenarios involving different floors of the same building, to consider the possibility that the user switches among different walking modalities (i.e., crawling, running, lying on the ground, or climbing), and finally to rely on more sensors, or sensors with superior performance.

References