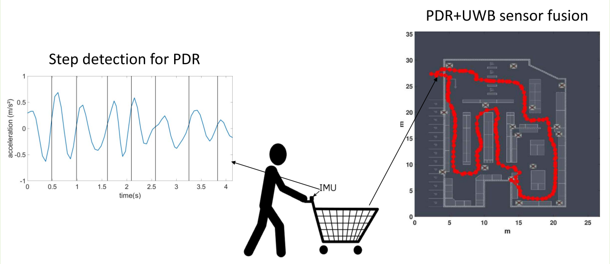
PDR/UWB Based Positioning of a Shopping Cart

Stef Vandermeeren and Heidi Steendam, *Senior Member, IEEE*

***Abstract*—In this article, we consider indoor positioning of a shopping cart in a store with a hybrid approach combining ultra wideband (UWB) and a people dead reckoning (PDR) system. While UWB can provide a very accurate positioning estimate in ideal circumstances, its accuracy reduces in non-line-of-sight situations and the update rate decreases when the number of users increases. To solve these issues, each shopping cart is not only equipped with a UWB tag, but also with an inertial measurement unit (IMU) sensor to determine the step length and heading of the user moving the shopping cart, in order to track the cart in between two UWB measurements. As the IMU is not attached to the body, the measured acceleration will be different than in other works considering PDR systems. In this article, we therefore first extract a model for the acceleration, and use the resulting model in the PDR system, where we look for the best acceleration component to track the cart. To combine the PDR and UWB information, we consider two approaches, i.e. Kalman and particle filtering, and compare both approaches. Moreover, we investigate the effect of the presence of map information of the store on the trajectory information. Our experiments show that the average positioning error using UWB only equals 62.6 cm, while the Kalman and particle filter result in an accuracy of respectively 34.1 cm and 41.3 cm, and when using map information in combination with particle filtering, the accuracy improves to 28.0 cm.**



***Index Terms*—Step counter, shopping cart, accelerometer, sensor fusion.**

I. INTRODUCTION

A PROMISING technology to provide accurate position estimates in retail environments is ultra wideband (UWB) [1]–[4], which can achieve an accuracy of the order of tens of centimetres. However, an issue with UWB-based positioning is that the accuracy will locally degrade in non-line-of-sight (NLOS) situations. In a retail environment, this can happen when the direct path between the UWB tag and anchor is obstructed by e.g. metal racks. Furthermore, due to scalability issues, the position update rate of the UWB system decreases for an increasing number of users. To reduce the effect of these drawbacks of the UWB system, we can combine the UWB system with another positioning system that does not suffer from NLOS conditions and does not depend on the number of users. A popular approach is to use the inertial measurement unit (IMU) sensors that nowadays are available in most mobile devices. This IMU consists of an accelerometer, gyroscope and sometimes a magnetometer, enabling us to track the motion and orientation of the mobile device so that it can be used e.g. in a People Dead Reckoning (PDR) application [5]–[8]. PDR is an indoor positioning technique that combines the number of detected steps with their respective length and orientation to keep track of the position of the user. A downside of this approach is that it can provide relative position updates only, and that due to errors in the estimated orientation and length of a step, the position estimate will quickly start to diverge from the true position. Combining the PDR system with the UWB system will resolve the initialization and drift problems of the PDR, as the UWB system provides regularly position updates that do not diverge in time, resulting in a system that is able to produce frequent, and in the long term stable, position updates.

In this article, we focus on tracking customers in a retail environment. As we combine the information from the UWB system with the IMU, we need a device with integrated IMU and UWB hardware. Although many customers have a mobile device with integrated IMU, integrated UWB hardware is currently not available in most smartphones. Further, privacy-issues could inhibit customers to use their own device. Hence, to track the customers, the shop must provide dedicated hardware to the customers. An option is to let the customers carry the tag during their visit, e.g. by incorporating it in a scanner to be used to scan the products. However, not all retailers provide a scanner to the customers, and moreover, the close proximity

---

Manuscript received December 9, 2020; revised February 10, 2021; accepted February 12, 2021. Date of publication February 18, 2021; date of current version April 5, 2021. This work was supported in part by the Excellence of Science (EOS) from the Belgian Research Councils Fonds Wetenschappelijk Onderzoek (FWO) and Le Fonds de la Recherche Scientifique (FNRS) under Grant 30452698 and in part by the Flemish Government (AI Research Program). The associate editor coordinating the review of this article and approving it for publication was Dr. Kyle O'Keefe. *(Corresponding author: Stef Vandermeeren.)*

The authors are with the Department of Telecommunications and Information Processing, Ghent University, 9000 Ghent, Belgium (e-mail: stef.vandermeeren@ugent.be; heidi.steendam@ugent.be).

Digital Object Identifier 10.1109/JSEN.2021.3060110

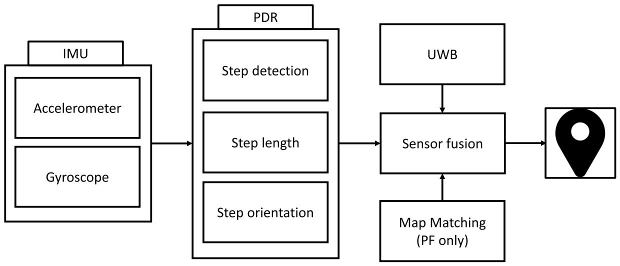


Fig. 1. System architecture.

of the human body can influence the accuracy of the UWB system [9]. Therefore, we consider another approach, where we incorporate the sensors in the handle of a shopping cart. When the IMU is fixed to the human body, we can directly detect the steps of a user by measuring the periodic pattern in the acceleration in the direction of gravity. However, when the IMU is attached to a shopping cart, the change in acceleration in the direction of gravity is almost absent, implying detecting the steps of a user will not be straightforward. In this work, we propose an algorithm that detects the steps of a customer, using a shopping-cart-mounted accelerometer. To this end, we first propose a model for the acceleration that is measured by the shopping-cart-mounted IMU, which is then used to derive the step detection algorithm. Further, we apply this step detector in a PDR system and implement a sensor fusion algorithm to combine the information from the PDR and UWB system, enabling us to accurately track a shopping cart in retail environments.

The rest of the article is organised as follows. In Section II, we introduce the system architecture of our positioning system. The PDR system consists of three building blocks, i.e. the step detection, the step length estimator and the step orientation estimator. In this article, we will mainly focus on the step detection algorithm, and briefly discuss the other two building blocks. In Section III, we explain the limitations of state-of-the-art step detection algorithms when the IMU is attached to the handle of a shopping cart. Further, we present our model for a shopping-cart-mounted accelerometer and propose a step detection algorithm based on the measurements. In Sections IV and V, we describe the two remaining building blocks of the PDR system, i.e. to respectively determine the step length and heading. Next, in Section VI, we elaborate on the sensor fusion algorithm that combines the information of the PDR and UWB systems. In the results section (Section VII), we evaluate the performance of the resulting step detection algorithm and the sensor fusion algorithms and finally, conclusions are given in Section VIII.

II. SYSTEM ARCHITECTURE

Before we go into the details of the different building blocks, we first give in this section an overview of the system architecture of our positioning system for a shopping-cart-mounted IMU. The high-level block diagram for the chosen system architecture is shown in Figure 1. The IMU collects the measurements from the accelerometer and gyroscope, and feeds them to the PDR block. The PDR updates the position of the shopping cart by extracting the steps, step length and heading of the user of the cart from the measured data. To correct the drift of the resulting position estimate, caused by the noisy IMU measurements, and obtain more reliable position estimates, we apply a sensor fusion algorithm to combine the position estimates from the PDR block with the measurements of the UWB system. In the remainder of this section, we give a more detailed description of the PDR, UWB and sensor fusion blocks.

A. PDR

A PDR system keeps track of a user's position in a relative manner. Given an initial position p_0, the PDR iteratively updates the position $\boldsymbol{p}_k = [p_{x,k}, p_{y,k}]$ of a user as follows:

$$\begin{aligned} p_{x,k} &= p_{x,k-1} + L_k \cos(\Psi_k) \\ p_{y,k} &= p_{y,k-1} + L_k \sin(\Psi_k), \end{aligned} \tag{1}$$

where $p_{x,k}$ and $p_{y,k}$ are respectively the x and y coordinate of the user at the end of the k^{th} step, and L_k and Ψ_k are respectively the length and orientation of the k^{th} step. Hence, to be able to update the position, the PDR must accomplish the following three subtasks:

- Step detection, i.e. detect the start and end of step k
- Step length estimation, i.e. determine L_k
- Step heading estimation, i.e. determine Ψ_k

The PDR in our algorithm receives input from the IMU, which measures the acceleration and rotation of an object, from which we can extract the needed information. Note that, although the update expression (1) does not depend on the location of the IMU, i.e. body-fixed or shopping-cart-mounted, the measurements made with the IMU will depend on this location, implying the algorithms to determine the steps, step length and orientation will differ depending on the location of the IMU. Many works on PDR research consider an IMU that is attached to the body, e.g. it is head-mounted [5], foot-mounted [6], waist-mounted [7] or handheld [8]. Less research was done on PDR systems using IMUs attached on a cart [10]. This can be explained as in contrast to body-fixed IMUs, steps are not directly visible in the accelerometer data of cart-mounted IMUs. To solve this issue, the authors of [10] extract from the accelerometer data of a cart-mounted IMU an effective motion frequency, from which they determine the time interval between successive steps. The authors then augment the step counter by one step when the time since the last step is larger than this time interval. Further, the authors also estimate the heading of the cart. Unfortunately, the authors implement their step detector in their PDR algorithm without analysing the step detector accuracy in detail.

The benefit of a PDR system is that no infrastructure is required in the environment, i.e. each user only needs to be equipped with a single IMU. However, the PDR approach also has its limitations. First, the PDR system needs an initial estimate of the position and heading, which are hard to determine without the use of another positioning technique. Secondly, errors in the estimation of the step length L_k and heading ψ_k will eventually lead to a large drift in the estimated position. As a consequence, the PDR system can only provide very accurate incremental position updates for a short period.

In Sections III-V, we will discuss the implementation of the different components of our PDR in detail. The accuracy of the step detector and the PDR algorithm will be analysed in the results section, i.e. Section VII.

B. UWB

To overcome the limitations of the PDR approach, we combine in the proposed positioning system the PDR system with a UWB system. This UWB system provides regular updates of the estimated position and promises an accuracy of the order of tens of centimetres. In contrast to the PDR approach, this system requires a dedicated infrastructure to estimate the position. The infrastructure consists of a number of anchors that must be placed at fixed and known positions in the environment. The carts, that are moving in the environment, are equipped with a tag. This tag communicates with one anchor at a time by transmitting a signal, and from the time delay of the line-of-sight (LOS) component of the received signal we can extract the distance between the tag and the anchor. Based on the distances to the different anchors, the position of the tag can be determined. As the distance is obtained from the LOS component, it is clear that an incorrect estimate of the distance to an anchor will be obtained if the direct path between the tag and an anchor is obstructed by e.g. a metal rack. An incorrectly estimated distance will in turn have an influence on the UWB position estimate, implying the accuracy of the UWB system may degrade locally due to non-line-of-sight (NLOS) conditions. Moreover, as each tag needs to range with all anchors, and at each time instant only one tag is allowed to communicate with a single anchor to avoid interference, the update rate for each user can be low, implying in between two updates, no position information is available.

C. Sensor Fusion

From the previous sections, it is clear that the PDR and UWB systems are complimentary, and that each system can help the other where it fails. Hence, combining the information of both systems can result in a hybrid system that provides frequent, and in the long term accurate, position estimates. To merge the information of the PDR and UWB systems, we use a sensor fusion algorithm in this work. In this fusion algorithm, we combine the position information from the PDR and UWB systems, to obtain a position estimate with a lower uncertainty than would be possible when the systems are used separately. Due to the non-linear nature of the positioning problem, we cannot employ the commonly used Kalman filter approach. However, the extended Kalman filter (EKF) is able to deal with non-linear dynamical systems. Similarly as the Kalman filter, the EKF consists of two phases, i.e. the prediction phase, where the new state is predicted based on the old state perturbed with Gaussian noise, and the update phase, where the predicted state is updated based on the measurements. The prediction and update are based on the non-linear relations f_k and h_k:

$$\boldsymbol{x}_k = f_k(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) \tag{2}$$

$$z_k = h_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{3}$$

where $\boldsymbol{x_k}$ is the state at timestep k, $\boldsymbol{z}_k$ is a vector containing the sensor measurements, and $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are the Gaussian noise terms in respectively the prediction and update phase. To solve the problem, the EKF linearizes the non-linear functions f_k and h_k. While this EKF approach is able to tackle non-linear problems, the performance is not always satisfactory due to the assumption that the noise term is Gaussian, which is not always satisfied in practical systems. Another approach that can be used for non-linear dynamical systems experiencing non-Gaussian noise is the Particle Filter (PF) approach, which uses a Sequential Monte Carlo method. In this approach, the particle filter starts with a large number of particles, where each particle corresponds to a possible state. The PF approach uses the same equations as EKF (note that now the noise components $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ no longer are restricted to be Gaussian), i.e. (2) predicts the next state for each particle, and (3) is used to determine a score for each particle that indicates how well the state of the particle matches with the measurements. This more general PF approach generally results in better performance, but comes at the cost of a higher computational complexity compared to EKF. In this article, we will consider both EKF and PF to estimate the state, i.e. the position of the cart, and compare the performance and complexity of both approaches. Further, the PF approach enables us to easily include map data of the environment in the fusion algorithm. Hence, we also consider a map matching algorithm to further improve the performance of our positioning system.

III. STEP DETECTION

The first step in the PDR system is the detection of the steps of the user, i.e. the determination of the number of steps and the start and end of each step. In this section, we discuss how this information can be extracted from the acceleration measured with an IMU mounted on a shopping cart. For comparison, we first discuss the case of a body-fixed IMU, which is often considered in the literature. When the IMU is fixed on the body of a user, the steps of the user will mainly be visible in the vertical component of the acceleration, due to the up and down movement of the body during taking steps, and the impact of the foot on the ground. Further, the acceleration will also reflect the repetitive nature of taking steps. As a consequence, most works in the literature operate in the time domain to detect the peaks and/or zero crossings in (mainly) the vertical component of the measured acceleration [11]–[15]. In contrast to the body-fixed IMU, the vertical movement of a shopping cart-mounted IMU will be limited, implying that the vertical component of the acceleration might be less suited to extract the steps from the measurements.

To determine the best approach to estimate the steps from the measured acceleration, we therefore first introduce a novel mathematical model for the acceleration of a cart-mounted IMU. We start by defining the coordinate system relative to the shopping cart, and assume that the x-axis of the IMU is aligned with the direction in which the shopping cart is heading, while the z-axis is aligned with gravity. The y-axis then corresponds to the direction in the horizontal plane perpendicular to the moving direction, as illustrated in Figure 2.

Fig. 2. Coordinate system shopping cart.

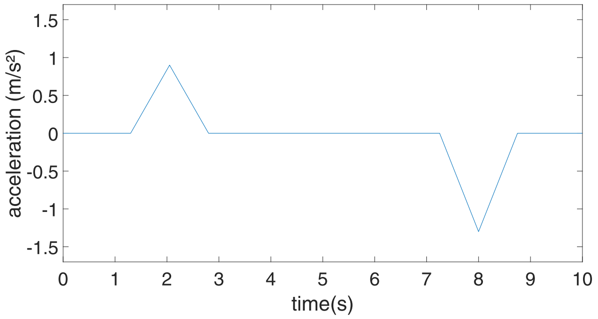


Fig. 3. Acceleration $a_{cart,x}$ from starting/stopping the shopping cart.

In the mathematical model, we decompose the acceleration into the following contributions:

- $\boldsymbol{a}_{steps}$, i.e. the acceleration due to taking steps that can be measured by the shopping-cart-mounted IMU. As taking steps is a repetitive activity, we model this component as a sinusoid with a frequency equal to the stepping frequency.
- $\boldsymbol{a}_{cart}$, i.e. the acceleration that is necessary to accelerate and slow down the shopping cart when the user starts or stops walking. This component mainly is present in the movement direction of the shopping cart, i.e. in the x-direction. We model this component as a set of triangular pulses, as illustrated in Figure 3. More specifically, when the user accelerates the cart from a rest state, this acceleration component is modeled as a positive triangular pulse and when the user stops the cart, as a negative triangular pulse.
- $\boldsymbol{a}_{turn}$, i.e. the acceleration caused by taking turns (centripetal force). This component mainly is present in the y-direction and results in a bias in the y-component that is correlated with the angular rate measured with the gyroscope in the z-axis.
- $\boldsymbol{g}$, i.e. the measured acceleration due to gravity. Due to the selection of the coordinate system, this component is present in the z-direction only.
- $\boldsymbol{a}_n$, i.e. the noise of the measured acceleration. This component also comprises the acceleration due to e.g. vibrations of the shopping cart.

Hence, the acceleration in the x-, y- and z-axis is approximated by:

$$\begin{aligned} a_x &= a_{steps,x} + a_{cart,x} + a_{n,x} \\ a_y &= a_{steps,y} + a_{turn,y} + a_{n,y} \\ a_z &= a_{steps,z} + g_z + a_{n,z} \end{aligned} \tag{4}$$

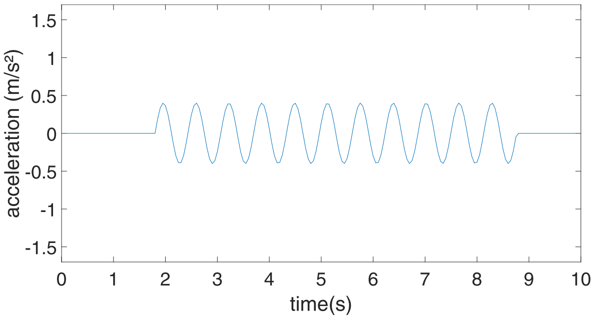


Fig. 4. Acceleration $a_{steps,x}$ related to taking 11 steps.

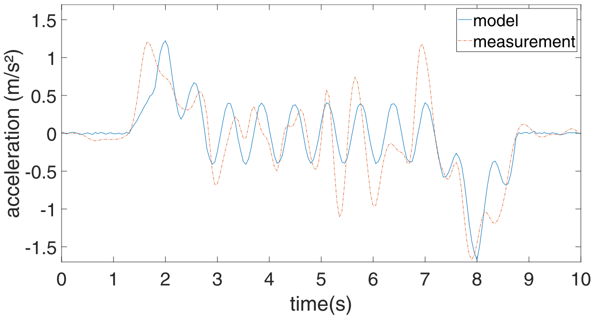


Fig. 5. Comparison of the modeled and low-pass filtered measured acceleration in the x-direction.

The noise contribution is modeled as a zero-mean random variable. In the following, we assume that the acceleration is filtered with a low-pass filter to remove the noise contribution, and neglect the presence of the noise in the example.

To illustrate the effectiveness of the model, we compare the low-pass filtered acceleration measured in the x-direction with the above model, for a user taking 11 steps in a straight line. The x-component of our model consists of two contributions: $a_{cart,x}$, shown in Figure 3 and $a_{steps,x}$, illustrated in Figure 4. In Figure 5, we show $a_x = a_{cart,x} + a_{steps,x}$ along with the low-pass filtered measured acceleration. The figure demonstrates that the proposed model matches fairly well with the measured acceleration. Comparing the measured acceleration with the model, we observe some differences. First, we notice that the amplitude of the sinusoidal component in the measured acceleration is not constant. A similar effect can be observed in body-fixed IMUs, and can be explained as each step we take is unique and has slightly different amplitude. A second observation is that the measured acceleration is not always centered around zero, i.e. there is a slowly time-varying bias. In our model, we assume the cart is moving at a constant speed between the triangles corresponding to starting and stopping the cart. However, in reality, the walking speed of a user is not constant, but subject to slowly time-varying variations. Further, the cart is pushed by the hands of the user, and the user does not have a rigid body. Hence, the pose of the body, arms and hands may change during the pushing of the cart, as the user slightly stretches and bends the arms. Consequently, the arms will behave like a sort of spring, resulting in variations in the acceleration that are visible in the x-component. It is clear from the above example that the x-component of the acceleration can be used to extract

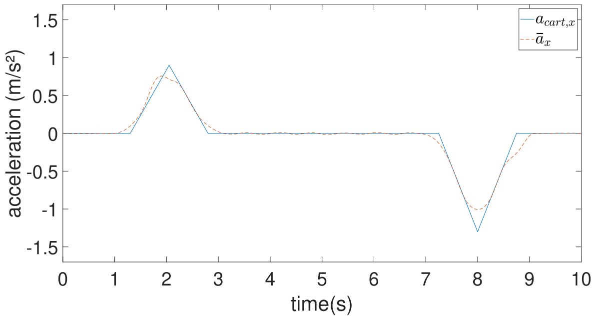


Fig. 6. Comparison of the modeled acceleration of the shopping cart and the moving average ($N = 13$) of the total modeled acceleration.

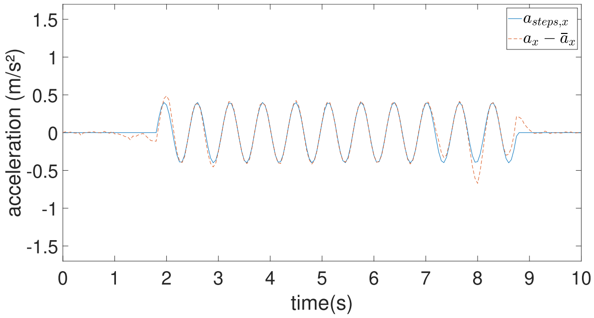


Fig. 7. Comparison of the modeled acceleration related to taking steps and the total modeled acceleration minus the moving average.

the steps of the user. However, the presence of $a_{cart,x}$ hampers the detection of the steps, as the triangular pulse from (de)accelerating the shopping cart is spread over several steps. To remove the contribution $a_{cart,x}$ from the acceleration a_x, we filter the acceleration a_x with a moving average filter with fixed length N corresponding to approximately the duration of one step, where N determines the number of samples over which we average the acceleration. The value of N depends on the sample frequency f_s used in the measurements. Given the step frequency f_{step} and the sample frequency f_s, we expect that a good value for the length of the moving average filter is approximately $N \approx \frac{f_s}{f_{step}}$. Typically, the step frequency ranges approximately from 1.5Hz to 2Hz and for our measurements, the sample frequency was approximately 20Hz so $N \in [10, 14]$ samples. In the results section, we will determine the value for N that results in the best performance on our collected accelerometer data. In Figure 6, we show the acceleration $a_{cart,x}$ from the shopping cart and the moving average $\bar{a}_x$ of a_x (4), where N was set equal to 13 samples. We can clearly see that $\bar{a}_x$ matches well with the acceleration $a_{cart,x}$ of the shopping cart. Figure 7, showing the difference between the total acceleration and the moving average, i.e. $a_x - \bar{a}_x$, together with the acceleration $a_{step,x}$, demonstrates that we can easily extract the steps from $a_x - \bar{a}_x$ by using e.g. zero-crossing or peak detection. In the evaluation of the effect of the moving average filter, we did not take into account the effect of the biases present in the measured acceleration. However, the moving average filter will also remove to a large extent the slowly time-varying biases present in the measured acceleration.

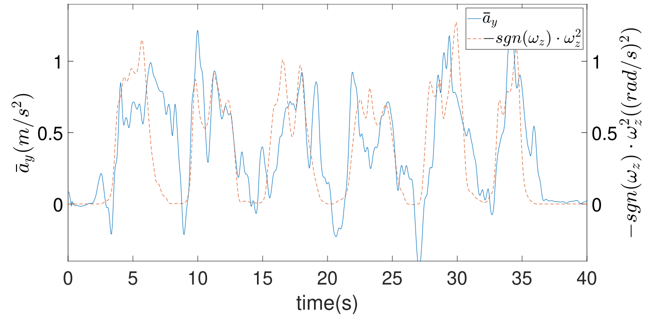


Fig. 8. Comparison of the moving average $\bar{a}_y$ of the low-pass filtered measured acceleration in the y-direction and $-sgn(\omega_z) \cdot \omega_z^2$, with ω_z the low-pass filtered angular rate measured by the gyroscope in the z-direction.

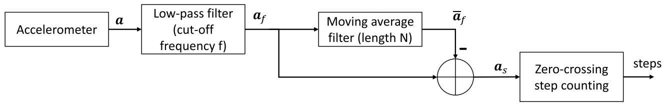


Fig. 9. Block diagram step detection algorithm.

A similar approach can be used to extract the acceleration related to taking steps from the y- and z-component of the acceleration. In our model for the y-component of the acceleration, a term related to taking turns $a_{turn,y}$ was included. Assuming that during a turn the IMU follows a perfectly circular path and that the x-axis of the IMU is tangential to the path, we can write $a_{turn,y}$ as $a_{turn,y} = -r \cdot sgn(\omega_z) \cdot \omega_z^2$, where r is the radius of the turn, ω_z is the angular rate in the z-direction measured by the gyroscope, and $sgn(\cdot)$ is the sign function. As a turn is spread over several steps, we can extract the contribution of $a_{turn,y}$ from a_y by filtering the measured acceleration with a moving average filter with length N, and remove the contribution to a large extent from the measured acceleration. In Figure 8, we show the moving average of the low-pass filtered measured acceleration $\bar{a}_y$ in the y-direction together with $-sgn(\omega_z) \cdot \omega_z^2$, where ω_z is the low-pass filtered angular rate, measured with the gyroscope, for a user that took 50 steps. From this figure, it is clear that $\bar{a}_y$ and $-sgn(\omega_z) \cdot \omega_z^2$ indeed are correlated. However, some differences are also visible due to the assumptions we made on the path we follow during the turn. In reality, the IMU does not take a perfect circular turn, implying writing $a_{turn,y}$ as $-r \cdot sgn(\omega_z) \cdot \omega_z^2$ is only a rough approximation of the acceleration the IMU measures during a turn. Similarly as in the x-direction, we also noticed a time-varying bias in the measured acceleration in the y-direction of the measured acceleration, which will also be removed by the moving average filter. According to the model for the z-component, the acceleration should contain a more or less constant bias of 1g due to gravity. In our measurements however, we also notice, as with the x- and y-component, a time-varying bias, which can, just as the constant bias, be removed with the moving average filter. By subtracting the moving average from the low-pass filtered measured acceleration in either the x-, y- or z-component, we will obtain an acceleration component related to the acceleration $\boldsymbol{a}_{steps}$ from taking steps.

In the remainder of this section, we propose the step detection algorithm that is based on the above described model for the acceleration. The block diagram of the step detection algorithm is shown in Figure 9. First, we collect the three outputs of the shopping-cart-mounted accelerometer, corresponding to the acceleration $\boldsymbol{a} = (a_x, a_y, a_z)$ in the x, y and z-direction. As most of the body-fixed step detectors use the magnitude of the acceleration, we also determine the magnitude $|\boldsymbol{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$ of the raw acceleration and test it in our step detection algorithm. Secondly, we use a low-pass filter to reduce the effect of the noise on the measured acceleration. In this work, we use a 3^{rd} order low-pass Butterworth filter with cut-off frequency f to obtain the smoothed signals $a_{f,\alpha} = LP(a_\alpha)$ and $|\boldsymbol{a}|_f = LP(|\boldsymbol{a}|)$, where $LP(h)$ applies the low-pass filter to signal h, and $\alpha \in \{x, y, z\}$. Next, we feed the smoothed signals to a moving average filter with length N. Subtracting the moving average $\overline{\boldsymbol{a}}_f = (\overline{a}_{f,x}, \overline{a}_{f,y}, \overline{a}_{f,z})$ or $\overline{|\boldsymbol{a}|}_f$ from the low-pass filtered measured acceleration approximately eliminates the acceleration components that are not related to taking steps, i.e $\boldsymbol{a}_s = \boldsymbol{a}_f - \overline{\boldsymbol{a}}_f \approx \boldsymbol{a}_{steps}$ or $|\boldsymbol{a}|_s = |\boldsymbol{a}|_f - \overline{|\boldsymbol{a}|}_f$. Finally, we apply $a_{s,\alpha}$ or $|\boldsymbol{a}|_s$ to a zero-crossing step counting algorithm with timing and amplitude constraints.

For convenience, we use a_s for the remainder of this section to signify $a_{s,\alpha}$ or $|\boldsymbol{a}|_s$. In this step counting algorithm, we first determine the possible start and end points of the steps in signal a_s. To this end, we iterate through the signal and look for time instants where a_s crosses threshold $th = 0g$ with a positive slope, i.e. $a_{s,i} > th$ and $a_{s,i-1} < th$. These instants are considered as possible start and/or end points of a step. The first time a sample $a_{s,i}$ meets this condition, it can only be the start of the first step. In that case, we set the start of the step under consideration equal to $i_{start} = i$. Otherwise, the sample $a_{s,i}$ can indicate the end of the current step that started at sample i_{start} (and the corresponding start of the next step). To validate if $a_{s,i}$ is indeed the end of the current step and limit the number of falsely detected steps, we determine the maximum M and minimum m of the the acceleration a_s in the interval $[i_{start}, i]$:

$$\begin{aligned} M &= \max_{l \in [i_{start}, i]} (a_{s,l}) \\ m &= \min_{l \in [i_{start}, i]} (a_{s,l}), \end{aligned} \tag{5}$$

and check that 1) M is larger than a threshold M_{th} 2) that the difference between the maximum and minimum, i.e. $M - m$, is larger than $(M - m)_{th}$, and 3) that the duration between the two zero crossings, i.e. $i - i_{start}$, lays in the interval $[i_{min}, i_{max}]$. The first two constraints prevent that acceleration patterns with only a small variation in the acceleration would lead to a step detection, while the third constraint makes sure that only patterns with a time duration that could correspond to a step are considered. If we find a positive-slope zero crossing that satisfies the above three conditions, we conclude that a valid step occurred: the step count is increased by one, we set the sample index of the found end of the step as the start i_{start} of the next step, and start the search procedure again to find the end of the next step. If no valid end of a step is found, i.e if the two first conditions are not satisfied within the interval $[i_{min}, i_{max}]$, we start looking for a new start of a step starting from the first zero crossing after time index i_{start} and start looking for a new positive zero-crossing from this point. In the results section, we will use a grid search to find the values of the multiple parameters in our algorithm that result in the step detector with the best performance. Additionally, we also compare the performance of our algorithm on the different axes and magnitude of the acceleration, i.e. we compare the performance for $a_{s,\alpha}$, with $\alpha \in \{x, y, z\}$ and $|\boldsymbol{a}|_s$.

IV. STEP LENGTH

Several approaches to estimate the step length exist. On the one hand, we can estimate the step length by finding a parametric model that maps different variables (e.g. the maximal acceleration in a step) to the step length of that step [16]–[19]. A drawback of this approach is that these models also contain some parameters that need to be tuned to a specific user. As we do not have ground truth for the step lengths, we cannot use this parametric approach. On the other hand, Kalman filters can be used to estimate the step length as in [20], [21]. A downside of this approach, however, is that it only performs well if the IMU is attached to the foot. As in this work, the IMU is shopping-cart-mounted, we also cannot use Kalman filters to estimate the step length. Hence, for simplicity, in this article, we model the step length in our PDR system as a constant, i.e. initially $L = 0.7m$. Although this a very rough approximation, it will not jeopardize the proper action of our positioning system as we later will combine the PDR system with UWB measurements in an extended Kalman filter (EKF) or particle filter (PF), which both allow us to update the step length based on the UWB measurements.

V. HEADING ESTIMATION

To unambiguously determine the orientation of the shopping cart, we first need to define the different coordinate frames that will be used in this article. The navigation frame is the frame in which we determine the position and orientation. This frame is fixed and can e.g. be aligned with a room. In this article, we align the navigation frame with the coordinate frame of the UWB system, i.e. the x- and y-axis lie in the horizontal plane and the z-axis is aligned with gravity. The body frame corresponds with the frame in which the IMU measures the acceleration and angular rate of the shopping cart. As the shopping cart moves relative to the navigation frame, the orientation between the body and navigation frame will continuously change. To track the orientation of the body frame relative to the navigation frame, we will use an extended Kalman filter based on [22] to determine the orientation in quaternion form from the IMU data. Quaternions are a four-dimensional representation of the orientation that have as an advantage over the more common Euler angles that they do not suffer from the well-known gimbal lock. In Figure 10, we show the high-level block diagram for the heading estimation.

First, we use the accelerometer to determine the initial orientation. Next, we use the gyroscope to continuously update the current orientation. To decrease the error of the orientation estimation, we use the measurements of the accelerometer and gyroscope while the shopping cart is static, e.g. when the user stops to take an item from the shopping racks, which only happens occasionally. In the remainder of this section, we will discuss the heading estimation in more detail.

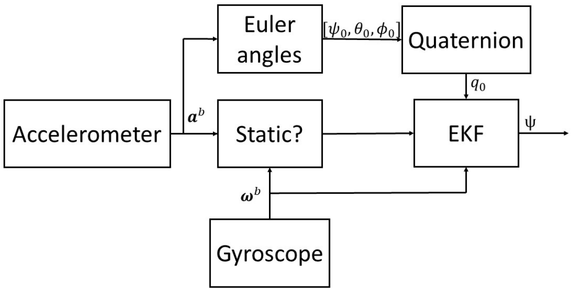


Fig. 10. Block diagram for heading estimation.

In this article, we write the state of our EKF at time index i as $\boldsymbol{x}_i = [\boldsymbol{q}_i^{bn}, \boldsymbol{b}_{\omega^b,i}, \boldsymbol{b}_{a^b,i}]^T$, where $\boldsymbol{q}_i^{bn} = [q_{i,0}^{bn}, q_{i,1}^{bn}, q_{i,2}^{bn}, q_{i,3}^{bn}]$ is the quaternion that describes the rotation from the navigation frame to the body frame, $\boldsymbol{b}_{\omega^b,i}$ and $\boldsymbol{b}_{a^b,i}$ are respectively the bias of the gyroscope and accelerometer both in the body frame, and $\square^T$ denotes the transpose operation. At the start of the algorithm, we assume that the shopping cart is static and that the initial heading is known, e.g. because the carts are lined up at the entrance, enabling us to easily extract the orientation of the IMU in Euler angle format from the measured acceleration. The initial (z,y,x)-Euler angles $[\psi_0, \theta_0, \phi_0]$ can be determined with

$$\begin{aligned}\psi_0 &= \tilde{\psi}_0\\ \theta_0 &= \arctan\Big(\frac{-a_{x,0}^b}{\sqrt{(a_{y,0}^{b\,2} + a_{z,0}^{b\,2})}}\Big)\\ \phi_0 &= \arctan\Big(\frac{a_{y,0}^b}{a_{z,0}^b}\Big),\end{aligned} \tag{6}$$

where $\boldsymbol{a}_x^b$, $\boldsymbol{a}_y^b$ and $\boldsymbol{a}_z^b$ are respectively the x-, y- and z-component of the measured acceleration in the body frame, and $\tilde{\psi}_0$ is the known initial heading. When the initial heading is not known, we can set $\tilde{\psi}_0$ to a random value, e.g. $\tilde{\psi}_0 = 0°$, as the state of the fusion algorithm of Section VI, amongst others, also contains the heading error. Hence, if we choose a high initial variance for the heading error, the fusion algorithm is able to estimate this error within a few UWB updates. However, as our EKF needs the orientation in quaternion form, we still need to transform the Euler angles into a quaternion with

$$\begin{aligned}q_{0,0}^{bn} &= \cos(\frac{\phi_0}{2})\cos(\frac{\theta_0}{2})\cos(\frac{\psi_0}{2}) + \sin(\frac{\phi_0}{2})\sin(\frac{\theta_0}{2})\sin(\frac{\psi_0}{2})\\ q_{0,1}^{bn} &= -\sin(\frac{\phi_0}{2})\cos(\frac{\theta_0}{2})\cos(\frac{\psi_0}{2}) + \cos(\frac{\phi_0}{2})\sin(\frac{\theta_0}{2})\sin(\frac{\psi_0}{2})\\ q_{0,2}^{bn} &= -\cos(\frac{\phi_0}{2})\sin(\frac{\theta_0}{2})\cos(\frac{\psi_0}{2}) - \sin(\frac{\phi_0}{2})\cos(\frac{\theta_0}{2})\sin(\frac{\psi_0}{2})\\ q_{0,3}^{bn} &= -\cos(\frac{\phi_0}{2})\cos(\frac{\theta_0}{2})\sin(\frac{\psi_0}{2}) + \sin(\frac{\phi_0}{2})\sin(\frac{\theta_0}{2})\cos(\frac{\psi_0}{2}).\end{aligned} \tag{7}$$

For the accelerometer and gyroscope, we assume that the initial bias is equal to zero, i.e. $\boldsymbol{b}_{\omega^b,0} = \boldsymbol{b}_{a^b,0} = [0, 0, 0]$.

In the prediction step of the Kalman filter, we predict the state $\hat{\boldsymbol{x}}_i = [\hat{\boldsymbol{q}}_i^{bn}, \hat{\boldsymbol{b}}_{\omega^b,i}, \hat{\boldsymbol{b}}_{a^b,i}]^T$ at time index i as a function of the state at time index $i-1$ using:

$$\begin{aligned}\hat{\boldsymbol{q}}_i^{bn} &= \exp\Big(\frac{-\Delta t_i}{2}(\boldsymbol{\omega}_i^b - \boldsymbol{b}_{\omega^b,i-1})\Big) \odot \boldsymbol{q}_{i-1}^{bn}\\ \hat{\boldsymbol{b}}_{\omega^b,i} &= \boldsymbol{b}_{\omega^b,i-1}\\ \hat{\boldsymbol{b}}_{a^b,i} &= \boldsymbol{b}_{a^b,i-1},\end{aligned} \tag{8}$$

where $\hat{\boldsymbol{q}}_i^{bn}$, $\hat{\boldsymbol{b}}_{\omega^b,i}$ and $\hat{\boldsymbol{b}}_{a^b,i}$ are respectively the predicted quaternion, accelerometer bias and gyroscope bias. Further, Δt_i corresponds to the time interval between time index i and $i-1$, $\boldsymbol{\omega}_i^b$ is the angular rate at time index i in the body frame, $\odot$ is the quaternion product, and $\exp(\cdot)$ the quaternion exponential defined as

$$\exp(\boldsymbol{v}) = \left(\cos(\|\boldsymbol{v}\|), \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\sin(\|\boldsymbol{v}\|)\right) \tag{9}$$

In the update step of the EKF, we first determine if the shopping cart is static, i.e. if the shopping cart is not moving. If so, the accelerometer will only measure the acceleration of the gravitational force and a bias, and the gyroscope will only measure the bias of the gyroscope. Hence, we can write this as

$$\begin{aligned}\boldsymbol{a}_i^b &= \boldsymbol{R}^{\hat{q}_i^{bn}}\left[0,\ 0,\ g\right]^T + \hat{\boldsymbol{b}}_{a^b,i}\\ \boldsymbol{\omega}_i^b &= \hat{\boldsymbol{b}}_{\omega^b,i},\end{aligned} \tag{10}$$

where $\boldsymbol{a}_i^b$ and $\boldsymbol{\omega}_i^b$ are respectively the measured acceleration and angular rate in the body frame, $\boldsymbol{R}^{q_i^{bn}}$, given by

$$\boldsymbol{R}^q = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2)\\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1)\\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix} \tag{11}$$

is the rotation matrix that performs the rotation from the navigation frame to the body frame and $g = 9.81 m/s^2$ is the gravitational acceleration. Using these measurements, the EKF updates the predicted state $\hat{\boldsymbol{x}}_i$ into the updated state $\boldsymbol{x}_i$. Finally, to find the heading ψ_i at each time index i, we use the updated quaternion $\boldsymbol{q}_i^{bn}$ to transform the angular rate $\boldsymbol{\omega}_i^b$ in the body frame into the angular rate $\boldsymbol{\omega}_i^n$ in the navigation frame. As the z-component of $\boldsymbol{\omega}_i^n$ determines the change of the heading, we can use this angular rate to iteratively determine the current heading ψ_i:

$$\begin{aligned}\psi_i &= \tilde{\psi}_0 + \sum_{j=1}^{i} \Delta t_j [\boldsymbol{R}^{q_j^{bn}}(:,3)]^T \boldsymbol{\omega}_j^b = \tilde{\psi}_0 + \sum_{j=1}^{i} \Delta t_j \boldsymbol{\omega}_{j,z}^n\\ &= \psi_{i-1} + \Delta t_i \boldsymbol{\omega}_{i,z}^n\end{aligned} \tag{12}$$

where $\boldsymbol{R}^{q_j^{bn}}(:,3)$ corresponds to the third column of $\boldsymbol{R}^{q_j^{bn}}$ and $\boldsymbol{R}^T$ denotes the transpose of matrix $\boldsymbol{R}$. Note that (12) assumes that $\psi_0 = \tilde{\psi}_0$.

VI. SENSOR FUSION ALGORITHM

In this section, we describe the algorithm that fuses the data from the PDR and UWB system to combine the benefits and limit the drawbacks of each system. To fuse the UWB and PDR data, we consider and compare an extended Kalman

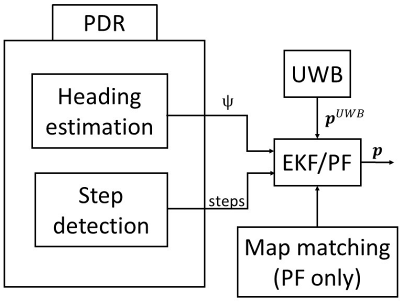


Fig. 11. Block diagram for sensor fusion.

filter (EKF) and a particle filter (PF). In Figure 11, we show the block diagram of the sensor fusion implementation.

First, we use the step detection from Section III to detect if a user took a step. If so, we determine the heading of the step with the heading estimation algorithm from Section V. The fusion algorithm can then use this information to predict the new position of the user. Next, at time instants where new UWB measurements are available, we use the UWB information to improve the accuracy of predicted position and finally, we use the floor map to correct the position if the position estimate is not possible, e.g. the position estimate would require that the user passed through a wall. In the remainder of this section, we describe each part of the fusion algorithm in more detail.

To model the system, we use the state vector $\boldsymbol{x}_k = [\boldsymbol{p}_k, L_k, \Delta\Psi_k]^T$, where $\boldsymbol{p}_k = [p_{x,k}, p_{y,k}]$ corresponds to the x and y coordinate of the shopping cart after step k, and L_k and $\Delta\Psi_k$ are respectively the step length and heading error of the k^{th} step. To initialise the sensor fusion algorithm (either the EKF or PF), we calculate the initial state using

$$\begin{aligned} p_{x,0} &= p^{UWB}_{x,0} \\ p_{y,0} &= p^{UWB}_{y,0} \\ \Delta\Psi_0 &= 0 \\ L_0 &= 0.7m \end{aligned} \tag{13}$$

where we use a UWB measurement to determine the initial position and assume that the initial heading error and step length are respectively 0 and $0.7m$.

As already mentioned, both the extended Kalman and particle filter consist of two phases, i.e. the prediction and the update phase. In the prediction phase, we use the data from the PDR system to predict the next state $\hat{\boldsymbol{x}}_k = [\hat{\boldsymbol{p}}_k, \hat{L}_k, \Delta\hat{\Psi}_k]^T$ using

$$\hat{p}_{x,k} = p_{x,k-1} + L_{k-1}\cos(\Psi_k + \Delta\Psi_{k-1}) \tag{14}$$

$$\hat{p}_{y,k} = p_{y,k-1} + L_{k-1}\sin(\Psi_k + \Delta\Psi_{k-1}) \tag{15}$$

$$\Delta\hat{\Psi}_k = \Delta\Psi_{k-1} \tag{16}$$

$$\hat{L}_k = L_{k-1} \tag{17}$$

where Ψ_k is the heading of the k^{th} step, found by averaging the heading reported by the heading estimator of the PDR system (Section V) over the entire duration of the step. Hence, we predict the new position of the user based on the previous position estimate, the estimated heading of the step and current estimate of the step length and heading error in the state. For the step length and heading error, we assume no change.

In the update phase, we use the information of other sensors to further improve the accuracy of the estimated state. To this end, we check if a UWB position update $\boldsymbol{p}^{UWB}$ occurred during the k^{th} step. If a UWB update is available, we use

$$p^{UWB}_{x,k} = p^{UWB}_x + L_{k-1}\cos(\Psi_k + \Delta\Psi_{k-1})\frac{t_{k,end} - t_{UWB}}{t_{k,end} - t_{k,begin}} \tag{18}$$

$$p^{UWB}_{y,k} = p^{UWB}_y + L_{k-1}\sin(\Psi_k + \Delta\Psi_{k-1})\frac{t_{k,end} - t_{UWB}}{t_{k,end} - t_{k,begin}} \tag{19}$$

to estimate the UWB position at the end of the k^{th} step. In these equations, $\boldsymbol{p}^{UWB}_k = [p^{UWB}_{x,k}, p^{UWB}_{y,k}]$ is the estimated UWB position at the end of the k^{th} step, $\boldsymbol{p}^{UWB} = [p^{UWB}_x, p^{UWB}_y]$ is a vector containing the x and y coordinate of the UWB position update at timestamp t_{UWB}, and $t_{k,begin}$ and $t_{k,end}$ are respectively the start and end timestamp of the k^{th} step. This $\boldsymbol{p}^{UWB}_k$ can then be used in the update phase to correct the predicted state $\hat{\boldsymbol{x}}_k$ to the final estimate $\boldsymbol{x}_k$ for the state at the end of the k^{th} step. For the EKF, this is achieved with $\boldsymbol{x}_k = \hat{\boldsymbol{x}}_k + \boldsymbol{K}_k \cdot (\boldsymbol{p}^{UWB}_k - \hat{\boldsymbol{p}}_k)$, where $\boldsymbol{K}_k$ is the Kalman gain, which can be derived from the models for the prediction and update phase, and determines how much the measurement will influence the final state estimate $\boldsymbol{x}_k$. For the PF on the other hand, we update the weight of each particle based on $\boldsymbol{p}^{UWB}_k$. The weight of a particle indicates how well it matches with the measurements, where a high weight implies that the particle matches well with the measurements. Hence, depending on how well a particle agrees with the measured $\boldsymbol{p}^{UWB}_k$, we increase or decrease the weight of that particle.

Till now, the fusion algorithm was similar for both the sensor fusion algorithms, i.e. the EKF and PF. The PF, however, is ideally suited to include information of a floor plan. Hence, in the update phase of the PF, we can also use the floor plan to detect if impossible state transitions occurred and improve the estimate of the state. Impossible state transitions would occur when e.g. some particles of the PF would go through a wall. Similar to [23], we use the floor plan to detect if particles cross a wall during the prediction phase and exclude these particles for the final position estimate. In the results section, we will compare the performance of the EKF and PF, and investigate the performance improvement when a floor plan of the environment is used with the PF.

VII. RESULTS

In this section, we evaluate the performance of our PDR/UWB based indoor positioning system. First, we show the results of the step detector of the PDR algorithm. Next, we discuss the different test environments in which we tested our positioning system and finally, we also evaluate the sensor fusion algorithms in these environments. For all performed trials, we attached a Pozyx device [24] to the handlebar of a shopping cart. This device incorporates amongst others an accelerometer, gyroscope, magnetometer, and a UWB tag. During each test, we captured the acceleration, angular rate and UWB updates at 20Hz. To investigate the influence of a lower UWB update rate on the performance of the sensor

TABLE I
OPTIMAL PARAMETERS STEP DETECTION ALGORITHM FOR x-, y- AND z-AXIS AND THE MAGNITUDE COMPONENT

	N (samples)	f (Hz)	M (m/s²)	M-m (m/s²)	tmin (s)	tmax (s)
x	9	2	0.11	0.25	0.3	1.3
y	9	2	0.15	0.15	0.3	1.2
z	15	2	0.09	0.3	0.3	1.2
Magnitude	13	2	0.13	0.25	0.3	1.3

fusion algorithm, we downsample the UWB measurements accordingly.

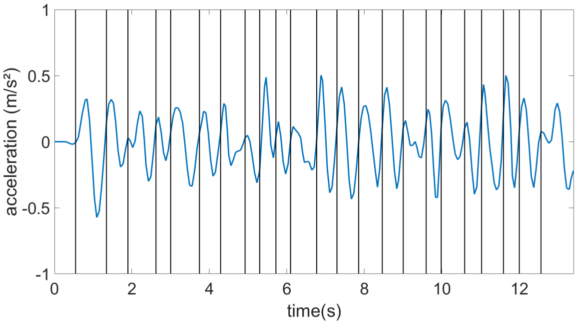
A. Results Step Detection

In this part, we discuss the results of the step detection algorithm discussed in Section III. To this end, we gathered data from the accelerometer corresponding to 32 trials with in total 1462 steps. During these trials, the user was free to choose his/her walking pace. Before and after each trial, the user holds the handlebar of the shopping cart while standing still, which allows excluding the acceleration from gripping and releasing the handlebar of the shopping cart.

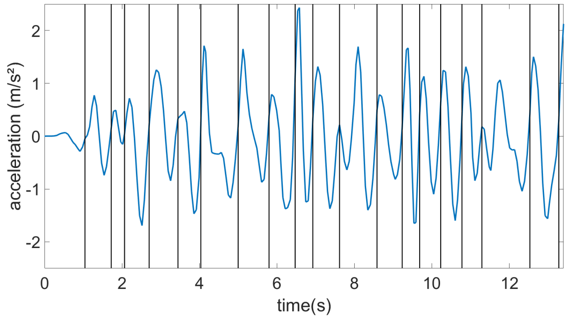
In Section III, we introduced several parameters in our step detection algorithm that required tuning. These parameters were the length N of the moving average filter, the cut-off frequency f of the low-pass filter, the threshold M_{th} for the maximum acceleration during a step, the threshold $(M-m)_{th}$ for the difference between the maximum and minimum acceleration during a step, and the minimum and maximum duration of a valid step t_{min} and t_{max}. As the average step frequency is typically around 2Hz [25], which approximately corresponds to a walking velocity of 5km/h, we expect that a cut-off frequency around $f \approx 2$Hz will eliminate acceleration components that do not correspond with steps and hence, will result in a good performance. The goal of the moving average filter was to determine the acceleration component not related to taking steps by averaging over the length of approximately one step. Hence, a logical value for the moving average length N would be equal to the length of a step. Using that the typical step frequency is around 2Hz and that we sampled the accelerometer at 20Hz, a logical value for N would be around 10.

In the remainder of this section, we determine the optimal parameters for the step detection algorithm using a grid search for the x-, y- and z-acceleration and the magnitude component. For the grid search, we use the average step error rate over all trials to optimise our algorithm, where the step error rate for one trial is equal to the absolute value of the difference between the true and estimated number of steps for that trial divided by the true number of steps. Hence, the lower the average step error rate the better the performance of the step detector.

In Table I, we show the optimal parameters for our step detection algorithm when either the x-, y- or z-component or the magnitude of the acceleration is used. To find these values, we used 50% of the steps from our trials, while the other 50% is used to validate the step detection algorithm on new data. From this table, we can see that for all four acceleration components most of the parameters have similar values. For all four acceleration components, we see that the optimal cut-off frequency is $f = 2$Hz as expected. For the x- and y-component, we see that the optimal length for the moving average filter equals $N = 9$, which is close to the expected value 10, while for the z-component and magnitude the length is higher.



(a)



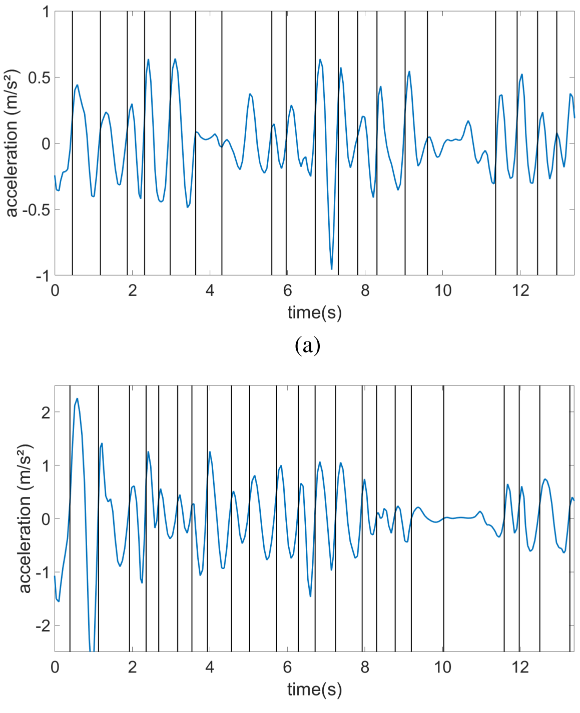
(b)

Fig. 12. Example of the processed acceleration a) in the x-axis b) of the magnitude component together with the detected steps.

In Figures 12 and 13, two fragments of the processed acceleration in the x-direction and of the magnitude component are shown together with the detected steps. While in Figure 12, we clearly see the periodic nature of the acceleration, based on which we selected our sinusoidal model for the steps, we observe in Figure 13 some anomalies around 4s for the x-axis and around 10s for both the x-axis and the magnitude. At these instants, the amplitude of the acceleration signal suddenly becomes very small although the user is continuously walking. This implies that at these instants, our step detection might not detect the zero-crossing of the signal, resulting in undetected steps.

In Figures 14 and 15, we again show two fragments of the processed x-component acceleration and of the magnitude component, but this time for a user respectively walking in a straight line and taking a turn. These figures show that, irrespective of the walking direction, our step detection algorithm will be able to extract the steps from the acceleration. Comparing the fragments for respectively the x-axis and the magnitude, we can see that with a few exceptions both result in approximately the same detected steps.

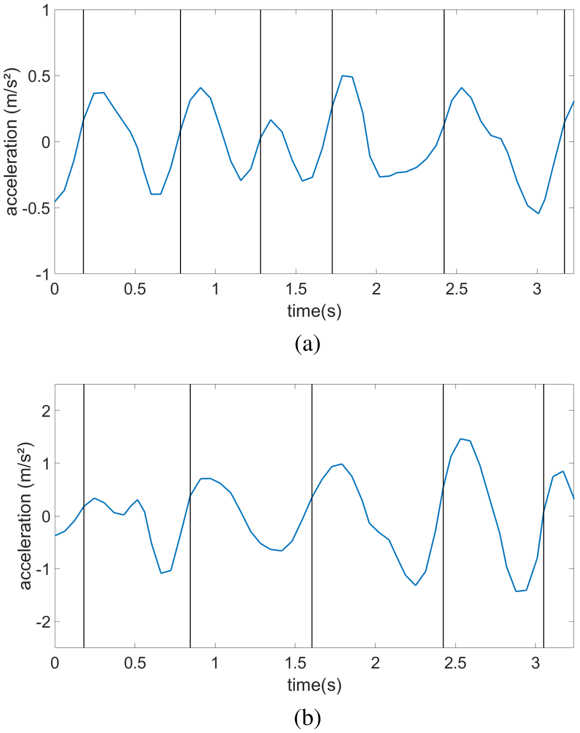
In Table II, we give the average step error rate of our step detector using the optimal parameters from Table I both for the training set, i.e. the trials that were used to find



(a)

(b)

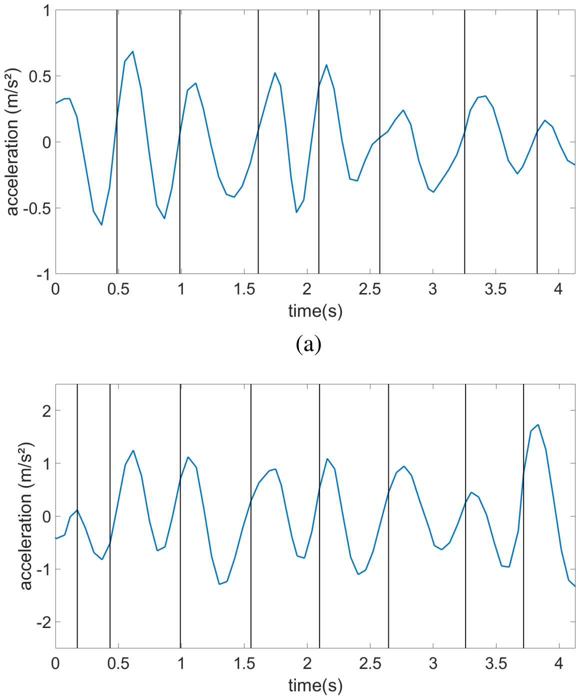
Fig. 13. Another example of the processed acceleration a) in the *x*-axis b) of the magnitude component together with the detected steps, where some steps are not detected.



(a)

(b)

Fig. 14. Example of the processed acceleration a) in the *x*-axis b) of the magnitude component together with the detected steps while walking straight.



(a)

(b)

Fig. 15. Example of the processed acceleration a) in the *x*-axis b) of the magnitude component together with the detected steps while taking a turn.

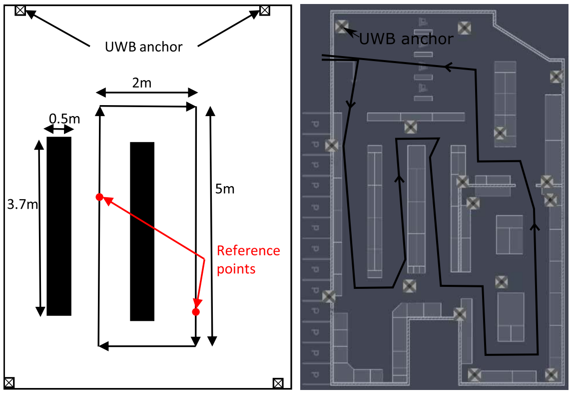
TABLE II
STEP ERROR RATE ON TRAINING AND TEST SET

	Step error rate(%)	
	training set	test set
x	8.80	6.89
y	3.31	7.80
z	4.32	7.18
Magnitude	3.83	4.83

the optimal values, and on the test set, i.e. the remaining trials. From this table, we can see that the x-axis has the worst performance on the training set and the y-axis has the best performance. On the test set, on the other hand, the magnitude performs the best, while the y-axis now results in the worst performance, although now the step error rates for the x-, y- and z-axis are much closer to each other. If we take the average of the step error rate over the training and test set, the magnitude component results in a step error rate of approximately 4.3%, the y- and z-component result in a step error rate of approximately 5.6%, while the x-component has a step error rate of around 7.9%, which is worse than most body-fixed step detectors and will hence result in a PDR system with a lower performance. However, as we combine the PDR system with UWB measurements, the UWB measurements will correct some of the errors of the PDR system. Now that we tuned the step detection algorithm, we can use it in the PDR system. In the next section, we test the PDR system and sensor fusion algorithm and determine if the PDR system can improve the accuracy of the UWB system.

B. Results PDR and Sensor Fusion

Test Environments: So far, we only evaluated the performance of the step detection algorithm from Section III. In this section, we use this step detector in the PDR system



(a) (b)

Fig. 16. a) Floor map of small-scale test environment ($6m \times 8m$). The position of the reference points and the traversed path are also shown. b) Floor map of the shopping environment ($17m \times 28m$). The position of the UWB anchors and the path through the shopping environment are also marked.

described in Section II-A and also combine it with the UWB measurements using an extended Kalman or particle filter. To evaluate the performance of our PDR system and sensor fusion algorithms, two different test environments are considered. The first small-scale test setup was installed in a room of approximately $6m$ by $8m$ at our department, where multiple metal cabinets were placed to mimic the metal racks in a shopping environment. In this room, four UWB anchors are installed and additionally, two reference points, which are used to evaluate the accuracy of the positioning algorithms, are marked on the floor. In Figure 16a, we show the floor map of this environment together with the anchors, reference points and the approximate traversed path. The second, more realistic, test environment was a shopping environment ($17m \times 28m$), for which the floor map is shown in Figure 16b together with the position of the 16 UWB anchors and the traversed path. As in this environment no accurate ground truth is available, the trials in this environment will only be used to visually inspect how the algorithms perform in more realistic environments.

Small-Scale Environment: For the tests in the small-scale environment, we repeatedly traverse the path shown in Figure 16a as close as possible. In Figure 17, we show an example of the estimated path we obtain with the EKF when no UWB updates are used, i.e. when only the PDR system is used. For the step detection algorithm, we used the magnitude component of the acceleration. From this figure, it is clear that the PDR system on its own cannot provide an accurate position estimate, i.e. although we still see in the figure the paths corresponding to the different tours around the metal cabinet, the position of the path moves to the lower left corner of the figure and the orientation of the path shows an increasingly larger error. The downward drift of the path results from errors made by the step detector, either detecting a step too many or too few, and the assumption of a fixed step length in our simple PDR algorithm. However, when we take a turn, the step length is typically smaller and consequently, the accuracy will decrease even further if we use a fixed step length. The orientation error, on the other hand, can be attributed to errors in the gyroscope data, causing an increasing skew relative to the correct path.

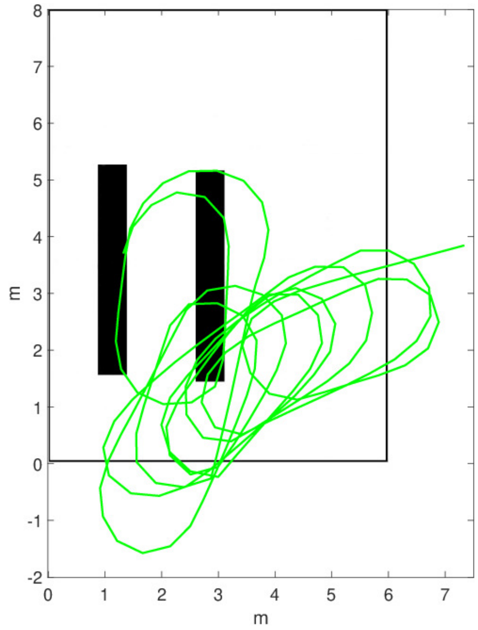


Fig. 17. Estimated path with the EKF without UWB updates and step detection on the magnitude of the measured acceleration.

Therefore, to improve the accuracy of our positioning algorithm, we want to combine the PDR system with the measurements of the UWB system, which on its own also has some limitations. Due to the metal cabinets in this environment, the line-of-sight between some of the UWB anchors and the UWB tag on the handlebar of the shopping cart is frequently obstructed, which results in less accurate UWB measurements, and hence a worse positioning accuracy. Especially when the user walks between the two rows of cabinets (see Figure 16a), we expect that the accuracy of the UWB system is severely degraded.

In this section, we compare four positioning algorithms, i.e. with UWB measurements only, the extended Kalman filter, the particle filter and the particle filter combined with floor map information. For these results, we set the update rate of the UWB measurements equal to 1Hz and assume that the initial orientation of the shopping cart is known.[1] For the EKF and PF, we also compare the accuracy when the x-, y- or z-component or the magnitude of the acceleration is used for step detection. To evaluate the accuracy of the different positioning algorithms, two reference points were marked on the floor. To ensure that the ground truth at these points was accurate, we indicated on the floor where the wheels of the shopping cart should stop, which results in a ground truth accuracy of approximately 5cm. During our experiments, we briefly stop at these points and press a button attached to

---

[1] In practice, this initial heading will often not be known. However, the fusion algorithm allows to correct for errors in the initial heading through the error heading state. After a few UWB updates, the heading will converge to the correct heading, implying the effect of an incorrect initial heading on the performance will be visible in the initial phase only. To remove the influence of the initial worse positioning on the overall performance, we assume in the results that the initial orientation is known.

TABLE III
Average Euclidean Distance Error (in cm) for UWB Only, the EKF Fusion Algorithm, PF Fusion Algorithm and the PF With Floor Map Information Fusion Algorithm When the x, y, z or Magnitude Acceleration Is Used for Step Detection

Average Euclidean error distance(cm)	UWB	EKF	PF	PF+ floor map
x	62.6	38.1	41.1	31.5
y	62.6	39.3	41.5	35.4
z	62.6	39.1	44.6	31.6
Magnitude	62.6	34.1	41.3	28.0
Average	62.6	37.7	42.1	31.6

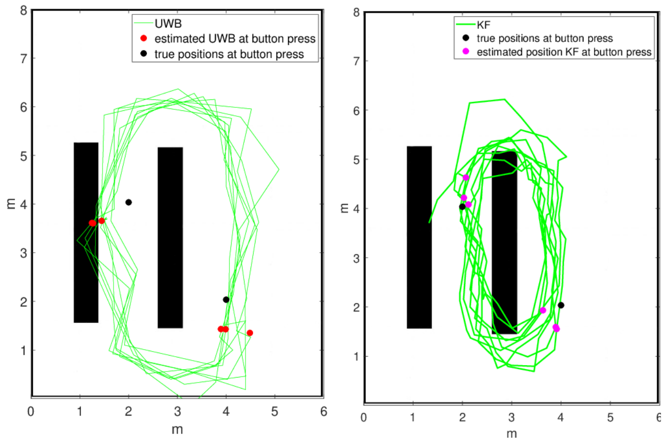
the UWB tag so that the exact time at which we need to compare the different algorithms is known.

To determine which of the algorithms performs the best, we performed a total of 15 tests in the small-scale environment, where we in total gathered 95 ground truth measurements, i.e. the user stopped at a ground truth position and pressed the button. For each of these ground truth measurements we calculated the Euclidean error distance between the true and the estimated position at each button press and determined the mean of these Euclidean distances.

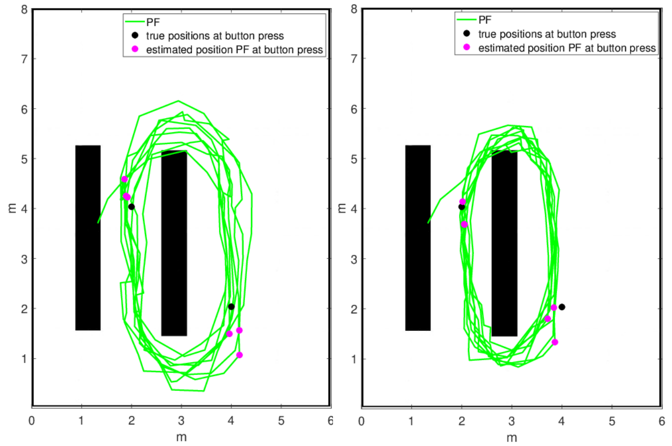
In Table III, we give the average Euclidean error distance for each of the four algorithms when the x-, y- or z-component or the magnitude of the acceleration is used for step detection. When comparing the different algorithms, a first thing we notice is that all three sensor fusion algorithms outperform the UWB-only algorithm, which has an average Euclidean error distance of 62.6cm. Comparing the EKF and PF algorithm, we see that the EKF with an average Euclidean distance error around 37.7cm performs slightly better than the PF with an average Euclidean error distance of 42.1cm. However, combining floor map information with the PF results in the best performance, i.e. an average Euclidean error distance of 31.6cm.

Comparing the algorithms when a different component of the acceleration is used for step detection, we notice that the magnitude of the acceleration results in the best performance. We also notice that the x-component performs slightly better than the y- and z-component, although the step detector using the x-component of the acceleration did not always result in the best step detection performance (see Table II). A reason for this can be that the sensor fusion algorithm is able to correct the position estimate when the step detector makes an error. Hence, from these measurements we can learn that although the step detector for a shopping-cart-mounted IMU might perform worse than for a body-fixed IMU, using this step detector can still significantly improve the accuracy when used in a sensor fusion algorithm.

In Figures 18(a) to 18(b), we show the estimated path for the four algorithms, where the magnitude of the acceleration is used in the step detection algorithm, for one of the trials together with the ground truth of the reference points and the estimated positions when the button was pressed. From these figures, we can see that in the UWB only case (Figure 18(a)), in some regions the accuracy of the UWB measurements drops, especially as expected when the user is located between the two rows of cabinets. If we compare Figures 18(b), 18(c) and 18(d), we can see that the estimated path matches more with the true path than in the UWB only case. One thing we notice is that in Figures 18(b), 18(c) and 18(d), the estimated path starts inside a black area (a cabinet), which is caused by an inaccurate initial UWB position estimate due to NLOS propagation of the UWB signals. Nonetheless, we notice that the fusion algorithms are able to correct themselves after this initial error.



(a) UWB (b) EKF



(c) PF (d) PF with floor map

Fig. 18. Estimated paths for a) UWB only b) the EKF fusion algorithm c) PF fusion algorithm d) PF with floor map fusion algorithm.

Shopping Environment: As a final test for our sensor fusion algorithms, we perform some additional trials in a more realistic shopping environment. First, we compare both the EKF and PF implementation and look at the influence of the UWB update rate. Secondly, we add floor map information to the particle filter and evaluate the effect on the performance. To test our algorithm we roughly followed the path shown in Figure 16b through the shopping environment. In Figure 19, we show the estimated trajectory when only UWB data at 20Hz and no information from the PDR system is used. In general, we see that the UWB system gives a very reliable estimate for the position in the store. However, in some regions

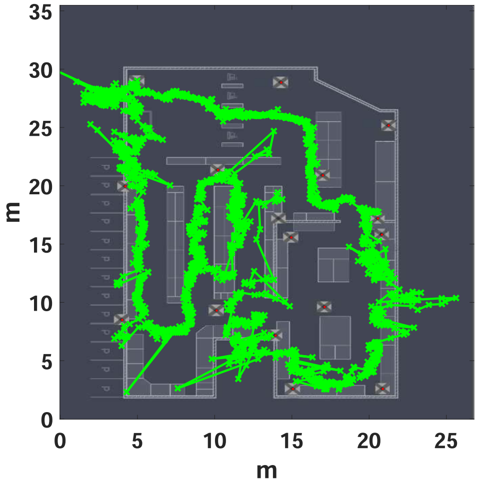
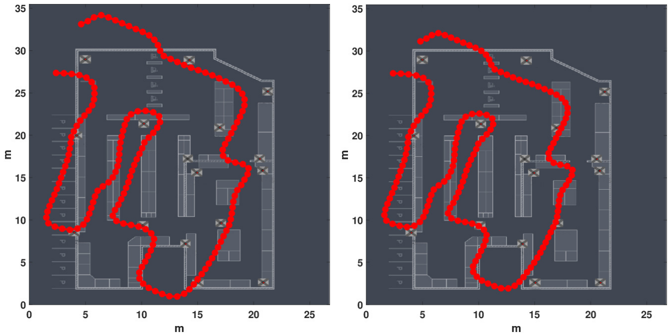


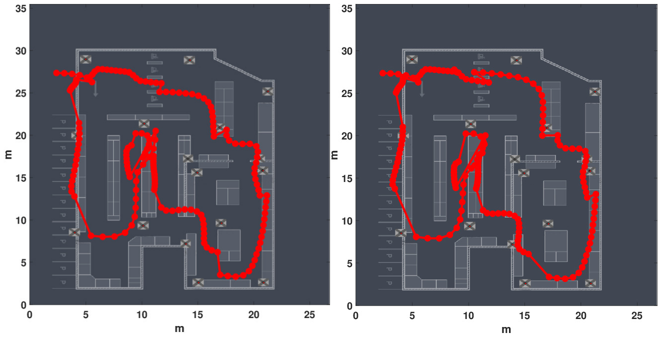
Fig. 19. Estimated trajectory using only UWB measurements at 20Hz.

we can clearly see that the accuracy is degraded, which can be caused by NLOS propagation. For example, on the right side of Figure 19 the estimated position at one point lies approximately 5m outside the store. This shows that another localisation technique is necessary to achieve good accuracy in the whole store. In Figure 20, we show the resulting trajectories of the EKF and PF for different UWB update rates. For these results, we used the magnitude of the acceleration for step detection and again assume that the initial orientation is known. In Figures 20a and 20b, we see that the estimated trajectories without the use of UWB measurements have a shape similar to the true path. However, we notice that due to incorrect step length and heading estimations the estimated path diverges from the true path. When we periodically use UWB updates to correct the position estimation, we see that the drift is regularly compensated for, and that the estimated trajectory follows the true trajectory better. A disadvantage we notice with the estimated paths using sensor fusion, however, is that the path is less smooth than without UWB due to jumps in the estimated position when a UWB measurement was available. Comparing respectively Figures 20c and 2e and Figures 20d and 20f, we notice that in general a higher UWB update rate results in a better estimate for the traversed trajectory. However, on some parts of the trajectory (e.g. in the bottom centre of the floor map), the estimated UWB position was inaccurate due to NLOS propagation caused by the metal racks in the shop, resulting in an erroneous position update in our sensor fusion algorithm.[2] Hence, on these parts higher UWB updates decrease the accuracy of our algorithm. Comparing Figures 20c, 20e, 20d and 20f, it follows that the fusion algorithm with an update rate of 0.1Hz suffers less from the erroneous UWB position estimates than with an update rate of 1Hz. Finally, if we compare the trajectories using an EKF and a PF, we notice that the performance is very similar for both.



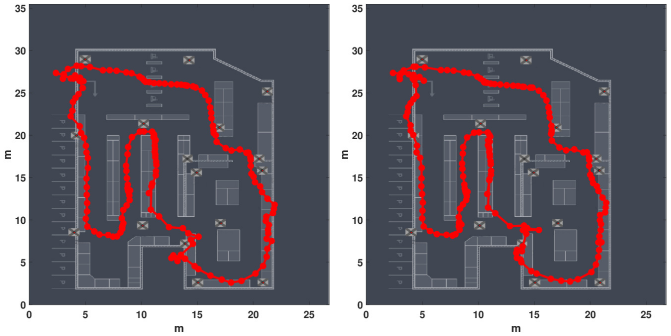
(a) EKF without UWB updates

(b) PF without UWB updates



(c) EKF with 0.1Hz UWB update rate

(d) PF with 0.1Hz UWB update rate



(e) EKF with 1Hz UWB update rate

(f) PF with 1Hz UWB update rate

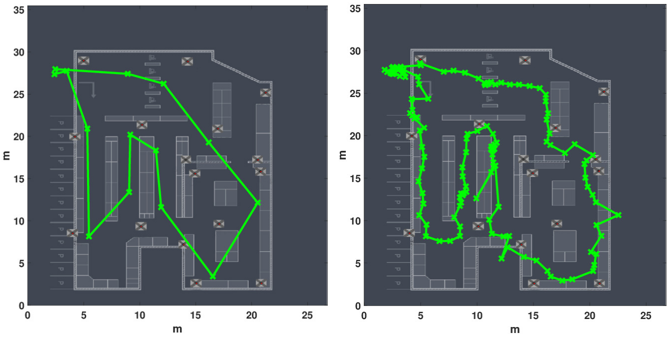
Fig. 20. Estimated trajectories for the EKF and PF with different UWB update rates.

In Figure 21, we show the estimated trajectories when only UWB measurements are used with an update rate of 0.1Hz and 1Hz. For an update rate of 0.1Hz, we can see that using only UWB measurements results in a trajectory with a limited number of points. Comparing Figure 21a with Figures 20c and 20d, it is clear that the sensor fusion algorithm provides a trajectory with a much higher update rate, although jumps in the trajectories are present. For an update rate of 1Hz, on the other hand, comparing Figure 20b with Figures 20e and 20f, we observe that the UWB-only case and the sensor fusion algorithm both result in a trajectory with a sufficient update rate. However, in the UWB-only case, outliers are present, which are partially rejected by the sensor fusion algorithm.

Until now, we have only used the data from the PDR and UWB system in our sensor fusion algorithm. However, in the

---

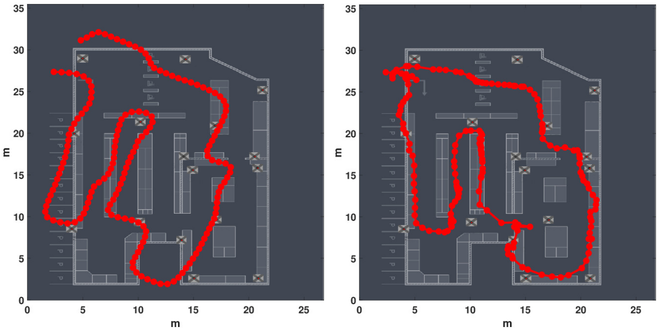
[2] Note that the performance of the UWB algorithm can be improved by detecting the outliers caused by NLOS measurements, and ignoring these outliers in the reconstruction of the path. However, outlier detection is only possible if the UWB update rate is sufficiently high. With a UWB update rate of 0.1 Hz, as used in Figures 20c and 20d, detecting an outlier will not be straightforward.



(a) UWB-only with 0.1Hz update rate

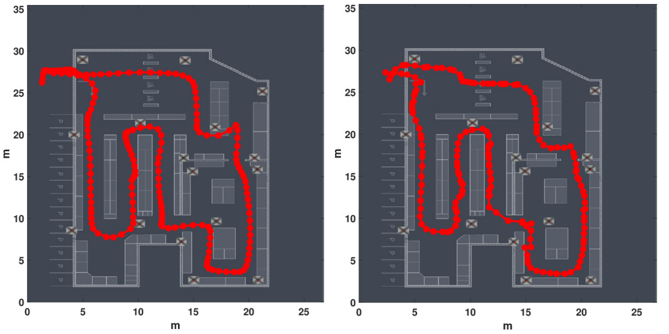
(b) UWB-only with 1Hz update rate

Fig. 21. Estimated trajectories for UWB-only with different UWB update rates.



(a) PF without map information and no UWB updates

(b) PF without map information and 1Hz UWB update rate



(c) PF with map information and no UWB updates

(d) PF with map information and 1Hz UWB update rate

Fig. 22. Comparison of estimated trajectories for PF with/without map information for multiple UWB update rates.

PF it is possible to easily include floor map information, preventing particles of the PF to go through e.g. walls or shopping racks. In Figure 22, we compare the estimated trajectories for the PF with and without using the floor map when no UWB updates are used and when the UWB update rate is 1Hz. Comparing Figures 22a and 22c, we see that without UWB updates, using the map information largely eliminates the drift we noticed before. Even without UWB updates, the PF results in a trajectory that is very similar to the true trajectory in shape. Previously, in some positions, NLOS propagation of the UWB signals deteriorated the position estimates such that the position estimates were outside of the store. If we compare Figures 22b and 22d, it is clear that using the floor map and a UWB update rate of 1Hz is much more robust to bad UWB measurements. The PF combined with map information and a UWB update of 1Hz results in a trajectory that is very close to the true trajectory.

VIII. CONCLUSION

The contributions of this article are twofold. First, we propose a model for the acceleration of a shopping-cart-mounted accelerometer, which we then used to derive a step detection algorithm. In our experiments, we compared the step detector in the case it was used on the different components of the measured acceleration. From these experiments, it followed that as well the x-, y- and z-axis of the acceleration resulted in a step detection error rate of approximately 7% on our validation set, while the magnitude of the acceleration resulted in a step detection error rate of 4.83%. Secondly, we then use this step detector in a PDR system and fuse the information of the PDR system with data from the UWB measurements. To this end, we both implement an EKF and PF and compare the performance of each sensor fusion algorithm. First, we test our algorithms in a small-scale environment. Using only UWB measurements and no sensor fusion results in a mean Euclidean error distance of 62.6cm, while the EKF and PF results respectively in a mean Euclidean error distance of 34.1cm and 41.3cm. When floor map information is used in the PF the mean Euclidean error distance even further improves to 28.0cm. Hence, fusing the data from the PDR and UWB system can result in a significant improvement. Next, we evaluate our algorithms in a more realistic shopping environment. From our experiments, we found that the performance of the EKF and PF is very similar but the EKF has the advantage that it requires less computational resources. Finally, we also use map information in the PF, which significantly improves the performance. Even without using UWB measurements, the combination of the PDR system and floor map information results in a good estimation of the traversed trajectory.

REFERENCES

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[2] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 13–32, 1st Quart., 2009.

[3] M. Kok, J. D. Hol, and T. B. Schon, "Indoor positioning using ultrawideband and inertial measurements," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1293–1303, Apr. 2015.

[4] G. Du, P. Zhang, and D. Li, "Human–manipulator interface based on multisensory process via Kalman filters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5411–5418, Oct. 2014.

[5] S. Beauregard, "A helmet-mounted pedestrian dead reckoning system," in *Proc. 3rd Int. Forum Appl. Wearable Comput.*, Mar. 2006, pp. 1–11.

[6] J.-O. Nilsson, A. K. Gupta, and P. Handel, "Foot-mounted inertial navigation made easy," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2014, pp. 24–29.

[7] A. Perttula, H. Leppakoski, M. Kirkko-Jaakkola, P. Davidson, J. Collin, and J. Takala, "Distributed indoor positioning system with inertial measurements and map matching," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 11, pp. 2682–2695, Nov. 2014.

[8] H. Zhang, W. Yuan, Q. Shen, T. Li, and H. Chang, "A handheld inertial pedestrian navigation system with accurate step modes and device poses recognition," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1421–1429, Mar. 2015.

[9] Q. Tian, K. I.-K. Wang, and Z. Salcic, "Human body shadowing effect on UWB-based ranging system for pedestrian tracking," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 4028–4037, Oct. 2019.

[10] A. Al-Hamad, A. Ali, M. Elhoushi, and J. Georgy, "Indoor navigation using consumer portable devices in Cart/Stroller," in *Proc. 30th Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS+)*, Nov. 2017, pp. 25–29.

[11] T. Do-Xuan, V. Tran-Quang, T. Bui-Xuan, and V. Vu-Thanh, "Smartphone-based pedestrian dead reckoning and orientation as an indoor positioning system," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2014, pp. 303–308.

[12] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM Conf. Ubiquitous Comput. UbiComp*, 2012, pp. 421–430.

[13] M. Alzantot and M. Youssef, "UPTIME: Ubiquitous pedestrian tracking using mobile phones," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 3204–3209.

[14] H.-H. Lee, S. Choi, and M.-J. Lee, "Step detection robust against the dynamics of smartphones," *Sensors*, vol. 15, no. 10, pp. 27230–27250, Oct. 2015.

[15] E. M. Diaz and A. L. M. Gonzalez, "Step detector and step length estimator for an inertial pocket navigation system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2014, pp. 105–110.

[16] H. Weinberg, "Using the ADXL202 in pedometer and personal navigation applications," Analog Devices, Norwood, MA, USA, Appl. Note 602, 2002.

[17] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *J. Global Positioning Syst.*, vol. 3, nos. 1–2, pp. 273–279, Dec. 2004.

[18] Q. Tian, Z. Salcic, K. I.-K. Wang, and Y. Pan, "A multi-mode dead reckoning system for pedestrian tracking using smartphones," *IEEE Sensors J.*, vol. 16, no. 7, pp. 2079–2093, Apr. 2016.

[19] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, Jun. 2012.

[20] A. Ferrari, P. Ginis, M. Hardegger, F. Casamassima, L. Rocchi, and L. Chiari, "A mobile Kalman-filter based solution for the real-time estimation of Spatio-temporal_newline gait parameters," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 7, pp. 764–773, Jul. 2016.

[21] A. Rampp, J. Barth, S. Schuelein, K.-G. Gassmann, J. Klucken, and B. M. Eskofier, "Inertial sensor-based stride parameter calculation from gait sequences in geriatric patients," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 4, pp. 1089–1097, Apr. 2015.

[22] J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schon, "Tightly coupled UWB/IMU pose estimation," in *Proc. IEEE Int. Conf. Ultra-Wideband*, Sep. 2009, pp. 688–692.

[23] S. Beauregard, Widyawan, and M. Klepal, "Indoor PDR performance enhancement using minimal map information and particle filters," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, May 2008, pp. 141–147.

[24] P. Labs. *Pozyx—Accurate Positioning*. Accessed: Nov. 19, 2020. [Online]. Available: https://www.pozyx.io/

[25] T. Ji and A. Pachi, "Frequency and velocity of people walking," *Struct. Eng.*, vol. 84, no. 3, pp. 36–40, 2005.

Stef Vandermeeren was born in Jette, Belgium, in 1992. He received the B.E. degree, and the M.Sc. degree in electrical engineering from Ghent University, Ghent, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Department of Telecommunications and Digital Information Processing (TELIN). In 2015, he joined TELIN. His research interests include sensor fusion, indoor localization, and machine learning.

Heidi Steendam (Senior Member, IEEE) received the M.Sc. degree in electrical engineering and the Ph.D. degree in applied sciences from Ghent University, Ghent, Belgium, in 1995 and 2000, respectively. Since September 1995, she has been with the Digital Communications (DIGCOM) Research Group, Department of Telecommunications and Information Processing (TELIN), Faculty of Engineering, Ghent University, first in the framework of various research projects. Since October 2002, she has been a Professor of digital communications. In 2015, she was a Visiting Professor with Monash University. She has authored more than 150 scientific articles in international journals and conference proceedings, for which several best paper awards were received. Her current research interests include statistical communication theory, carrier and symbol synchronization, bandwidth-efficient modulation and coding, cognitive radio and cooperative networks, radar sensing, positioning, and visible light communication.

Since 2002, she has been an executive Committee Member of the IEEE Communications and Vehicular Technology Society Joint Chapter, Benelux Section, since 2012 the vice chair and since 2017 the chair. She was active in various international conferences as Technical Program Committee chair/member and Session chair. In 2004, 2011 and 2018, she was the conference chair of the IEEE Symposium on Communications and Vehicular Technology in the Benelux. From 2012 till 2017, she was associate editor of IEEE Transactions on Communications and EURASIP Journal on Wireless Communications and Networking.