Solin, Arno; Cortes Reina, Santiago; Rahtu, Esa; Kannala, Juho

Inertial Odometry on Handheld Smartphones

Please cite the original version:
Solin, A., Cortes Reina, S., Rahtu, E., & Kannala, J. (2018). Inertial Odometry on Handheld Smartphones. In
2018 21st International Conference on Information Fusion, FUSION 2018 (pp. 1361-1368). [8455482] IEEE.
https://doi.org/10.23919/ICIF.2018.8455482

Inertial Odometry on Handheld Smartphones

Arno Solin
Aalto University
Espoo, Finland
arno.solin@aalto.fi

Santiago Cortes
Aalto University
Espoo, Finland
santiago.cortesreina@aalto.fi

Esa Rahtu
Tampere Univ. of Tech.
Tampere, Finland
esa.rahtu@tut.fi

Juho Kannala
Aalto University
Espoo, Finland
juho.kannala@aalto.fi

Abstract—Building a complete inertial navigation system using the limited quality data provided by current smartphones has been regarded challenging, if not impossible. This paper shows that by careful crafting and accounting for the weak information in the sensor samples, smartphones are capable of pure inertial navigation. We present a probabilistic approach for orientation and use-case free inertial odometry, which is based on double-integrating rotated accelerations. The strength of the model is in learning additive and multiplicative IMU biases online. We are able to track the phone position, velocity, and pose in real-time and in a computationally lightweight fashion by solving the inference with an extended Kalman filter. The information fusion is completed with zero-velocity updates (if the phone remains stationary), altitude correction from barometric pressure readings (if available), and pseudo-updates constraining the momentary speed. We demonstrate our approach using an iPad and iPhone in several indoor dead-reckoning applications and in a measurement tool setup.

I. Introduction

The deployment of global navigation satellite systems (GNSSs) has solved many large-scale positioning problems. However, these systems are not suited for precise tracking or for indoor use, which is where people spend most of their time. Accurate and fast indoor localization and tracking has many potential uses, including safety and emergency assistance, security, resource efficiency, navigation and augmented reality.

The idea of an inertial navigation system (INS, see [1, 2]) is to use the fusion of inertial sensors (accelerometers and gyroscopes) to continuously estimate the position, orientation, and velocity of a moving object. This type of tracking, known as dead-reckoning, is typically associated with air-craft, submarines, and missile technology. Recent advances in MEMS sensors have brought motion and rotation sensors to standard consumer smartphones and devices, and introduced the potential for new INS applications.

Smartphones and tablet devices are equipped with MEMS sensors in order to enhance human-computer interaction and enable new applications. For example, thanks to the accelerometer, devices can automatically rotate the screen based on the device orientation with respect to gravity. Furthermore, gyroscopes have enabled new ways to interact with digital content, such as watching of panoramic video or controlling games by rotating the device. In fact, besides gravitation sensing and tracking [3], information fusion from accelerometers, gyroscopes and magnetometers can be utilised for robust real-time tracking of the full device orientation [4, 5]. Such

approaches are sometimes referred to as *attitude and heading reference systems* (AHRS).

Tracking the translational motion of devices based on inertial sensors is considerably harder than orientation tracking. However, certain applications, like pedestrian tracking and indoor positioning, would greatly benefit from accurate inertial navigation on smartphones. The difficulty of inertial navigation is due to the need to double-integrate the observed accelerations, which rapidly accumulates errors from the high noise-level of MEMS accelerometers. Small errors in the attitude estimation will make this even more challenging as the gravitation may 'leak' to the integrated accelerations [6].

In order to solve the aforementioned challenges, many current systems resort to additional hardware, such as foot-mounted sensors [7, 8] or video cameras. While providing accurate results, these are quite impractical for wide use in consumer applications. For example, camera-based approaches do not work when the device is in a closed bag or pocket, and capturing and processing video consumes a lot of energy compromising battery longevity. Further, while foot-mounted sensors can provide accurate tracking thanks to frequent zero-velocity updates and high-quality sensors, they are inconvenient for large-scale consumer use and the current solutions do not work well when the movement happens without steps, for example in a trolley, elevator, or escalator.

In this paper we show how an inertial navigation system can be built to work on the limited-quality data provided by a standard smartphone. We propose a general inertial navigation approach which is not based on detecting steps and therefore works in various use cases, covering both legged motion and motion with wheels, as well as motion in elevators and escalators. Moreover, the approach does not require constraining the device orientation, and thus the device can be held freely. In addition, the approach is computationally light-weight and capable for real-time processing on a smartphone. To the best of our knowledge, this is the first paper demonstrating such a system with a standard smartphone.

Figure 1 summarizes the features of the proposed INS system in a test performed with a standard iPhone 6. In this example, the path was started on the ground floor with zero-velocity updates for calibrating the sensors (no pre-calibrations done). After walking up the stairs to the first floor holding the phone in the hand, a position fix was given, after which the phone was put in a bag. Next, the phone was taken out of the bag and put in the trouser pocket. Before descending to
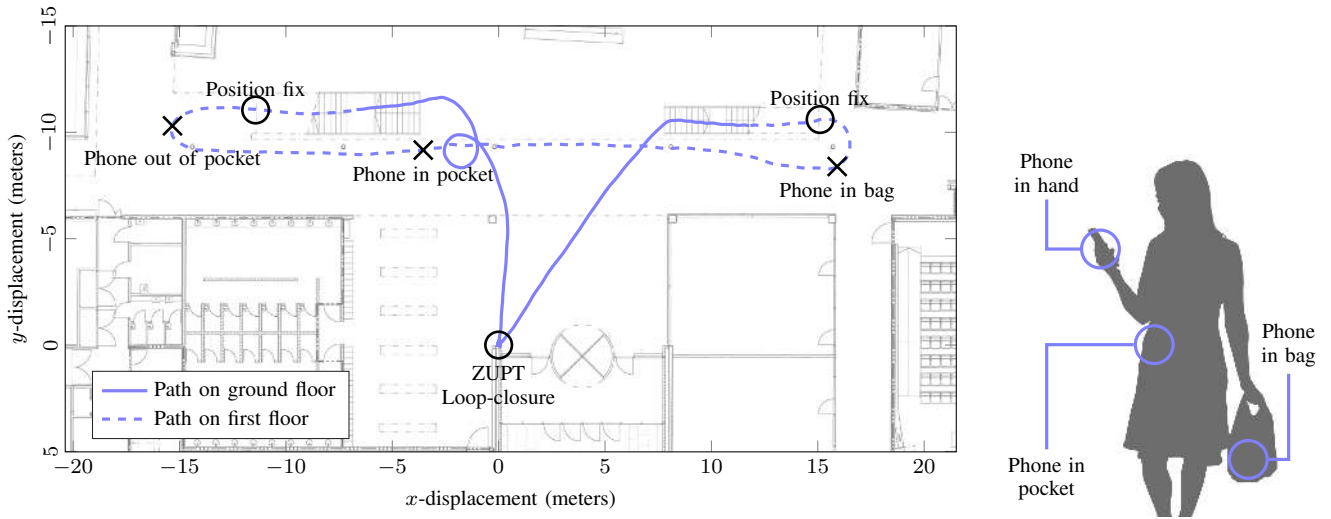
Fig. 1. Features of the INS system summarized into one figure: The path was started on the ground floor with zero-velocity updates. After walking up the stairs to the first floor a position fix was given, after which the phone was put in a **closed bag**. Then the phone was put in the **pocket**. Before descending to the ground floor, the phone was taken out of the pocket and a second position fix was given (aligning the path to the map). On the ground floor a manual loop-closure was given. The data was collected by an iPhone 6 and calibrations were performed on the fly.

the ground floor, the phone was taken out of the pocket and a second position fix was given, which aligned the path to the map. On the ground floor a manual loop-closure indicated that we were where we started.

The contributions of this paper are two-fold:

- We show that inertial navigation on a standard smartphone is feasible by careful crafting of the model, taking advantage of weak signals, and accounting for uncertainties in data.
- We present a streamlined estimation approach for the INS problem which builds upon learning the dynamical sensor bias parameters as a part of the state variables. The probabilistic inference is solved by a sequential filtering scheme, where the only approximations come from the linearizations inside the extended Kalman filter. The approach is complemented with zero-velocity updates and pseudo-measurements limiting the momentary speed.

This paper is structured as follows. In the next section we provide a brief literature review of previous work. In Section III we present the INS model. The exact model is presented in detail, and measurement updates for fusing measurements with dynamics are described. Section IV presents empirical studies where the inertial navigation algorithm is employed in pedestrian dead-reckoning examples, a generalized dead-reckoning example, and as a measurement tool. Finally, the results are discussed in Section V.

II. RELATED WORK

Inertial navigation systems have been studied for decades. The classical literature cover primarily navigation applications for aircraft and large vehicles [1, 2, 9, 10]. The development of handheld consumer-grade devices has awakened an interest in pedestrian navigation applications, where the challenges are slightly different from those in the classical approaches.

That is, the limited quality of smartphone MEMS sensors and abrupt motions of hand-held devices pose additional challenges which have so far prevented generic inertial navigation solutions for smartphone applications.

In order to focus on the relevant previous literature, we restrict our scope to tracking algorithms that use the sensors available in a smartphone, primarily accelerometers, gyroscopes, and magnetometers.

The extensive survey by Harle [11] covers many approaches with different constraints for the use of inertial sensors for pedestrian dead-reckoning (PDR). Typically INS systems either constrain the motion model or rely on external sensors. In fact, we are not aware of any previous system which would have all the capabilities that we demonstrate in this paper.

One prominent INS solution relying on external hardware is the OpenShoe project [8, 12]. It uses foot-mounted inertial sensors with several pairs of accelerometers and gyroscopes to estimate the step-by-step PDR (in an INS-SHS framework, see below). The model is constrained by zero-velocity updates (ZUPTs) on each step once the foot touches the ground.

Step and heading systems (SHS, see *e.g.* [5, 13–16]) use the inertial sensor to estimate the heading and the step length of the user. These are introduced into a constrained model that estimates the walking path by accumulating the step vectors in order to do PDR. These systems have been proven to work well for PDR in short and medium range but they typically impose constraints for the device orientation. For example, the device orientation is often known or fixed with respect to the walking direction. Further, they are very sensitive to changing gaits and are prone to false positives (see discussion in [11]). A recent approach [17] uses bipedal locomotion models to model the periodical behavior of the INS in a smartphone and, thus, estimate steps. Although they are able to relax the constraint of known and fixed device orientation to some extent, their

approach is still step-based, and heading estimation is error-prone, especially if there are frequent and abrupt changes in orientation.

Besides inertial PDR systems, there exist many camera-aided inertial tracking solutions (visual-inertial odometry), which can provide accurate tracking in visually distinguishable environments (*e.g.* [18–21]). However, as these approaches require constant use of a video camera, causing increased battery usage, and unobstructed visibility of surroundings, they are not directly comparable to our approach.

Finally, it should be noted that often odometry estimation techniques, either inertial or visual, are part of larger localization systems, which combine odometry with various kinds of maps or fingerprinting methods that provide reference positions. Examples of mapped signals, which have been utilized for indoor localization, include signal strengths of Wi-Fi and Bluetooth radio beacons [22], cellular communications radio [23], RFID tags [24], and variations of the ambient magnetic field [25, 26].

III. METHODS

Even though, the physical interpretation of how an inertial navigation system works is straight-forward, this setup has many pitfalls. All inertial navigation systems suffer from integration drift. Small errors in the measurements of acceleration and angular velocity cause progressively larger errors in velocity—and even greater errors in position. The dominating component in the accelerometer data is gravity, which means that even slight errors in orientation make the gravity 'leak' into the estimates. The sequential nature of the problem makes the errors accumulate. Once the estimates start to drift, they quickly diverge.

These problems underline the importance of accurately modelling and handling the inherent noises, sampling times, uncertainties, and numerical instabilities in the system. We use the data provided by the inertial measurement unit (IMU) in the smartphone to continuously infer the relative change in position, velocity and orientation of the device with respect to a starting point (see [1, 2]). The three-axis IMU measures data of the specific force (accelerometer data) and angular rate (gyroscope data).

A. Non-Linear Estimation

An inertial navigation system is non-linear both in the dynamics and observations. Non-linear filtering methods (see [27] for an overview) are concerned with this kind of estimation problems. Consider a non-linear state-space equation model of form

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \boldsymbol{\varepsilon}_k), \tag{1}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \boldsymbol{\gamma}_k), \tag{2}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state at time step t_k, $k = 1, 2, \ldots$, $\mathbf{y}_k \in \mathbb{R}^m$ is a measurement, $\boldsymbol{\varepsilon}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_k)$ is the Gaussian process noise, and $\boldsymbol{\gamma}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_k)$ is the Gaussian measurement noise. The dynamics and measurements are specified in terms of the

dynamical model function $\mathbf{f}_k(\cdot)$ and the measurement model function $\mathbf{h}_k(\cdot)$, both of which can depend on the time step k.

We employ the extended Kalman filter (EKF, [9]) which provides a means of approximating the state distributions

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \simeq \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k) \tag{3}$$

with Gaussians through first-order linearizations. In the experiments, we also employ the fixed-interval extended Rauch–Tung–Striebel smoother (see [27] for detailed presentation) for obtaining the state distributions $p(\mathbf{x}_k \mid \mathbf{y}_{1:N})$ conditioned on the entire track of observations.

B. Dynamical Model

The state variables hold the knowledge of the system state at any given time step. The state variables are:

$$\mathbf{x}_k = (\mathbf{p}_k, \mathbf{v}_k, \mathbf{q}_k, \mathbf{b}_k^{\mathrm{a}}, \mathbf{b}_k^{\omega}, \mathbf{T}_k^{\mathrm{a}}), \tag{4}$$

where $\mathbf{p}_k \in \mathbb{R}^3$ is the position, $\mathbf{v}_k \in \mathbb{R}^3$ the velocity, and $\mathbf{q}_k$ the orientation unit quaternion at time step t_k. The remaining components are the additive accelerometer and gyroscope bias components, and $\mathbf{T}_k^{\mathrm{a}}$ denotes the diagonal multiplicative scale error of the accelerometer.

The dynamical model (Eq. 1) is based on the assumption that position is velocity once integrated, and velocity is acceleration (with the influence of gravity removed) once integrated. The orientation of the acceleration is tracked with gyroscope measurements. The accelerometer and gyroscope readings are regarded as control signals, and their measurement noises are seen as the process noise of the system.

The dynamical model given by the mechanization equations (see, *e.g.*, [8, 10] for similar model formulations) is

$$\begin{pmatrix} \mathbf{p}_k \\ \mathbf{v}_k \\ \mathbf{q}_k \end{pmatrix} = \begin{pmatrix} \mathbf{p}_{k-1} + \mathbf{v}_{k-1}\Delta t_k \\ \mathbf{v}_{k-1} + [\mathbf{q}_k(\tilde{\mathbf{a}}_k + \boldsymbol{\varepsilon}_k^{\mathrm{a}})\mathbf{q}_k^{\star} - \mathbf{g}]\Delta t_k \\ \boldsymbol{\Omega}[(\tilde{\boldsymbol{\omega}}_k + \boldsymbol{\varepsilon}_k^{\omega})\Delta t_k]\mathbf{q}_{k-1} \end{pmatrix}, \tag{5}$$

where the time step length is given by $\Delta t_k = t_k - t_{k-1}$ (note that we *do not* assume equidistant sampling times), the accelerometer input is denoted by $\tilde{\mathbf{a}}_k$ and the gyroscope input by $\tilde{\boldsymbol{\omega}}_k$. Gravity $\mathbf{g}$ is a constant vector. The quaternion rotation is denoted by the $\mathbf{q}_k[\cdot]\mathbf{q}_k^{\star}$ notation, and the quaternion rotation update is given by the function $\boldsymbol{\Omega} : \mathbb{R}^3 \to \mathbb{R}^{4\times4}$ (see [10] for details).

The system is deterministic up to the uncertainties (measurement noises and biases) associated with the accelerometer and gyroscope data. The process noises associated with the inputs are modelled as i.i.d. Gaussian noise $\boldsymbol{\varepsilon}_k^{\mathrm{a}} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}^{\mathrm{a}}\Delta t_k)$ and $\boldsymbol{\varepsilon}_k^{\omega} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}^{\omega}\Delta t_k)$. The Jacobians of (5), required for the linearizations in filtering, can be constructed in closed-form.

The accelerometer and gyroscope readings provided by the low-cost sensors in the mobile device may suffer from misalignment errors and scale errors in addition to white measurement noise. These are taken into account inside the dynamic model as follows:

$$\begin{aligned} \tilde{\mathbf{a}}_k &= \mathbf{T}_k^{\mathrm{a}}\, \mathbf{a}_k - \mathbf{b}_k^{\mathrm{a}}, \\ \tilde{\boldsymbol{\omega}}_k &= \quad \boldsymbol{\omega}_k - \mathbf{b}_k^{\omega}, \end{aligned} \tag{6}$$

where the accelerometer and gyroscope sensor readings at t_k are $\mathbf{a}_k$ and $\boldsymbol{\omega}_k$. The additive biases are denoted by $\mathbf{b}_k^{\mathrm{a}}$ and $\mathbf{b}_k^{\omega}$, respectively. The diagonal scale error matrix $\mathbf{T}_k^{\mathrm{a}}$ accounts for miscalibrations in the accelerometer scale.

The biases and diagonal scale error terms are estimated online as a part of the state estimation problem. They are considered fixed over the entire time horizon, thus the dynamic model for their part is fixed and without any process noise:

$$\mathbf{b}_k^{\mathrm{a}} = \mathbf{b}_{k-1}^{\mathrm{a}}, \quad \mathbf{b}_k^{\omega} = \mathbf{b}_{k-1}^{\omega}, \quad \text{and} \quad \mathbf{T}_k^{\mathrm{a}} = \mathbf{T}_{k-1}^{\mathrm{a}}. \tag{7}$$

This means that their values are controlled by the prior state and information provided by the measurement updates.

The complete dynamical model must be differentiated both in terms of the state variables and process noise terms in order to fit the EKF estimation scheme (see [27]). These derivatives can be derived in closed-form in order to preserve the stability of the system. The initial (prior) state is given by $\mathbf{p}_0 \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_0^{\mathrm{p}})$, $\mathbf{v}_0 \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_0^{\mathrm{v}})$, and $\mathbf{q}_0$ chosen such that it defines the initial orientation (deduced from gravity direction). The additive biases are initialized to zero and the scale bias to an identity matrix.

C. Position Fixes and Loop-Closures

In terms of the sequential inference scheme all auxiliary observation data is combined with the model through the measurement model in Equation 2. Position fixes are noisy measurements of the position vectors $\mathbf{p}_k$ in the state (*i.e.* $\mathbf{h}_{\mathrm{pos.}}(\mathbf{x}) = \mathbf{p}$). The additive Gaussian measurement noise represents the uncertainty associated with the given position.

Position fixes provide uncertain information of the position and thus also the distance travelled between the position fixes. This first-hand information helps the model pin down the bias estimates very accurately. Loop-closure points do not provide any exact position information, but indicate that at two different points in time, the positions are the same. Also this information is valuable in inferring sensor biases.

Manual loop-closures can be combined with the estimation scheme by augmenting the current position estimate in the state by a Kalman update at loop-opening. The state dimension grows by three at opening the loop, and the state becomes

$$\mathbf{x} = (\mathbf{x}^{\mathrm{old}}, \mathbf{p}^{\mathrm{LC}}), \tag{8}$$

where the prior $\mathbf{p}^{\mathrm{LC}} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_0^{\mathrm{LC}})$ with the $\boldsymbol{\Sigma}_0^{\mathrm{LC}}$ sufficiently large indicating the non-informativity of the initial location of the loop-closure point. In practice, both at t_{open} and t_{close} (the loop can be closed many times) the measurement model

$$\mathbf{h}_{\mathrm{LC}}(\mathbf{x}) = \mathbf{p} - \mathbf{p}^{\mathrm{LC}} \tag{9}$$

defines an observation $\mathbf{y} = \mathbf{0}$ with some measurement noise $\boldsymbol{\gamma} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}^{\mathrm{LC}})$. The measurement noise covariance $\boldsymbol{\Sigma}^{\mathrm{LC}}$ should reflect the mismatch of the user not exactly being at the loop-closure spot. Both the position fix and loop-closures are linear observations of the state, and can thus be implemented using a standard (linear) Kalman update.

D. Zero-Velocity Updates

In this paper, the most important source of auxiliary information is so called zero-velocity updates (ZUPTs, see [8] for discussion). Once the phone is detected to be stationary for any period of time, it is known to the model that the system velocity must be zero ($\mathbf{v} = \mathbf{0}$). In terms of the measurement model, this means

$$\mathbf{h}_{\mathrm{ZUPT}}(\mathbf{x}) = \mathbf{v} \tag{10}$$

and the additive measurement noise $\boldsymbol{\gamma}$ is small (by only specifying a pseudo-noise scale). This update can be performed as a standard (linear) Kalman update.

For triggering, we use an iterative Dickey–Fuller stationarity test [28] on a rolling window of accelerometer data (window size 250 ms) with an additional requirement of the sample standard deviation being small. This means that trends in the data are used as a proxy for movement.

E. Pseudo-Measurement Updates

Without position fixes, loop-closures, or ZUPTs the inertial navigation system quickly becomes unstable. Once the estimates start diverging, they easily loose their numerical precision. The main source of these problems is gravity 'leaking' into the acceleration input and corrupting the velocity vector. Once the velocity starts to drift, the position diverges almost instantly. However, even without other auxiliary information, it is possible to keep the system informed about a reasonable scale of velocity. In our model, we present a simple yet powerful pseudo-update formulation that keeps the speed in the range of some meters per second and discourages the system from accelerating into higher velocities.

The pseudo-update model is defined in terms of the speed of the object, when it is not stationary. The speed (the Euclidean norm of the velocity) is

$$h_{\mathrm{pseudo}}(\mathbf{x}) = \|\mathbf{v}\|. \tag{11}$$

In our experiments the pseudo-updates are parametrized as follows. The speed observation $y = 0.75$ m/s with a measurement noise $\gamma = \mathrm{N}(0, 2^2)$. The large measurement noise variance keeps the update non-informative in comparison to other information sources.

F. Barometer Readings

Barometric air pressure data (typically also available in high-end smartphones) can be mapped to heights through linearization of the barometric formula around sea level. Over short time periods the air pressure at a given altitude tends to stay constant. In this case the barometer readings relative to the starting point can provide absolute height updates (corresponding to position fixes as presented above).

The barometric pressure drifts over longer time horizons (in the order of tens of minutes), leading to accumulation of measurement errors. Another approach is to only use the relative pressure differences between two consecutive barometer observations mitigating drift issues. This alternative corresponds to opening an altitude loop-closure point on each barometer observation and closing them on the next.

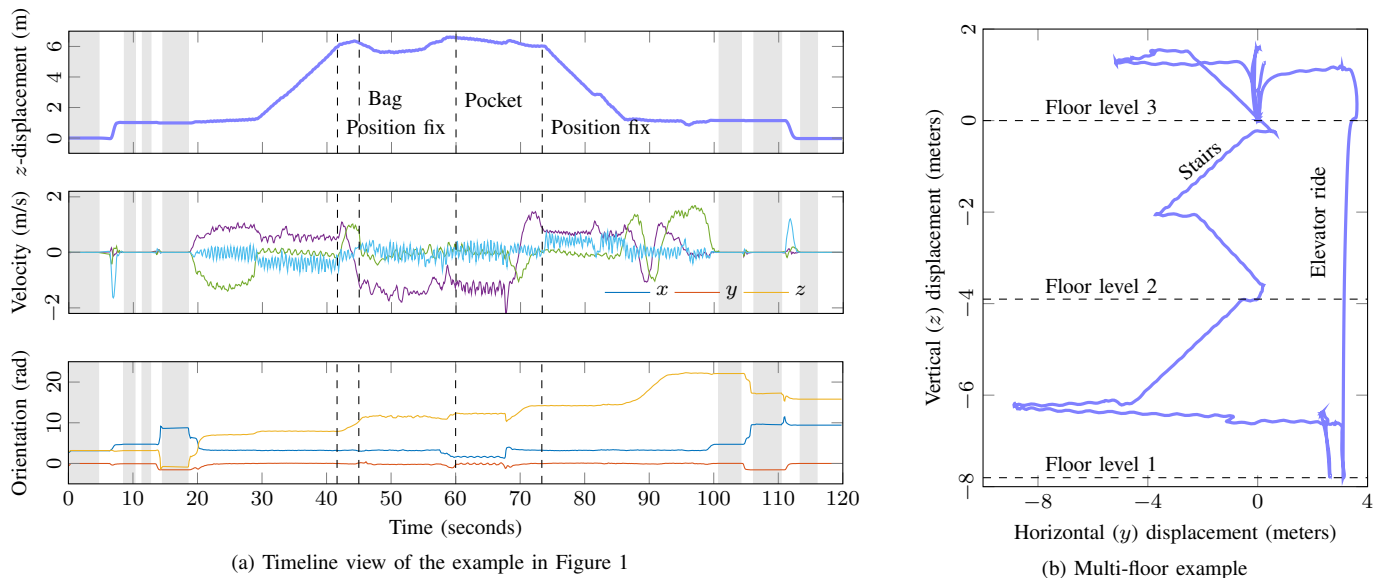(a) Timeline view of the example in Figure 1

(b) Multi-floor example

Fig. 2. (a) The altitude (vertical) profile, the velocities, and the orientations of the phone along the path in Figure 1. The shading shows the stationarity detection outcome, where zero-velocity updates were triggered. The subtle periodicity in the path is due to walking, and the drop in altitude on the first floor is because of the phone being in a bag for parts of the path. (b) A PDR example with first descending two levels down and then taking the elevator back up. The path was started at origin and the path ends with a loop-closure in the same place. The points where the phone touches the floor level (the sharp drops in vertical position) are zero-velocity updates. No absolute position info was given to the model.

IV. EXPERIMENTS

In the examples, the interest was put on Apple phones and tablets—mostly because of their uniform hardware and good software compatibility between devices. The device models used in the examples are the iPhone 6 and the iPad Pro (12.9-inch model). Both these models are equipped with built-in IMUs (InvenSense MP67B) and a barometric sensor (Bosch Sensortec BMP280). In all experiments, the IMU sensor data and the associated timestamps were collected at 100 Hz, and the barometer data (when used) at approximately 0.75 Hz. The data was collected using an in-house developed data collection application, and the paths were reconstructed on the iPhone hardware off-line after the data acquisition.

A. Pedestrian Dead-Reckoning

The most apparent use case for the presented model is to apply it to pedestrian dead-reckoning, where the mobile phone (iPhone 6) is carried by the user indoors. There exist a multitude of methods for dead-reckoning using data provided by mobile phones. Therefore the aim of this experiment is to show how this method differs from others by its generality.

Figures 1 and 2(b) summarize features of the proposed INS system; the example includes the use of zero-velocity updates, position fixes, pseudo-measurements constraining the speed, and barometer observations. This experiment covers traditional navigation-like PDR use cases (walking with the phone in a fixed orientation), where SHS systems are often used, cross-floor tracking, where visual tracking methods are usually the method of choice, and bag/pocket use cases, which currently often require resorting to radio based positioning. The generality of our INS system can cover them all with only one method and no external hardware.

In the first example, the path was started on the ground floor with zero-velocity updates (no pre-calibrations done). First the user walked up a flight of stairs to the first floor holding the phone in the hand. On the first floor a position fix was given, after which the phone was put in a bag. Next, the phone was taken out of the bag and put in the trouser pocket. Before descending to the ground floor, the phone was taken out of the pocket and a second uncertain position observation was given, which aligned the path to the map and was able to provide absolute information of the scale. On the ground floor a manual loop-closure was given to indicate that the phone had returned to the starting point, and the phone was placed on the floor for some final zero-velocity updates. The tracking path is accurate and follows the true path up to decimetres.

Figure 2(a) shows the altitude profile of the path. The ZUPTs where the phone is placed on the floor are clearly showing, as well as the stair climbing. The drop in altitude on the first floor is due to the phone being in the bag for a part of the path. The figure also shows the estimated velocity. The periodicity is due to walking. This effect is less evident when the phone is in the bag, and at clearest when the phone is in the trouser pocket firmly attached to the body.

We briefly present a second PDR example which is shown in Figure 2(b). In this example the path was started at origin with zero-velocity updates in different phone orientations. After this the user walked two floors down. When waiting for the elevator on floor level 1, further ZUPTs were done. The path is completed with taking the elevator back to floor level 3 and closing the loop at the starting point. In this example no absolute position information was given. The scale comes entirely from the accelerometer data. In both examples, an backward smoother pass (see Sec. III-A) is run after every

Fig. 3. The experiment setup with a wheeled baby pushchair/stroller and an iPhone placed on the top for tracking.
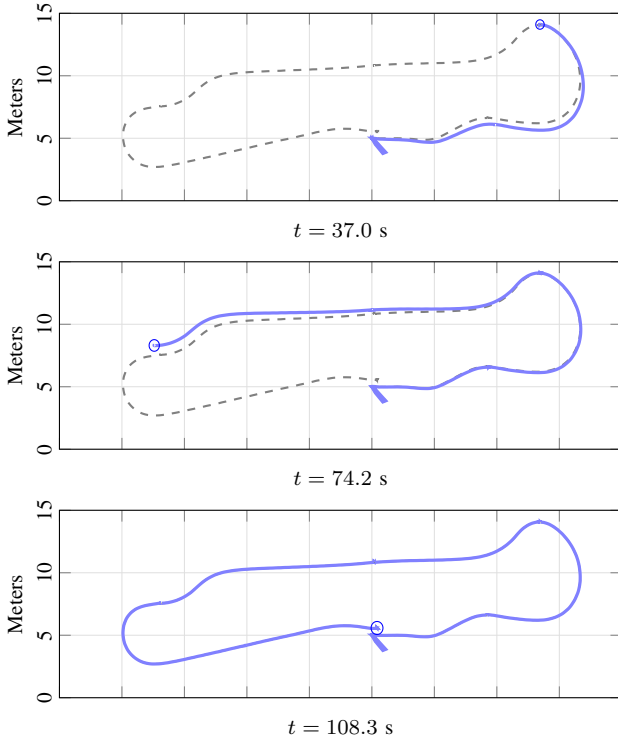

Fig. 4. The evolution of the position estimate and path are visualized along the way at those points where the pushchair was momentarily stopped. The dashed line shows the final path for reference. The phone remained leaning on the top for the entire experiment. No position fixes nor loop-closures were used.

update, thus also correcting the past estimates.

These expeiments demostrate the unconventional nature of the proposed method; this odometry method delivers a combination of use cases, which cannot be delivered with other methods running on the same device. Visual methods fail in the bag/pocket, and SHS methods fail when the device orientation is not fixed or steps/motion cannot be observed.

B. Generalized Dead-Reckoning

Wheel based motion and general non-legged motion are use cases which are not covered by conventional step counting PDR methods. We now set the method in a more general scope for general dead-reckoning that can be applied to any wheeled, sliding, or flying objects indoors or outdoors. Applications include push-carts, trolleys, robots, hover boards, quadcopters, *etc.* We include an example with a human manoeuvred wheeled object with an intrinsic noise source—that is a baby pushchair/stroller with a baby on board. Figure 3 shows the test setup, where the phone is placed leaning on the top. The phone (iPhone 6) remains fixed to the pushchair body for the entire experiment.

Walking was started from a stationary state, where the phone automatically performed ZUPTs. Along the route the pushchair was stopped irregularly and ZUPTs triggered if it became stationary enough. INo position fixes nor loop-closures were used. The only measurement data are the automatic zero-velocity updates, the barometer observations, and pseudo-updates constraining the momentary speed. The total path length was ∼93 m. Figure 4 shows the path estimate at the times when the pushchair was stopped. The dashed line is the path estimate at $t = 108.3$ s which is shown for reference. The zero-velocity updates in the various heading angles of the pushchair are clearly enough to capture the bias estimates and stabilize the system. The final estimate is off from the starting point by ∼0.73 m (0.78%). As the phone orientation is fixed to the pushchair and only moves in a plane, this use case is well suited for SHS sytems [11]. For comparison we implemented a 2D odometry method combining movement detection with turn rates projected to the horizontal plane [3]. The final estimate is only 1.80 m (1.94%) off from the starting point at the end, which is good for an SHS method.

C. Comments on Computational Complexity

The odometry method was implemented in C++ with wrappers in Objective-C for running on the device. The implementation uses the Eigen matrix library. The computational complexity scales linearly with the number of data points, meaning a constant computational burden per sample.

For development purposes, the method was run on the device hardware, but not online. For example, running the odometry for the track in Figure 4 (108.3 s of data) took 0.30 s on the iPhone 6 hardware (single-threaded). Thus the method is capable of running in a real-time application.

D. Sensing of Surroundings

The model proposed in this paper has many potential applications beyond simple odometry and tracking. For example, the orientation and ZUPT information can be used as a measurement tool per se. By consecutively placing the phone flat on the walls of a room, a model of the geometry of the room can be built. From each ZUPT on the trajectory, a wall can be projected parallel to the phone screen, thus capturing the geometry of the room. Associating several ZUPTs to the same wall with the additional knowledge that the points span

a plane through the space, can also make it possible to better estimate the wall placement and orientation.

The model is flexible enough that new constraints—in form of estimated quantities and prior information about them—can be introduced. For this particular application there are several useful constraints, such as coplanarity between some ZUPT positions. A similar smartphone application that delivers these functionalities is publicly available. Therefore we seek to deliver comparable results to the RoomScan (Locometric Ltd, http://locometric.com) application.

Conventional loop-closures are not suited for this particular purpose. In this case the loop-closure points are touching the same plane. This plane is a line in the xy-plane, and the coefficients of the equation of the line for each wall can be augmented in the state vector. Each ZUPT is thus an observation of a point and orientation on a line representing a wall. In our setup, we do not enforce any prior information about walls being orthogonal to each other, whereas we speculate that the RoomScan application enforces some shape constraint for the room.

Markers were placed upon the walls of a room of known geometry (7.30 m × 8.45 m). The phone was moved along the walls stopping for 3 s at each marker until arriving at the starting point, such that the first two markers were visited twice. Figure 5 shows the results obtained by using our model. Measurement #1 was done by stopping at all the available markers, while measurement #2 was done using only every second marker. The resulting rooms are not exactly rectangular, but remarkably close considering no orthogonality constrains were implemented. The estimated size of the room was approximately 7.4 m × 8.3 m for measurement setup #1 and 7.4 m × 8.4 m for setup #2. In both cases the figure shows that in the beginning of the capture the ZUPTs have not matched the wall that well, but the next observations are well matching the wall planes.

For comparison, RoomScan measurements were performed on the same markers and following a similar trajectory between them. The RoomScan application gave a rooms size of 7.3 m × 8.3 m for measurement setup #1 and 7.0 m × 8.6 m for setup #2. This means that the proposed method can deliver comparable results to the black-box method implemented in RoomScan. More testing would be necessary to make an objective comparison, but the purpose of the experiment was to showcase the flexibility and potential uses of the general model presented in this paper.

V. DISCUSSION AND CONCLUSIONS

In this paper we have presented a general framework for inertial navigation using the limited quality data provided by standard handheld smartphones. Up till now, this has been regarded challenging, and we are not aware of any prior published work where the same would have been accomplished.

We presented a probabilistic approach building on extended Kalman filtering for continuous estimation of the position, velocity, and orientation of the mobile device. Furthermore, the IMU sensor biases and scale errors were estimated as

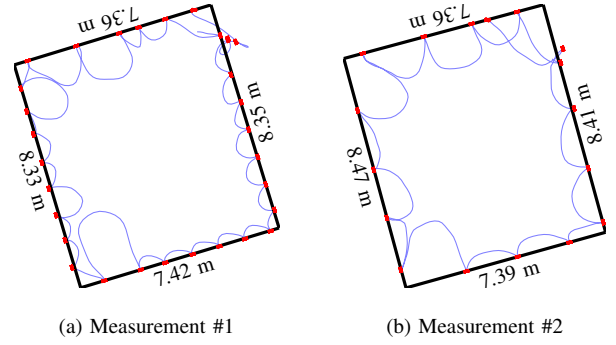(a) Measurement #1   (b) Measurement #2

Fig. 5. Two examples of measuring the wall placements of an indoor space with an iPad Pro. The walls (the equations of the planes) are a part of the state variable. Points where the phone is stationary (against the wall) are visualized in red. The true size of the room is 7.30 m × 8.45 m.

a part of the system state. Our approach differentiates itself from prior models by directly employing the Bayesian (fully probabilistic) interpretation of non-linear state estimation (in the spirit of [27]), and handling the non-additive process noise inside the dynamic model. The estimation scheme avoids unnecessary approximations or error state transformations. Furthermore, we do not assume the sensor sampling rate to be fixed, but use the actual observation timestamps of the sensor events. This helps mitigate problems with missing samples and other unexpected issues with the inputs.

In order to work, the dynamic model needs to be fused with observations. We presented several types of alternative measurements that can be combined with the model. These were position fixes (see Fig. 1), position loop-closures and barometric air pressure data (see Figs. 1 and 2(b)), zero-velocity updates (all examples), and plane tangent observations (Fig. 5). We also introduced constraining the speed estimate from exploding by introducing a pseudo-update for the speed. Even though many of these constraints are not general enough to fit all applications, they still cover many potential use cases.

The presented method has many strong sides. It is general and does not requiring any steps to be detected, specific orientation to be held in, or field of vision to cover any visual features. This differentiates it from conventional PDR and odometry methods for mobile phones. The method is also not limited to estimation in a two-dimensional plane. All these aspects were covered in the experiments, where the phone was held in the pocket, in a bag, on a baby pushchair/stroller, and in an elevator. The last experiment demonstrated how the very same algorithm can be used as a measuring tool for estimating the shape and size of an indoor space. In the PDR experiments we chose to show what the method is capable of as such. While there exists a multitude of well-tailored methods for all of the isolated test scenarios, there are no exact competing methods for mobile phones which could cover all of these scenarios. Implementing separate methods for comparison with respect to each use case was not viable, and we rather chose to put our focus on providing a convincing set of application examples.

In this paper, the data was collected using the mobile device, but the path was calculated off-line. However, the method is lightweight and capable of running in real-time on an iPhone

or iPad (even older models). The computational efficiency comes from the sequential nature of the data processing, which scales linearly in the number of sensor samples.

The method still has some challenges and room for improvement. This kind of inertial navigation systems either work very well or fail miserably (*i.e.* diverge)—there is no middle ground. Therefore handling of the noise scales and biases are crucial for success. Estimation of the sensor biases requires some auxiliary information to be fused with the model—be that ZUPTs, loop-closures, position fixes, or something else. Even though ZUPTs can be implemented to be performed subtly in the background (*e.g.* when the user places the phone on the table), there are use cases which might be problematic. Even though, it has been argued that estimating the sensor biases as a part of the state would not be useful [8], our experiences are quite the contrary. However, this requires the model to be both derived and implemented in a stable way avoiding unnecessary approximations in the error propagation.

The model is also sensitive to the noise scale parameters. The results in this paper benefit from the good sensors (*e.g.* good dynamic range) in the Apple devices. High-end Android phones show comparable results. On Android devices the sampling rate can be set higher, which benefits the modelling (conventional strapdown INS use thousands of Hz, see [10]).

In indoor positioning and tracking the INS presented in this paper could serve as a PDR replacement. The requirement for the zero-velocity updates could perhaps be loosened if the model would receive external position estimates based on Wi-Fi, BLE, RFID, or magnetic field anomalies.

Supplementary material for this paper available on: https://aaltovision.github.io/handheld-INS/

REFERENCES

[1] C. Jekeli, *Inertial Navigation Systems with Geodetic Applications*. Berlin, Germany: Walter de Gruyter, 2001.

[2] K. R. Britting, *Inertial Navigation Systems Analysis*. New York: Wiley-Interscience, 2010.

[3] S. Särkkä, V. Tolvanen, J. Kannala, and E. Rahtu, "Adaptive Kalman filtering and smoothing for gravitation tracking in mobile systems," in *Proceedings of IPIN*, 2015, pp. 1–7.

[4] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proceedings of ICORR*, 2011, pp. 1–7.

[5] V. Renaudin and C. Combettes, "Magnetic, acceleration fields and gyroscope quaternion (MAGYQ)-based attitude estimation with smartphone sensors for indoor pedestrian navigation," *Sensors*, vol. 14, no. 12, pp. 22 864–22 890, 2014.

[6] D. Sachs, "Sensor fusion on Android devices: A revolution in motion processing [Accessed Sep 29, 2016]," https://www.youtube.com/watch?v=C7JQ7Rpwn2k, 2010.

[7] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Computer Graphics and Applications*, vol. 25, no. 6, pp. 38–46, 2005.

[8] J.-O. Nilsson, D. Zachariah, I. Skog, and P. Händel, "Cooperative localization by dual foot-mounted inertial sensors and interagent ranging," *Journal on Advances in Signal Processing*, vol. 2013, no. 1, pp. 1–17, 2013.

[9] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: Wiley-Interscience, 2001.

[10] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*. The Institution of Electrical Engineers, 2004.

[11] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.

[12] J.-O. Nilsson, A. K. Gupta, and P. Händel, "Foot-mounted inertial navigation made easy," in *Proceedings of IPIN*, 2014, pp. 24–29.

[13] O. J. Woodman, "Pedestrian localisation for indoor environments," Ph.D. dissertation, University of Cambridge, Cambridge, UK, September 2010.

[14] W. Kang and Y. Han, "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization," *Sensors*, vol. 15, no. 5, pp. 2906–2916, 2015.

[15] X. Yuan, S. Yu, S. Zhang, G. Wang, and S. Liu, "Quaternion-based unscented Kalman filter for accurate indoor heading estimation using wearable multi-sensor system," *Sensors*, vol. 15, no. 5, pp. 10 872–10 890, 2015.

[16] G. Chen, X. Meng, Y. Wang, Y. Zhang, P. Tian, and H. Yang, "Integrated WiFi/PDR/smartphone using an unscented Kalman filter algorithm for 3D indoor localization," *Sensors*, vol. 15, no. 9, pp. 24 595–24 614, 2015.

[17] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Robust pedestrian dead reckoning (R-PDR) for arbitrary mobile device placement," in *Proceedings of IPIN*, 2014, pp. 187–196.

[18] M. Li, B. Kim, and A. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling shutter camera," in *Proceedings of ICRA*, 2013, pp. 4712–4719.

[19] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *Transactions on Robotics*, vol. 30, no. 1, pp. 158–176, 2014.

[20] M. Bloesch, S. Omari, M. Hutter, and R. Y. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proceedings of IROS*, 2015, pp. 298–304.

[21] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, "PIVO: Probabilistic inertial-visual odometry for occlusion-robust navigation," in *Proceedings of WACV*, 2018, pp. 616–625.

[22] P. Mirowski, T. K. Ho, S. Yi, and M. MacDonald, "Signal-SLAM: Simultaneous localization and mapping with mixed WiFi, Bluetooth, LTE and magnetic signals," in *Proceedings of IPIN*, 2013, pp. 1–10.

[23] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proceedings of MM*, 2010, pp. 787–790.

[24] A. R. J. Ruiz, F. S. Granja, J. C. P. Honorato, and J. I. G. Rosas, "Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements," *Transactions on Instrumentation and Measurement*, vol. 61, no. 1, pp. 178–189, 2012.

[25] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä, "Modeling and interpolation of the ambient magnetic field by Gaussian processes," *IEEE Transactions on Robotics*, 2018.

[26] A. Solin, S. Särkkä, J. Kannala, and E. Rahtu, "Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning," in *Proceedings of ENC*, 2016, pp. 1–9.

[27] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[28] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *Journal of the American Statistical Association*, vol. 74, no. 366, pp. 427–431, 1979.