# *Positioning Algorithms for Indoor Navigation Using Sensors Fusion*

Ievgen Gorovyi, Feliks Sirenko, Alexey Roienko, Yevhen Chervoniak
IT-Jim
Kharkov, Ukraine
ceo@it-jim.com, sirenkofelix@gmail.com, alexey.rnk@gmail.com, eugenecher94@gmail.com

***Abstract* – It is known, that nowadays almost every indoor positioning and navigation system (IPNS) consists of a radio signals part (Wi-Fi or BLE) and a part based on smartphone inertial sensors. Both parts contain a number of challenges complicating a precise user positioning using mobile phones or tablets. In the paper, we describe several contributions. Firstly, a problem of BLE packets recovering is considered. A specific version of a Kalman filter for received signal strength indicator (RSSI) data analysis is developed. The proposed modification allows recovering lost data as well as providing sufficient signal smoothing. Secondly, a custom step detection procedure based on an inertial navigation system (INS) is developed. Unlike to a common solution based on the thresholding of linear acceleration amplitude, an advanced version of the detector is highlighted. Finally, a hybrid indoor localization and navigation (HILN) system developed on the basis of a particle filter (PF) and the proposed modifications for BLE and INS parts is described. Experimental results are provided.**

***Keywords: indoor navigation; BLE beacon; Kalman filter; inertial sensor; Particle filter; RSSI.***

## I. INTRODUCTION

Indoor navigation is a cutting-edge problem and has no unanimous verdict. Many solutions have been proposed since the introduction of Apple iBeacon protocol in 2014. Such big companies like Estimote, Infsoft, Senion and others offer their software development kits (SDK) and mobile applications for building the IPNS. To a greater or lesser extent, a modern indoor navigation system has a part for processing radio signals, usually from BLE beacons, and INS, which incorporates data from wearable sensors such as accelerometer (A), gyroscope (G) and magnetometer (M) [1-3].

All known techniques for INS implementation are based on a step-based pedestrian dead-reckoning (PDR) algorithm or its numerous modifications. At the first stage, the step is detected using different techniques [4]. Next, the step length is evaluated [3-5] followed by the attitude and heading estimation by either Madgwick, Mahoney or Kalman algorithm. Finally, sensor readings are transformed from the local coordinate system (CS) to a global one using either rotation matrices or quaternions [3]. After all, the detected steps are summed up to get a user track in the building. The most problems faced with the INS algorithms are bias and noise of sensor readings. It makes the straight double integration approach impossible to track the pedestrian. The noise turns into a real problem for the step detection algorithm with a constant boundary or its numerous modifications, because the noise causes numerous false step detections. Every false event turns into 0.7-0.8 m error in the pedestrian position [5]. Thus, special signal processing is obligatory for sensor readings and in order to decrease false responses more advanced step detector, which takes into account all possible properties of a step signal pattern, should be applied.

BLE or Wi-Fi radio signals are not inertial, but considerably suffer from fluctuations. The main approach to the BLE positioning is based on the trilateration. It requires at least three beacons near the user to calculate the position. In addition to the RSSI fluctuations another problem, found from the real-life IPNS operation, is the fact that not all packets are usually received. Moreover, the number of received packets depends greatly on smartphone manufacturer and model, beacons manufacturer, whether the user moves or is in a steady state. To our best knowledge, there is no information about this peculiarity in the literature. The missing packets result in permanent hopping of the triplets (three beacons selected to run the trilateration). The considerable fluctuations of the RSSI result in the fact that the exact position can never be determined, i.e. the user can be localized in some area only. Kalman filtering approach is often applied for smoothing RSSI variations [6] or processing user coordinates at the output of the IPNS radio part [7]. But, to our best knowledge, the solution for the task of processing the missing RSSI packets has not been published yet.

Note that the most modern indoor navigation systems fuse the BLE and INS approaches for their mutual improvement. Such systems are known as HILN systems [2, 8]. They are drift-free, low-cost, light-weight, easy-to-integrate inertial positioning systems, enabling ubiquitous navigation of pedestrians in buildings equipped with beacons or Wi-Fi spots. A particle filter (PF) is often chosen as an algorithm for fusion of INS data and IPNS radio part output [2, 8].

The paper's structure is the following. At first, the main features of the BLE navigation approach are discussed. Modification of Kalman filter for the task of recovering of missing RSSI values is proposed in Section 2. Section 3 is devoted to the description of the proposed step detector and its main features. Finally, HILN system designed on the basis of the particle filter is discussed and the comparative analysis of three kinds of IPNS, namely, BLE-based, INS and HILN, is performed.

978-1-5386-5635-8/18/$31.00 ©2018 IEEE

## II. BLE NAVIGATION

### A. Shortcomings of BLE beacon signals

There are a number of parameters which can be used in the radio part of IPNS. They are the received signal strength, received and propagation time, etc. [1]. Among them, the first parameter described by RSSI value is used most often. The higher the RSSI value in the received BLE or Wi-Fi packet, the stronger is the signal and, hence, the closer a user is to the beacon. The main RSSI drawback is its considerable fluctuations that may reach 5-8 dBm. After applying the pass-loss model such a deviation can be interpreted as 3-12 m of distance error. In order to cope with the abovementioned shortcomings, the Kalman filter for RSSI signals is often applied for smoothing the data [6].

Another difficulty in IPNS radio part connects with the application of trilateration algorithm. It is quite obvious that the input to this algorithm must contain the information about three beacons selected in a proper way. Let us analyze the real-life so-called BLE packets map, which represents the dependence of RSSI values received from each BLE beacon by a smartphone upon time. The packet map recorded by the Samsung Galaxy Note 5 for the case when the user stays immovable in the test room surrounded by nine Sensoro BLE beacons is shown in Figure 1. Beacons advertising interval was equal to 417.5 sec. From the initial map of received RSSI values (Fig. 1a) it is clearly seen that there are missing packets outlined with dotted lines. Such packets must be understood as the packets, that due to different reasons were not received by the smartphone, despite they were sent by the beacons.

In the considered approach, three beacons with the highest RSSI values are selected to run the trilateration. The gaps on the map result in misselection of the beacons and, as a result, to a user position discontinuous change. For example, in Fig. 1a the selected beacons at the first step are beacons #1, #6 and #5, at the second – #1, #4 and #5, at the third – #6, #5 and #7, which makes no sense at all because the user did not move and it is reasonable to expect the same selected beacons at each time moment. Different selected beacons are caused by the missing packets problem (step 2 - packets from beacons #2 and #6 are missed; step 3 - packets from beacon #1 and #4 are missed).

### B. RSSI packets recovering using Kalman filter

The described RSSI signal shortcomings can be overcome by the Kalman filter for RSSI [6]. The model of the process used in the Kalman filter is

$$\mathbf{X}_i = \mathbf{A} \cdot \mathbf{X}_{i-1} + \mathbf{O}_i \tag{1}$$

where $\mathbf{X}$ is a state vector equaled to $\mathbf{X}_i = (\mathrm{RSSI}_i\ \Delta\mathrm{RSSI}_i)^T$, $\Delta\mathrm{RSSI}_i = \mathrm{RSSI}_i - \mathrm{RSSI}_{i-1}$, denotes the RSSI change within the beacon advertising interval, $\mathbf{A}$ is a transfer matrix, $\mathbf{A} = \begin{pmatrix} A_1 & A_2 \end{pmatrix}^T$, $A_1 = \begin{pmatrix} 1 & \Delta t_i \end{pmatrix}$, $A_2 = \begin{pmatrix} 0 & 1 \end{pmatrix}$, $\Delta t_i = t_i - t_{i-1}$, $\mathbf{O}$ is a process noise vector $\mathbf{O}_i = (v_i^{RSSI}\ \ \Delta v_i^{RSSI})^T$, which is assumed to be drawn from a zero-mean multivariate normal distribution.

The model of the measurements is

$$\mathbf{Z}_i = \mathbf{H} \cdot \mathbf{X}_i^{meas} + \mathbf{O}_i^{meas} \tag{2}$$

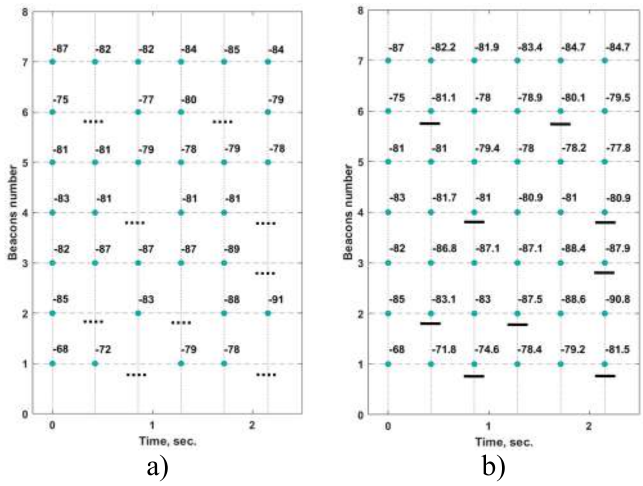



Figure 1 – RSSI packets map before (a) and after (b) Kalman filtering: dotted lines – missed packets, solid lines – recovered packets

where $\mathbf{X}_i^{meas}$ denotes the measured state vector for the $t_i$ time sample, $\mathbf{H} = (1\ \ 0)$ is a correspondence matrix and $\mathbf{O}_i^{meas} = (v_i^{RSSImeas})$ is a measured noise vector which is assumed to be zero-mean Gaussian white noise.

As described in [6], Kalman filter can smooth the RSSI fluctuations, but in the proposed modification it also restores the missing packets with high confidence. For doing this we use the algorithm shown in Fig. 2. The signal from each beacon has its own version of Kalman filter. When there is no packet from one of the beacons, Kalman gain vector, $\mathbf{K}_i$, for the corresponding filter is set up to zero. As a result, the corrected RSSI value, $\mathbf{X}_i^{corr}$, at the filter output is entirely determined by the predicted RSSI value, $\mathbf{X}_i^{pred}$, according to the filter model (1).

As you can see in Figure 1b, at first, there are no missing packets after filtering and, the second, beacons #1, #5 and #6 will be selected for trilateration in almost all timestamps. I.e. the beacons are properly chosen as well as their readings are smoothed.

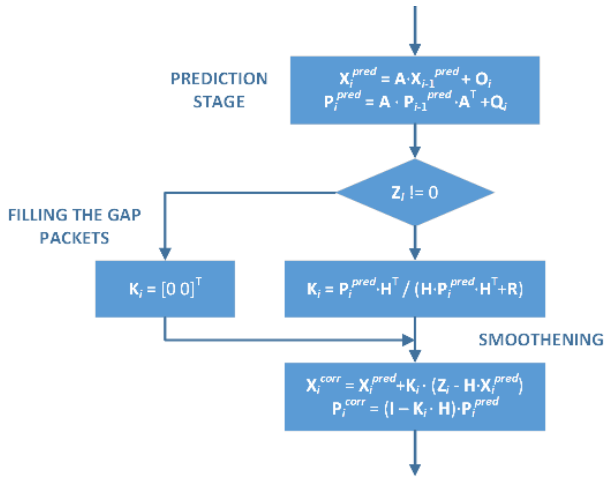



Figure 2 – The flow of the Kalman filter modification for RSSI signals: **Q** and **R** are the process covariance matrix and observation noise matrix, respectively

## III. INS PART OF THE IPNS

INS relies on the data captured by the microelectromechanical system (MEMS) sensors. The mobile devices nowadays are accomplished with a wide range of them [3, 4]. The navigation approach, which was used for our IPNS implementation, addresses the readings from only A, M and G sensors, which are necessary for attitude and heading algorithms as well as step detector [2-4, 8]. The readings of the sensors are usually spoiled with noise but the most considerable problem of the INS part, dealt with the evaluation of position increments, arrives when one must detect the step event.

The physical nature of the problem is caused by the differences in the gait of men and women [9], in the place where the device is kept during the track (handheld, in the pocket, phoning, dangling etc.) in the proximity of obstacles [10]. Numerous proposed step detectors, for example [4, 10], are very unlikely to have the commercial future due to different reasons. The step detectors with the constant threshold are too sensitive to anthropometric statistics of the pedestrian, his/her gait and the walking location (flat surface or stairs). The method with the characteristic points, [10], requires high-precision sensors with the high-quality output signals to distinguish the patterns. Moreover, such patterns correspond to the case when the smartphone is in the pocket.

Due to the described shortcomings of the overviewed algorithms, new step detector based on an adaptive threshold and the amplitude analysis was designed. It analyzes the magnitude of acceleration $a_{abs}$. When the pedestrian is not moving, then $a_{abs}(i)$ oscillates around its average value $a_{avg}$. When the pedestrian starts moving, the algorithm is going to detect the characteristic sinusoidal oscillations. If the new measurement is obtained, the $a_{abs}$ average value is updated:

$$a_{avg}(i) = (1-\omega_{avg}) \cdot a_{avg}(i-1) + \omega_{avg} \cdot a_{abs}(i) \quad (3)$$

where $\omega_{avg}$ is a coefficient characterizing the impact of current measurement on the acceleration average value. In this way, the step detector by its own updates the $a_{avg}$ value despite the initial guess. The step start is detected when the next condition turns true:

$$a_{abs}(i) > a_{avg}(i) \;\&\; a_{abs}(i-1) < a_{avg}(i) \;\&\; t_{start} = 0 \quad (4)$$

where $t_{start}$ is a time when the step start had been detected (by default, the parameter equals to zero). The step end is detected when the next condition is true:

$$\begin{aligned}&\left[a_{abs}(i) > a_{avg}(i)\right] \& \left[a_{abs}(i-1) > a_{avg}(i)\right] \& \\ &\mathrm{t}_{start} \neq 0 \;\&\; \left[a_{abs}^{\max}(s) - a_{abs}^{\min}(s) > C_A \cdot A(s-1)\right]\end{aligned} \quad (5)$$

where $s$ is an integer value and denotes the step counter, $a_{abs}^{\max}(s)$ and $a_{abs}^{\min}(s)$ are the maximum and minimum A values in a current step $s$, $A(s)$ is the acceleration amplitude value at the beginning of the current step $s$, $C_A$ is a tuning factor. The amplitude $A(s)$ is recalculated every step as

$$A(s) = (1-\omega_{amp}) \cdot A(s-1) + \omega_{amp} \cdot \left[a_{abs}^{\max}(s) - a_{abs}^{\min}(s)\right] \quad (6)$$

where $\omega_{amp}$ is a coefficient characterizing an impact of current amplitude to the average amplitude value. Every step must pass the verification procedure:

$$t(i) - t_{start} > C_t \cdot T_{avgST}(s-1) \quad (7)$$

where $t(i)$ is the time moment when the step end had been detected, $T_{avgST}(s\text{-}1)$ is an average time of the step at the beginning of the current step, $C_t$ is a tuning factor. The average time is recalculated every verified step as

$$T_{avgST}(s) = (1-\omega_t) \cdot T_{avgST}(s-1) + \omega_t \cdot (t(i) - t_{start}) \quad (8)$$

where $\omega_t$ is coefficient characterizing an impact of a current time interval to the average time.

The example of the step detection operation is shown in Figure 3 for the test case when the user was moving straight ahead and held the phone in the hand at the chest height. 30 steps were made and 30 step events were counted by the proposed step detector. Note that the first and the last steps differ from the other ones and it is rather difficult to detect them due to the smaller values of the acceleration signals.

## IV. HILN NAVIGATION

The hybrid IPNS fuses INS and BLE-based system or INS and Wi-Fi-based system for better user position estimation [1, 2, 8]. No doubts, that the INS component of a HILN system is very accurate for up to 1-minute term. However, such period of time is not typical for the navigation. Hence, it sounds reasonable to expect the INS to have corrections from time to time to eliminate the accumulated drift. The data for the correction is typically obtained from the non-inertial systems, like BLE-based, Wi-Fi-based, or even ID card terminals [1, 2, 8].

There are three known methods for fusing the inertial and non-inertial navigation systems. The first one implements the feedback filter weighting the position evaluated using BLE or Wi-Fi signals and by INS [8]. This method is rather inefficient and its navigation precision rarely breaks the 3 m boundary.

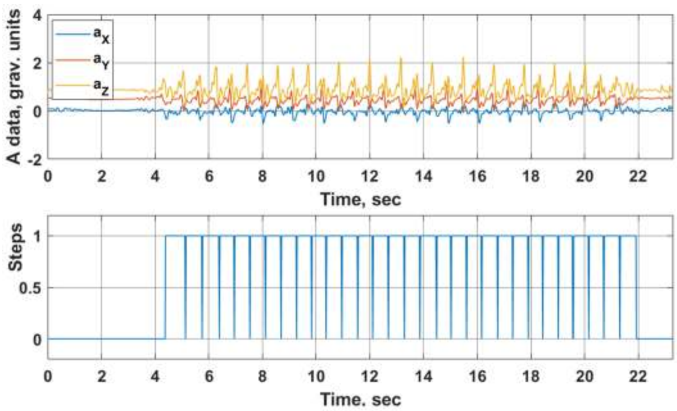



Figure 3 – The example of accelerometer signals (top) and operation of designed step detection application (bottom)

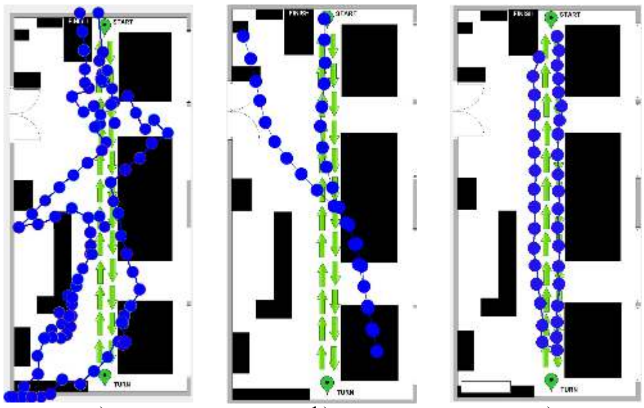
Kalman filtering is a very powerful fusion tool able to automatically determine the trust rates to different sources. However, the monolith structure of the filter makes its modification to be a very complex task [6]. The most prospective method for the fusion is PF. It was first proposed in 1996 in [11] and since that time takes a considerable portion of cases that relate to Markov processes.

The PF has three stages that happen every iteration and one stage that happens once (in the ideal case) or several times. The filtering starts with seeding the first generation of particles which may cover all map when there is no information about the user position, or just a part of the map when some very approximate position is known. The initial information for seeding can be obtained from the radio part of HILN. At the next step, the initial generation of the particles is subjected to a displacement according to the INS part. The positions of all particles are updated based on the step length at each moment of time.

At the second step, the particles must get weights, which are calculated according to the known information about the map with its black (forbidden for navigation) and white (allowed for navigation) regions and the position determined by the beacon. If a particle at the $i^{th}$ step is in the forbidden map region or crosses the forbidden map region while relocating from the $(i-1)^{th}$ step to the $i^{th}$ step, then this particle is considered to be dead and its weight becomes equal to zero. The closer the particle to the position determined by BLE part, the greater its weight. At this step, the correction of inertial disturbances happens, because the most drifted particles get the minimal weights.

The last stage of the algorithm is called resampling. There are numerous techniques for performing this step [11]. The number of particles at this stage must be brought back the initial amount, while the higher probability of getting to a new generation belongs to particles with higher weights.

Finally, Figure 4 represents the results obtained by three IPNS types with the ground-truth marked with arrows. The modification of Kalman filter for lost packets recovering as well as designed step detector were applied in proper system parts. It is clearly seen that there is a discontinuous change of a user position and quite low positioning accuracy for the BLE-based component of IPNS (Fig. 4a), however the main direction of a user movement is observable. One may observe a trajectory drift caused by the residual noise of the G for INS-based navigation in Fig. 4b. As was expected, the best performance is shown by designed HILN (Fig. 4c), which provides very accurate positioning with no visible track drifting in time. In all experiments held with hybrid IPNS, the accuracy positioning error varied in the range 0.5 to 1m m on an area of 15×6 meters, which is competitive with the leading commercial solutions.



a) b) c)

Figure 4 – Example of user position estimation performed by BLE-based IPNS (a), INS-based navigation system (b) and HILN system on the basis of PF (c)

## V. Conclusions

We have presented several ideas allowing to improve the indoor positioning accuracy. In particular, specifically constructed Kalman RSSI filter properly restores the lost beacon packets and suppress the signal fluctuations. A novel step detector allows to control user's movement and avoid a lot of false positives. Finally, the proposed fusion scheme gives noticeable effect on mobile navigation system efficiency.

## Acknowledgment

The authors would like to thank Björn Wedler (VisionLab GmbH & Co. KG) for financial support of research activities and mobile SDK creation.

## References

[1] Streich, Katja . (2017, Julie 13). Technologies for Server-Based Indoor Positioning Compared: Wi-Fi vs. BLE vs. UWB vs. RFID [Web site]. Available: https://www.infsoft.com/blog-en/articleid/188/technologies-for-server-based-indoor-positioning-compared

[2] Tian, Qinglin et al. "A Hybrid Indoor Localization and Navigation System with Map Matching for Pedestrians Using Smartphones." Ed. Kourosh Khoshelham and Sisi Zlatanova. Sensors (Basel, Switzerland) 15.12 (2015): 30759–30783. PMC. Web. 14 Mar. 2018.

[3] Tian Q., Salcic Z., Wang K.I.-K., Pan Y. A Multi-Mode Dead Reckoning System for Pedestrian Tracking Using Smartphones. IEEE Sens. J. 2016;16:2079–2093. doi: 10.1109/JSEN.2015.2510364.

[4] Ho, N.-H., Truong, P. H., & Jeong, G.-M. (2016). "Step-Detection and Adaptive Step-Length Estimation for Pedestrian Dead-Reckoning at Various Walking Speeds Using a Smartphone. Sensors" (Basel, Switzerland), 16(9), 1423. http://doi.org/10.3390/s16091423

[5] "Walking with Attitude - Article - How to Measure Stride or Step Length for your Pedometer", Walkingwithattitude.com, 2018. [Online]. Available: https://www.walkingwithattitude.com/articles/features/how-to-measure-stride-or-step-length-for-your-pedometer. [Accessed: 08-Apr- 2018].

[6] "What is the Kalman Filter and How can it be used for Data Fusion?" Robotics Math 16-811 Sandra Mau December 2005

[7] Seoung-Hyeon Lee, "Method for Improving Indoor Positioning Accuracy Using Extended Kalman Filter," Hindawi, Daejeon 34129, Republic of Korea, 2016.

[8] Tian, Q., Salcic, Z., Wang, K. I.-K., & Pan, Y. (2015). A Hybrid Indoor Localization and Navigation System with Map Matching for Pedestrians Using Smartphones. Sensors (Basel, Switzerland), 15(12), 30759–30783. http://doi.org/10.3390/s151229827

[9] H. Si, "Normal gait characteristics under temporal and distance constraints.," *J. Biomed Eng,* no. 11(6):449, p. 56, Nov 1989.

[10] Gusenbauer D., "Self-Contained Indoor Positioning on Off-The-Shelf Mobile Devices.," *Proceedings of IEEE 2nd Conference on Indoor Positioning and Indoor Navigation,* p. 15–23, 15-17 September 2010.

[11] Del Moral, Pierre (1996). "Non-Linear Filtering: Interacting Particle Solution". Markov Processes and Related Fields. 2 (4) pp. 555–580.