Free-walking 3D pedestrian large trajectory reconstruction from IMU sensors

Haoyu Li, Stéphane Derrode, Lamia Benyoussef, Wojciech Pieczynski

HAL Id: hal-01870621
https://hal.archives-ouvertes.fr/hal-01870621

Free-walking 3D Pedestrian Large Trajectory Reconstruction from IMU Sensors

Haoyu Li, Stéphane Derrode
École Centrale de Lyon, Univ. Lyon
LIRIS, CNRS UMR 5205, France.
haoyuli1990@gmail.com, stephane.derrode@ec-lyon.fr

Lamia Benyoussef
EPITA Lyon, France
lamia.derrode@epita.fr

Wojciech Pieczynski
Telecom SudParis, Univ. Paris-Saclay
SAMOVAR, CNRS UMR 5157, France
wojciech.pieczynski@telecom-sudparis.eu

Abstract—**This paper presents a pedestrian navigation algorithm based on a foot-mounted 9-DOF Inertial Measurement Unit (IMU), which provides tri-axial accelerations, angular rates and magnetics. Most of algorithms used worldwide employ Zero Velocity Update (ZUPT) to reduce the tremendous error of integration from acceleration to displacement. The crucial part in ZUPT is to detect stance phase precisely, a cyclic left-to-right style Hidden Markov Model is introduced in this work which is able to appropriately model the periodic nature of signals. Stance detection is then made unsupervised by using a suited learning algorithm. Then orientation estimation is performed independently by a quaternion-based method, a simplified error-state Extended Kalman Filter (EKF) assists trajectory reconstruction in 3D space, neither extra method nor prior knowledge is needed to estimate the height. Experimental results on large free-walking trajectories show that the proposed algorithm can provide more accurate locations, especially in z-axis compared to competitive algorithms, *w.r.t.* to a ground-truth obtained using OpenStreetMap.**

Index Terms—**Pedestrian navigation, Inertial sensor, Hidden Markov model, Kalman filter, Stance detection.**

I. INTRODUCTION

In recent years, Pedestrian Navigation System (PNS) using a foot-mounted IMU sensor has been investigated extensively. IMUs have great advantages as they are small and can be worn on the body. It measures motion kinematic information such as acceleration, angular rate and magnetics, so that it is theoretically possible to transfer the signals from the sensor frame to the earth frame, also called "global frame", based on the sensor orientation, and then to compute the velocity and displacement of motion. Unfortunately, a simple numerical integration of acceleration to obtain the displacement suffers from a tremendous error, which is mainly caused by accelerometer noise and numerical integration. Therefore, ZUPT was then developed and showed great performances in reducing the cumulative double integration errors. The idea behind ZUPT is the following: when a person is walking, his feet alternatively swing in the air and step on the ground, so that his foot velocity is zero when it is attached to the ground. The period during which the velocity is zero is called "stance phase". Foxlin [1] firstly proposed an error-state EKF-aided ZUPT algorithm for PNS, which exploits the periodical stance phases to reduce both velocity and displacement errors for a much better precision of position. From this work, numerous strategies were then proposed that mainly enrich the measurement vector of EKF, by using for examples angular rate [2] or digital compass [3].

State-of-the-art algorithms on using an EKF-aided ZUPT algorithm allow to precisely reconstruct the trajectory in X-Y plane, also called horizontal plane, but fail in reconstructing the z-axis as they have huge bias in estimating the height. So getting reliable reconstructed trajectory in 3D space requires extra method for robust height estimation. Y. Hsu *et al.* [4] used a Probabilistic Neural Network (PNN) classifier to identify and count the number of steps climbed. The height can then be estimated assuming that the height of each stair is known, which is limited. Y. Liu *et al.* [5] proposed a floor identification method for height estimation, based on a prior knowledge of buildings. SK Park *et al.* [6] proposed a height compensation algorithm by estimating road inclination angle. Also, we can also cite fusion algorithms of inertial sensor data with extra sensors (barometer: [7], [8], and WI-FI: [9]). Hence, a large majority of works depend on some prior knowledge of the environment or on some additional sensors to assist height estimation.

This paper proposes a new EKF-aided ZUPT based 3D PNS algorithm with only one 9-DOF IMU sensor. By introducing a decomposed Gradient Descent Algorithm (GDA) based quaternion estimation method, the walking orientation of sensor is computed independently from the displacement. Stance phases are then detected by a cyclic Left-to-Right Hidden Markov Model (LR-HMM), so that the EKF-aided ZUPT allows to compensate for the integration error. The precision of the algorithm has been evaluated through long walks down the valley. The remaining of the paper is organized as follows. Section II describes the sensor used and the structure of the proposed PNS algorithm. Section III then explains how the independent quaternion estimation method is developed and works within the PNS. Section IV describes the LR-HMM based stance detection method, and a simplified error-state EFK is introduced. Results of experiments are proposed in Section V.

II. PEDESTRIAN NAVIGATION SYSTEM

The PNS is based on one 9-DOF IMU sensor from Shimmer Sensing [1] with 100Hz sampling rate. The scale of ac-
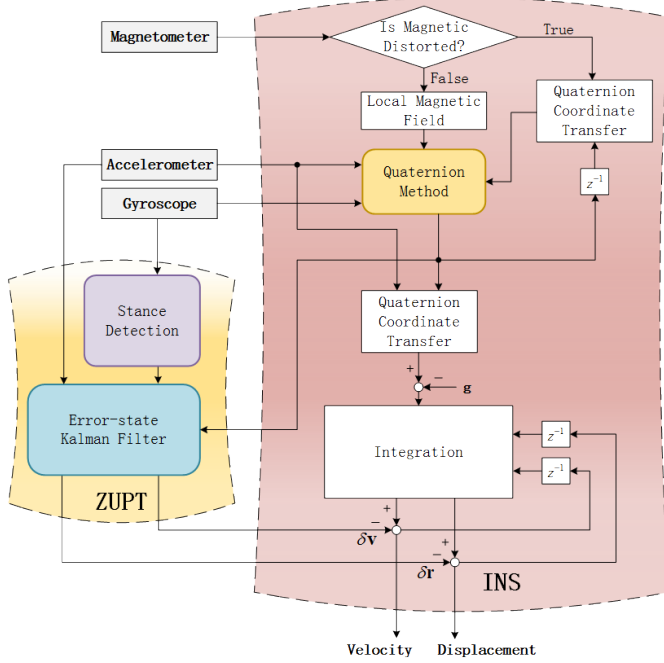

Figure 1: Up: Shimmer3 sensor and its placement on a shoe. Down: Structure of PNS algorithm.

celerometers, gyroscopes and magnetometers are set to $\pm 8\mathbf{g}$, $1000deg/s$ and $2.5Ga$ respectively. The data is stored in the embedded SD card and retrieved after experiments. The sensor is placed on a foot as shown in Fig. 1. The trajectory is represented in the North-West-Up (NWU) earth frame, and the original point of the earth frame is set to the starting point of walking.

The entire PNS algorithm follows the framework of Foxlin [1], see Fig. 1. It is mainly composed of two modules, the ZUPT and INS (Inertial Navigation System):

— The ZUPT module takes data collected from the accelerometers and gyroscopes. Stance detection is firstly operated. Then the error-state EKF updates the velocity error $\delta\boldsymbol{v}$ and displacement error $\delta\boldsymbol{r}$ according to the result from stance detection.

— The INS module has two main steps. First, it computes orientation by a quaternion method, which is based on a decomposed GDA. Second, it transfers the measured acceleration from the sensor frame (s) to the earth frame (e), for the use of integration to obtain velocity and displacement.

At last, velocity and displacement are compensated by $\delta\boldsymbol{v}$ and $\delta\boldsymbol{r}$ from ZUPT module. Compared to many works of PNS

[1] [4] [10], the main difference comes from the separation of orientation estimation with the error-state EKF, which will provide more precision to the trajectory reconstruction, as explained in Section III.

III. Inertial Navigation System

Let us detail some aspects of the INS.

A. Magnetic Disturbance Detect

The magnetic field can be easily affected by other magnetic sources. Therefore, it is necessary to detect magnetic disturbance before estimating orientation. If $^e\boldsymbol{m} = [^em_x, ^em_y, ^em_z]^\mathsf{T}$, the measured magnetic field in the earth frame coordinates, satisfies the two criterions below, then the magnetic measurement cannot be used directly (see below).

$$\frac{|\,^e\boldsymbol{m} - ^e\boldsymbol{b}\,|}{|\,^e\boldsymbol{b}\,|} \geq \varepsilon \tag{1}$$

$$\left| \frac{\arctan\left(^em_x/^em_z\right) - \arctan\left(^eb_x/^eb_z\right)}{\arctan\left(^eb_x/^eb_z\right)} \right| \geq \varepsilon, \tag{2}$$

in which the earth magnetic field $^e\boldsymbol{b} = [^eb_x, 0, ^eb_z]^\mathsf{T}$ is supposed to be known. The threshold ε is set to 0.2 in this work.

B. Orientation Estimation

Let $^s_e\hat{\boldsymbol{q}} = [\hat{q}_0, \hat{q}_1, \hat{q}_2, \hat{q}_3]^\mathsf{T}$, $|\,^s_e\hat{\boldsymbol{q}}\,| = 1$, denotes an estimated quaternion used to transfer coordinates from the earth frame to the sensor frame. A simple way to compute quaternion is integrating from its differential $^s_e\dot{\boldsymbol{q}}$, which is computed from angular rate. In this work, quaternion from the simple integration is denoted as $^s_e\boldsymbol{q}_{\boldsymbol{\omega},n}$, $\boldsymbol{\omega}$ and n respectively indicate the angular rate and time epoch. Quaternion is related to Euler angles, $ie.$ yaw ψ, pitch θ and roll ϕ, [11] introduced the conversion from quaternion to Euler angles, $(\hat{\psi}, \hat{\theta}, \hat{\phi}) = h\left(^s_e\hat{\boldsymbol{q}}\right)$.

Since gyroscopes have drift and noise, the quaternion integration error will inevitably increase as time elapses. So, [12] propose a GDA-based quaternion estimation method as a guess to reduce the drift, if the time sampling interval is small enough. Their method is based on two objective functions related to acceleration and magnetic, $f_{\mathbf{g}}$ and $f_{\boldsymbol{b}}$ respectively, and on their corresponding Jacobian, $J_{\mathbf{g}}$ and $J_{\boldsymbol{b}}$ respectively.

$f_{\boldsymbol{b}}$ and $J_{\boldsymbol{b}}$ use the normalized earth magnetic field as a reference of north, the x-axis of the earth frame. If no disturbance is detected, the local earth magnetic field is referred to as the x-axis of the earth frame. Once a disturbance is confirmed by the method in section III-A at time epoch n, it regards $^e\boldsymbol{m}_n|_{^s_e\hat{\boldsymbol{q}}_{n-1}}$ as the reference of the x-axis of the earth frame, $^e\boldsymbol{m}_n|_{^s_e\hat{\boldsymbol{q}}_{n-1}}$ represents the magnetic measurement at time n that transferred into the earth frame by $^s_e\hat{\boldsymbol{q}}_{n-1}$. It is because that magnetic measurement will consist of extra magnetic resources once $^e\boldsymbol{b}$ is interfered, a change in yaw angle is easier to be estimated if the new combined magnetic field is referred to as the x-axis of the earth frame. Therefore, transferring disturbed magnetic measurement back to the earth frame according to $^s_e\hat{\boldsymbol{q}}_{n-1}$ will be an appropriate option.

The basic GDA-based quaternion estimation method merges the two objective functions into one. This may provide good estimation when the sensor is static or moving slowly, but may have bias when there is a large acceleration. It is because that the objective function for acceleration regards the normalized acceleration measurement as a reference of gravity $\mathbf{g}$, different extents of motions may lead to different gradient values even if the objective function for magnetic is the same. It means that $f_{\mathbf{g}}$ is influenced by acceleration and angular rate, but $f_{\boldsymbol{b}}$ is only influenced by angular rate. Therefore, the gradient quaternion estimation needs to be computed separately:

$$
{}_e^s\hat{\boldsymbol{q}}_{\nabla_{\mathbf{g}},n} = {}_e^s\hat{\boldsymbol{q}}_{n-1} - \mu_n \frac{J_{\mathbf{g}}^T f_{\mathbf{g}}}{\| J_{\mathbf{g}}^T f_{\mathbf{g}} \|}, \tag{3}
$$

$$
{}_e^s\hat{\boldsymbol{q}}_{\nabla_{\boldsymbol{b}},n} = {}_e^s\hat{\boldsymbol{q}}_{n-1} - \mu_n \frac{J_{\boldsymbol{b}}^T f_{\boldsymbol{b}}}{\| J_{\boldsymbol{b}}^T f_{\boldsymbol{b}} \|}, \tag{4}
$$

where $\mu_n = \alpha \parallel {}_e^s\dot{\boldsymbol{q}}_n \parallel \Delta t$, $\alpha > 1$ is a ratio for the gradient, Δt is the sampling interval. ${}_e^s\hat{\boldsymbol{q}}_{\nabla_{\mathbf{g}},n}$ and ${}_e^s\hat{\boldsymbol{q}}_{\nabla_{\boldsymbol{b}},n}$ are quaternion estimations based on GDA, it is reasonable because Δt is small and quaternion does not change a lot within a tiny period. A data fusion method is then used to combine the estimations from eq. (3) and (4):

$$
{}_e^s\hat{\boldsymbol{q}}_{\mathbf{g,n}} = \rho_{\mathbf{g,n}} \cdot {}_e^s\hat{\boldsymbol{q}}_{\nabla_{\mathbf{g}},n} + (1 - \rho_{\mathbf{g,n}}){}_e^s\boldsymbol{q}_{\boldsymbol{\omega},n} \tag{5}
$$

$$
{}_e^s\hat{\boldsymbol{q}}_{\boldsymbol{b},n} = \rho_{\boldsymbol{b},n} \cdot {}_e^s\hat{\boldsymbol{q}}_{\nabla_{\boldsymbol{b}},n} + (1 - \rho_{\boldsymbol{b},n}){}_e^s\boldsymbol{q}_{\boldsymbol{\omega},n} \tag{6}
$$

$$
\rho_{\mathbf{g,n}} = \frac{\beta_{\mathbf{g}}}{\mu_n/\Delta t + \beta_{\mathbf{g}}}, \quad \rho_{\boldsymbol{b},n} = \frac{\beta_{\boldsymbol{b}}}{\mu_n/\Delta t + \beta_{\boldsymbol{b}}}, \tag{7}
$$

${}_e^s\hat{\boldsymbol{q}}_{\mathbf{g,n}}$ and ${}_e^s\hat{\boldsymbol{q}}_{\boldsymbol{b},n}$ are the estimated quaternions *w.r.t* acceleration and magnetic, $\beta_{\mathbf{g}}$ and $\beta_{\boldsymbol{b}}$ represent the divergence rate of ${}_e^s\boldsymbol{q}_{\boldsymbol{\omega},n}$ based on the extent of motion (a larger value means a more intensive motion). Hence, they are dependent on the value of acceleration and angular rate. $\rho_{\mathbf{g,n}}$ and $\rho_{\boldsymbol{b},n}$ are the weights yield to μ_n and $\beta_{\mathbf{g}}$, $\beta_{\boldsymbol{b}}$.

Since pitch θ and roll ϕ depend on the estimation of gravity and yaw ψ depends on the estimation of north (the direction of magnetic field), it is possible to extract Euler angles from $(\hat{\psi}_{\mathbf{g}}, \hat{\theta}_{\mathbf{g}}, \hat{\phi}_{\mathbf{g}}) = h({}_e^s\hat{\boldsymbol{q}}_{\mathbf{g,n}})$ and $(\hat{\psi}_{\boldsymbol{b}}, \hat{\theta}_{\boldsymbol{b}}, \hat{\phi}_{\boldsymbol{b}}) = h({}_e^s\hat{\boldsymbol{q}}_{\boldsymbol{b},n})$, then use the pitch and roll represented by ${}_e^s\hat{\boldsymbol{q}}_{\mathbf{g,n}}$ and yaw represented by ${}_e^s\hat{\boldsymbol{q}}_{\boldsymbol{b},n}$ to get the final quaternion estimation:

$$
{}_e^s\hat{\boldsymbol{q}}_n = h^{-1}\left(\hat{\psi}_{\boldsymbol{b}}, \hat{\theta}_{\mathbf{g}}, \hat{\phi}_{\mathbf{g}}\right). \tag{8}
$$

IV. Zero-Velocity Update

In ZUPT, stance detection is firstly performed for finding the stance phase, then the error of acceleration, velocity and displacement is updated by an error-state EKF, and at last the estimated error is used for compensating integration error during every stance.

A. Stance Detection

Generally, one walking step is broken down into four phases [13]: stance, push-up, swing and step-down. The four phases switch one after one and repeat periodically, thus we introduce a LR-HMM to represent the walking model, in which the state can only stay the same or switch to the next state.

Let's assume a hidden Markov chain model with observations $\boldsymbol{Y} = \{Y_1, \ldots, Y_N\}$, each $Y_n \in \mathbb{R}^3$, and with unknown states $\boldsymbol{X} = \{X_1, \ldots, X_N\}$, each $X_n = k \in \Omega = \{1, \ldots, 4\}$. Ω represents the stance, push-up, swing and step-down phases respectively. Assuming a discrete time independent Markov process, $\boldsymbol{X}$ can be parametrized by an initial probabilities vector $\boldsymbol{\pi} = p(x_1)$ and a transition matrix $\boldsymbol{A}(x_1, x_2) = p(x_2|x_1)$. In a cyclic LR-HMM, the transition matrix only allows states to keep the same or to switch to the next:

$$
\boldsymbol{A}(x_1, x_2) = \begin{bmatrix} 1 - \Delta_2 & \Delta_2 & 0 & 0 \\ 0 & 1 - \Delta_3 & \Delta_3 & 0 \\ 0 & 0 & 1 - \Delta_4 & \Delta_4 \\ \Delta_1 & 0 & 0 & 1 - \Delta_1 \end{bmatrix}, \tag{9}
$$

where $\Delta_k = p(x_n = k | x_{n-1} = k - 1)$ denotes the transition probability from state $k - 1$ to state k.

The observation $y_n = ({}^s\omega_{xn}, {}^s\omega_{yn}, {}^s\omega_{zn})$ comes from the measurement of gyroscopes, which represents the angular rate along the three axes of the sensor frame. The distributions of observations conditional to states are assumed to be Gaussian

$$
p\left(y_n \mid x_n = k\right) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{10}
$$

where $\boldsymbol{\mu}_k$ (3×1 vector) and $\boldsymbol{\Sigma}_k$ (3×3 matrix) are the mean and co-variance of observations corresponding to state k. So that the LR-HMM model is parametrized by the following set of parameters $\boldsymbol{\Theta} = \{\pi_k, \Delta_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k \in \Omega}$. Then the state transition probability $p(x_n|x_{n-1})$ and observation probability conditioned on state $p(y_n|x_n)$ can be easily derived by eq. (9) and (10).

Learning of $\boldsymbol{\Theta}$ parameters can be done by Baum-Welch algorithm [14], which is based on the EM principle (Expectation-Maximization) for finding the maximum likelihood iteratively, starting from an initial guess $\boldsymbol{\Theta}^{(0)}$ of parameters and stopping after a criterion or a maximum number of iterations is reached.

B. Error-State Extended Kalman Filter

Once the stance phase is successfully detected in Section IV-A, velocity is assumed to be zero and the velocity integration error can be obtained easily, unlike the displacement integration error which needs to be estimated appropriately. Based on the work of [10], an EKF was used for estimating the integration error of velocity and displacement, so the elements in the state of EKF are the integration error and the EKF used here is also called as error-state EKF. In this work, we presents a simplified error-state EKF, since orientation estimation is performed independently in Section III-B, the error-state only takes into account the acceleration error ($\delta^s \boldsymbol{a}_n$), velocity error ($\delta^e \boldsymbol{v}_n$) and displacement error ($\delta^e \boldsymbol{r}_n$):

$$
\delta\boldsymbol{\eta}_n = \delta\boldsymbol{\eta}_{n|n} = \begin{bmatrix} \delta^e \boldsymbol{r}_n^{\mathsf{T}}, & \delta^e \boldsymbol{v}_n^{\mathsf{T}}, & \delta^s \boldsymbol{a}_n^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}. \tag{11}
$$

The error-state model is the one of [10], $\delta\boldsymbol{\eta}_n = \boldsymbol{\Phi}_n\delta\boldsymbol{\eta}_{n-1} + \boldsymbol{w}_n$, where $\boldsymbol{w}_n$ represents the process noise with covariance

matrix $\boldsymbol{Q}_n = E\left(\boldsymbol{w}_n \boldsymbol{w}_n^\mathsf{T}\right)$ and where the error-state transition model $\boldsymbol{\Phi}_n$ is a 9×9 matrix given by

$$\boldsymbol{\Phi}_n = \begin{bmatrix} I_{3\times3} & \Delta t \cdot I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & \Delta t \cdot {}_s^e C_n \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix}. \tag{12}$$

$\boldsymbol{\Phi}_n$ is time-variant and depends on the value of ${}_s^e C_n$, which represents the rotation matrix required to convert vectors from the sensor frame to the earth frame at time n [11].

Now, the measurement equation writes $\boldsymbol{z}_n = \boldsymbol{H}\delta\boldsymbol{\eta}_{n|n} + \boldsymbol{\nu}_n$, where $\boldsymbol{z}_n$ is the measurement, $\boldsymbol{H} = [0_{3\times3}, I_{3\times3}, 0_{3\times3}]$ is the measurement model, and $\boldsymbol{\nu}_n$ is the measurement noise, assumed as Gaussian with covariance $\boldsymbol{R}_n = E\left(\boldsymbol{\nu}_k \boldsymbol{\nu}_k^\mathsf{T}\right)$.

When the stance phase is detected, the error-state measurement in stance phase is $\boldsymbol{z}_n = {}^e\boldsymbol{v}_n - [0,0,0]^\mathsf{T}$ (zero represents the real velocity), then the displacement integration error can be estimated because of the correlation between $\delta^e\boldsymbol{v}_n$ and $\delta^e\boldsymbol{r}_n$ described in the error-state model.

The error-state is predicted at every time, while is only updated during stance phases, since measurements are only available during stance phases. The prediction and update computation is illustrated in [15]. Then the velocity and displacement can be compensated through ${}^e\boldsymbol{v}_n - \delta^e\boldsymbol{v}_n$ and ${}^e\boldsymbol{r}_n - \delta^e\boldsymbol{r}_n$ respectively, $\delta^e\boldsymbol{r}_n$ and $\delta^e\boldsymbol{v}_n$ in error-state should be reset to zero once the compensation is completed.

V. EXPERIMENTAL RESULTS

An experiment was conducted on a road nearby the campus of École Centrale de Lyon, Écully, France. The ground truth were obtained using OpenstreetMap, the total travel distance is $1075m$ with a walking speed of about $4.2km/s$. The magnetic sensor was calibrated manually using the calibration toolbox proposed by Alex Faustino [2].

Before starting to walk, a short time of standing without motion is necessary for initializing the quaternion corresponding to the earth frame, and the local magnetic field was obtained by transferring the mean value of magnetic measurements to the earth frame. The magnetic declination at Écully is $1.2°$. The LR-HMM method is tested and compared to another threshold based stance detection method detailed in [16]. Parameters of both algorithms are learned or tuned to get the best results.

The total steps number in experiment is 1458, so the steps number of one foot is 729. Compared with the threshold based stance detection method (see Fig. 2), LR-HMM obtains a more regular stance pattern, detecting rarely a false negative detection or missing rarely one step. Indeed, the missing and false negative detection number by LR-HMM are 0 and 6 (0.82% in total) respectively, while the numbers by threshold method are 17 and 14 (1.92% in total) respectively. Stance phases detected by LR-HMM have similar durations, which is not the case for the threshold-based method.

The proposed PNS algorithm involves LR-HMM to detect stance phase, and the decomposed GDA-based quaternion estimation (GDA$_d$) and simplified error-state EKF (EKF$_s$)

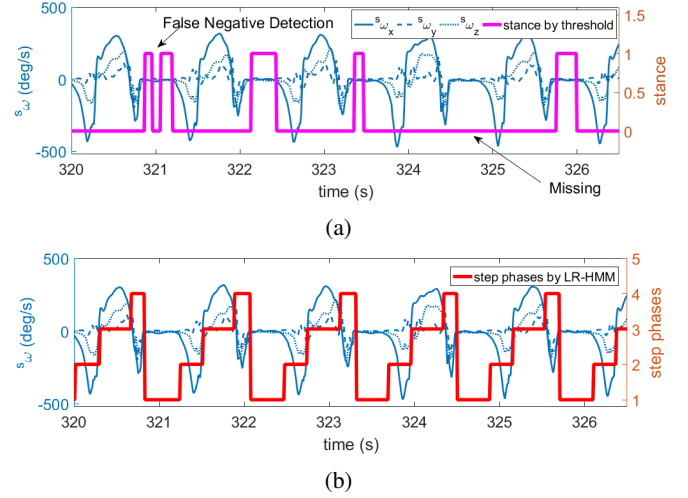2. Github repository: https://github.com/alex-faustino/Mag-Cal

(a)

(b)

Figure 2: Step segmentation, jump 1 represents the stance phase. Stance detection by (a) a threshold method, where false negative detection and missing ones can be observed. (b) the proposed LR-HMM method.

to reconstruct trajectory. Comparative algorithms include the threshold stance detection, basic GDA-based quaternion estimation (GDA$_b$), a full error-state EKF (EKF$_f$) that contains compass measurement as described in [10], and Zero Angular Rate Update (ZARU). Various combinations of these algorithms are tested and compared to the proposed algorithm, denoted here by LR-HMM+GDA$_d$+EKF$_s$+ZUPT.

All the parameters needed in the algorithms are tuned and learned for the best results, particularly $\beta_\mathbf{g}$ and $\beta_\mathbf{b}$ are set to 0.0215 and 0.026 respectively. In Fig. 4 and Table I, the proposed algorithm reconstructs a travel distance of $1298.9m$, with an end-to-end error of $18.07m$, the position error is 1.68%. It can be seen that independent orientation estimation provides a good accuracy in height, the proposed algorithm giving a final height error of $11.41m$. GDA$_d$-based quaternion estimation leads to a better result than the GDA$_b$, which means that the decomposed GDA algorithm proposed in this paper can provide a much better dynamic performance than the basic
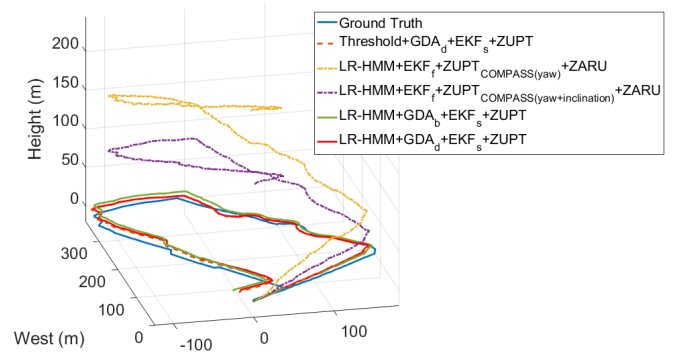

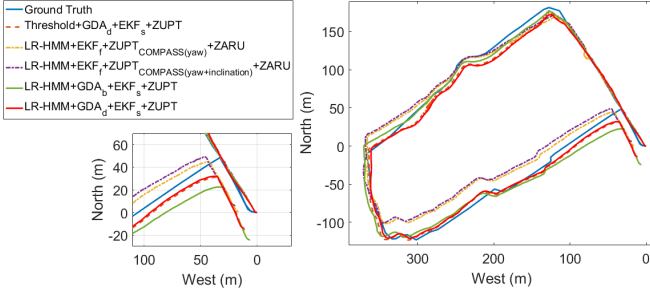Figure 3: Reconstructed 3D trajectory by different algorithms.

Figure 4: Reconstructed trajectory in X-Y plane by different algorithms.

Table I: PNS Trajectory Error.

	Algorithms[a]					Ground Truth
	①	②	③	④	⑤	
End-to-End Error (m)	18.45	16. 11	19.47	25.06	18.07	0
Height Error (m)	9.81	241.39	145.10	13.84	11.14	0
Total Distance in XY Plane (m)	1376.9	1199.6	1180.8	1303.1	1224.5	1075
Total Distance in 3D (m)	1464.4	1452.3	1346.9	1387.3	1298.9	1097.6
DTW Distance	1.149E4	16.36E4	9.514E4	1.348E4	1.279E4	–

[a] ①Threshold+GDA$_d$+EKF$_s$+ZUPT,
②LR-HMM+EKF$_f$+ZUPT$_{COMPASS(yaw)}$+ZARU,
③LR-HMM+EKF$_f$+ZUPT$_{COMPASS(yaw+inclination)}$+ZARU,
④LR-HMM+GDA$_b$+EKF$_s$+ZUPT,
⑤LR-HMM+GDA$_d$+EKF$_s$+ZUPT.

GDA algorithm. The threshold-based stance detection has a good precision at the end point. However the highly enlarged travel distance ($1376.9m$ in X-Y plane and $1464.4m$ in 3D space) is explained by some smaller stance phases duration than the truth. Thus more displacement compensation are made because of the unexpected extra integration error, which leads to an extra travel distance. The huge height error by EKF$_f$ is due to that it only takes the yaw angle into consideration, estimation for pitch or roll is not involved. At last, the DTW distance between the reconstructed trajectory and the ground-truth shows that the proposed algorithm reconstructed the best-fitted 3D trajectory.

VI. CONCLUSION

This paper presents a new strategy for PNS algorithm that uses a foot-mounted IMU sensor. Compared to the works proposed before, the orientation estimation is separated from the error-state EKF and a decomposed GDA-based quaternion estimation algorithm is employed for better dynamic performance. Due to the orientation is estimated independently, the error-state EKF can be simplified for reducing the complexity. Then a LR-HMM is used for stance detection for the use of ZUPT algorithm. Finally 3D trajectory is reconstructed by this new PNS algorithm structure.

The experimental results show that LR-HMM detects stance phase accurately and the proposed PNS algorithm reconstructs 3D trajectory accurately without extra method or prior knowledge. Our future work is to improve the dynamic performance of orientation estimation and to allow the algorithm to deal with different walking speeds during the same walking.

REFERENCES

[1] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Comput. graph. and app.*, vol. 25, no. 6, pp. 38–46, 2005.

[2] S. Rajagopal, "Personal dead reckoning system with shoe mounted inertial sensors," Master's thesis, KTH, Sweeden, 2008.

[3] A. Norrdine, Z. Kasmi, and J. Blankenbach, "Step detection for ZUPT-aided inertial pedestrian navigation system using foot-mounted permanent magnet," *IEEE Sensors J.*, vol. 16, no. 17, pp. 6766–6773, 2016.

[4] Y.-L. Hsu, J.-S. Wang, and C.-W. Chang, "A wearable inertial pedestrian navigation system with quaternion-based ekf for pedestrian localization," *IEEE Sensors J.*, vol. 17, no. 10, pp. 3193–3206, 2017.

[5] Y. Liu, M. Ren, P. Wang, and H. Guo, "Height error correction based on shoe-mounted pedestrian navigation systems," in *35th Chinese Control Conf. (CCC)*, July 2016, pp. 5683–5686.

[6] S. K. Park and Y. S. Suh, "Height compensation using ground inclination estimation in inertial sensor-based pedestrian navigation," *Sensors*, vol. 11, no. 8, pp. 8045–8059, 2011.

[7] B. Shin, C. Kim, J. Kim, S. Lee, C. Kee, H. S. Kim, and T. Lee, "Motion recognition-based 3D pedestrian navigation system using smartphone," *IEEE Sensors J.*, vol. 16, no. 18, pp. 6977–6989, Sept 2016.

[8] Y. K. Kim, S. H. Choi, H. W. Kim, and J. M. Lee, "Performance improvement and height estimation of pedestrian dead-reckoning system using a low-cost MEMS sensor," in *12th Int. Conf. on Control, Automation and Systems*, Oct 2012, pp. 1655–1660.

[9] W. Chai, C. Chen, E. Edwan, J. Zhang, and O. Loffeld, "2D/3D indoor navigation based on multi-sensor assisted pedestrian navigation in Wi-Fi environments," in *Ubiquitous Positioning, Indoor Navigation, and Location Based Service*, Oct 2012, pp. 1–7.

[10] A. R. Jiménez, F. Seco, J. C. Prieto, and J. Guevara, "Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU," in *Workshop on Positioning Navigation and Communication (WPNC)*, 2010, pp. 135–143.

[11] L. Wang, Z. Zhang, and P. Sun, "Quaternion-based Kalman filter for AHRS using an adaptive-step gradient descent algorithm," *Int. J. of Advanced Robotic Systems*, vol. 12, no. 9, p. 131, 2015.

[12] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *IEEE Int. Conf. on Rehabilitation Robotics (ICORR)*, 2011, pp. 1–7.

[13] A. Mannini and A. M. Sabatini, "A hidden Markov model-based technique for gait segmentation using a foot-mounted gyroscope," in *Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, Boston, MA, USA, Aug 2011, pp. 4369–4373.

[14] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[15] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006, ch. 2, pp. 123–138.

[16] S. Qiu, Z. Wang, H. Zhao, K. Qin, Z. Li, and H. Hu, "Inertial/magnetic sensors based pedestrian dead reckoning by means of multi-sensor fusion," *Information Fusion*, vol. 39, pp. 108–119, 2018.