Error Analysis of Pedestrian Inertial Navigation Algorithms: L-IONet and ZUPT

Shreya Vaish, Deepshikha, Snehil Saluja, Avinash Bansal, Darshana Singh, Divyansh Shikar, Heetesh Kumar Dabura, Keshav Maheshwari, Navshruti Singh, Nidhi, Nikhil Mishra

INTRODUCTION

Contemporary inertial measurements units[*1] (IMUs) are tiny (usually a few mm2), cheap, energy-saving, eco-friendly, and prevalent micro-electro-mechanical (MEMs[*2]), which are widely utilized in smart devices and mobile robots. These are turning out to be the basis of widespread research for location-based services and motion-sensing because of their low cost and potent sensing modality. Nowadays, smartphones have embedded IMUs, and users use them for different location-dependent services. The various applications can be seen in

- maneuver aircraft, unmanned aerial vehicles (UAVs), spacecraft, satellites, etc.
- wristbands, VR/AR headsets for continuous health monitoring and mesmerizing gaming experiences, and,
- intelligent technology devices and autonomous robot systems.

The reason behind their proliferating popularity is their low dependency on physical infrastructure, environmental factors, and widespread usage as fittings in our devices such as smartphones, which gives them an edge over traditional navigation techniques like GPS, which may not work in areas such as tunnels or with electronic interference where the signals cannot reach.

The growth of IMUs in the above applications is governed by a method known as inertial navigation, which uses the orientation and position of the user based on the acceleration and rotation measurements of IMU sensors. Compared to the high-dimensional visual data of traditional techniques, IMU data are 6-dimensional time series that make computing possible in real-time, even a resource-limited device. Due to the utilization of low-cost IMUs, unbounded system error grows, which becomes an obstacle in acquiring long-term inertial navigation.

But as with any new technology that we come across, along with its merits, specific challenges need to be addressed to fully utilize them as future technology upgrades on a societal range. IMU data tends to accumulate navigational error approximately proportional to time cubed and

can exceed 1m of error within seconds, making it unfit for consumer-level use. So, several algorithms have been developed to convert the data into expected trajectories and to study and reduce the error between data collected by IMUs and the actual data. These algorithms include such as IONet, L-IONet (Light Inertial Odometry Neural Networks), PDRs (Pedestrian dead reckoning systems), and ZUPT (Zero-Velocity-Update). In order to overcome this limitation, one solution is to attach the IMUs on the user's foot and then utilize the benefits of Zero-Velocity-Update (ZUPT).

PDRs are intended to find the trajectories by detecting the user's steps and estimating the heading (i.e., the future steps). However, these classical algorithms are difficult to use in everyday usage since this human motion has some unrealistic assumptions: (i) the solution cannot be used on the consumer's devices as ZUPT demands that the inertial sensors be fixed firmly on the user's foot. (ii)PDRs are dependent on the personal walking models. In contrast, some inertial navigation models such as IONet are based on deep learning and can estimate motion and generate the trajectory of the paths directly from raw material data without using any handcrafted algorithms. Nowadays, we see the emerging interest in applying deep neural networks to understand the motion from time-series data.

The significant challenge generally confronted while developing data-driven ideas is that the ground truth values of location, velocity, and orientation are all needed to train, validate, and test deep neural network models.

LITERATURE REVIEW

2.1

Pedestrian Navigation System (PNS) provides information such as position and orientation of the user has various applications in military and civilian uses. Dependent PNS uses infrastructural requirements such as GPS, LTE, and WIFI signals. But in the unavailability of these assets, self-dependent or wearable PNS come into the picture. How can we reduce the error in this method by changing different variables like the position from where the data should be collected and by using different data processing methods?

We have different studies regarding the improvement of Pedestrian Navigation Systems for data collection, inertial sensors can be mounted on the human hand, waist, shoulder, or foot, where the changes during movement can be recorded. To improve accuracy, use of compact and cheap Micro-Electro-Mechanical Systems (MEMS)-based Inertial Measurement Units (IMUs) are mounted on foot according to the following study, forefoot of boot proved to be a preferable position due to lower shock level and uncertainty in velocity. Sensors are assisted by the ZUPT algorithm to suppress the error and improve the accuracy of the PNS system. A double-stage quaternion-based extended Kalman

filter to fuse acceleration, angular velocity, and magnetic signals in which navigation system was based on course angle IMU's were used for estimating the stride, and magnetic sensors which were placed on the waistband were used to measure direction for improved accuracy.

IMU's are combined with the traditional continuous integration algorithms in indoor navigation systems but these assume a particular position for installation of sensors and are inflexible for different walking patterns. But use of IONet breaks the traditional cycle of continuous error propagation by reconstructing accurate and robust trajectories from raw inertial data which is achievable by the usage of DNN framework such as walking, running also trolly tracking and it outperforms expectations.

IONet brought evolution in the indoor navigation system but it required high error sensors and processors to process high amounts of data evaluations. LIONet, a lightweight neural network, is more memory, inference and training efficient. LIONet also works effectively on cheap hardware. LIONet is able to achieve this by the application of convolutional neural networks.

2.2

From a Study on the Mounting Position of IMU for Better Accuracy of ZUPT-Aided Pedestrian Inertial Navigation, we concluded that this paper is about mounting IMU behind the heel is less preferable as compared to the forefoot in ZUPT-aided pedestrian inertial navigation. Reasons for this are a longer stance phase, reduced shock level, reduced velocity uncertainty throughout the stance phase. This also shows a relation between navigation error and IMU mounting position.

2.3

By Pedestrian Navigation Method Based on Machine Learning and Gait Feature Assistance, we understand that this study puts forward a navigation method based on the construction of a virtual foot IMU, also pedestrian gait feature assistance. Based on the VGG-LSTM neural network model, the establishment of a nonlinear mapping relationship between inertial information from the human foot and leg is done, which leads to the construction of virtual foot IMU and VINS. Based on recurring characteristics of accelerometer and gyroscope output of virtual foot IMU. The results show that the unified method of modifying proposed in the paper can slow down the accumulation of position error in the gait types that exceed inertial sensors measuring range.

2.4

Another great study is Enhanced Pedestrian Navigation Based on Course Angle Error Estimation Using Cascaded Kalman Filters. This study puts forward a pedestrian navigation algorithm that is based on a TCKF. TCKF works on two filters, first one evaluates the course error between the walking course and the magnetic heading of the waist-mounted sensors, second evaluates other navigation errors using the ZUPT. Among both filters, the first works in this way, walking course is forecasted with the help of a position trace which is along the direction of the body, and the magnetic heading of the waist-mounted sensors is used for generating measurements. Now course error is estimated. In the second filter, the calculated error in the course

is used for estimating the heading error of the foot-mounted sensors. It is observed that the heading error of the foot-mounted sensors is the reason for the walking course error. Thus, course error shares various similarities to the heading error of the foot-mounted sensors, given the course of a body is a bit different from the heading of the foot-mounted sensors. Therefore, the proposed TCKF can estimate and correct the heading error indirectly by applying the concept of course error. These outdoor and indoor experiments help attain efficacy and improved performance. The results clearly state that the accuracy of position is improved by a maximum of 90% compared with the conventional ZUPT-based pedestrian navigation algorithm.

OBJECTIVE

This paper aims to cover a few analytical goals regarding the nature of error propagation in IMU data and two algorithms which predict the trajectory error :

1. Find out the extent of systematic and statistical error in trajectory predicted when IMU data processed by ZUPT and related algorithms is used versus raw data.

2. Find out the extent of systematic and statistical error in trajectory predicted when IMU data processed by LIONet deep learning network and related algorithms is used versus raw data.

3. Find out how the nature of error is responsible for variations in errors recorded for different types of movements.

METHODOLOGY

Selection of Algorithms

We selected two inertial navigation models as our baselines: ZUPT and L-IONet.

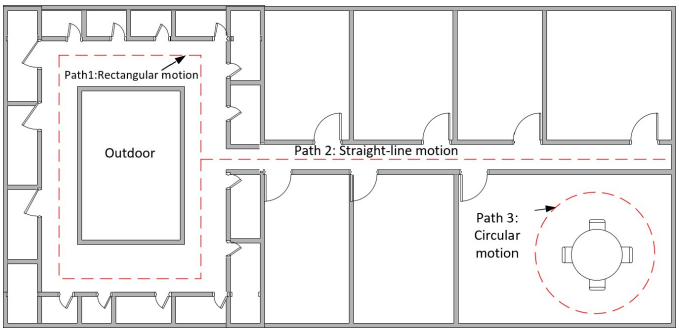
In this paper, the most important source of auxiliary information is so called zero-velocity updates (ZUPT). Once the phone is detected to be stationary for any interval of time, velocity becomes zero and the additive measurement error becomes small. The orientation of the acceleration is tracked with gyroscope measurements. The accelerometer and gyroscope readings are seen as control signals.

In a MEMS-based pedestrian navigation, the knowledge of when stance phase occurs is very important. A few methods of detection are using the magnitude of total acceleration, moving acceleration variance, magnetometer based, neural network based and the magnitude of gyro signals. By comparing these data with a certain threshold, the start and end of the stance phase can be detected. The detection of stance phase is important because this is the only time in an autonomous pedestrian navigation where the data is known beforehand. Velocity of the IMU in its body frame are theoretically zero during this condition and this has allowed the use of ZUPT. Any nonzero measurements during this moment can therefore be assumed to be errors. We also employ Kalman filter which provides a means of approximating the state distributions. The Kalman Filter operates on prediction and update mode. In this paper, the knowledge of velocity errors during ZUPT is used as the measurement updates to KF to better estimate the position solution and the MEMS sensor errors.

We also attempt to implement the Lightweight Inertial Odometry Neural Network (L-IONet), a lightweight framework to learn inertial tracking, which is more efficient in resource and computational consumption using NARX Neural Network.

3.2 Research instrument data and Procurement

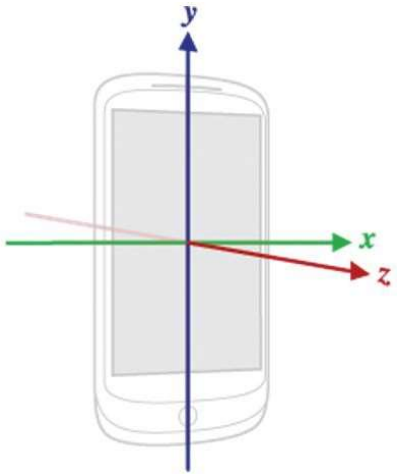
We obtain the data provided by the inertial measurement unit (IMU) in our smartphones using an application named as "Sensorstream IMU+GPS" to continuously infer the relative change in position, velocity and orientation of the device with respect to a starting point. For ZUPT, 6 participants using different smartphones were asked to follow the path of a circle, square, straight line and the infinity symbol.



The frame of reference of the mobile phones coordinate systems is non inertial. The android operating system's rotation matrix R is given by

$$\boldsymbol{R} = \begin{bmatrix} E_x & E_y & E_z \\ N_x & N_y & N_z \\ G_x & G_y & G_z \end{bmatrix}$$

where x, y and z are axes as marked in the figure and $\boldsymbol{E}, \boldsymbol{N}, \boldsymbol{G}$ defined as follows:



$\boldsymbol{E} = (E_x, E_y, E_z)$ = a unit vector that points towards East.

$\boldsymbol{N} = (N_x, N_y, N_z)$ = a unit vector that points towards North.

$\boldsymbol{G} = (G_x, G_y, G_z)$ = a unit vector that points away from the center of the earth

The Euler angles Φ, θ and ψ are expressed as

$$\begin{aligned} \text{yaw} \quad &= \psi = \text{ rotation about } \boldsymbol{G} \\ \text{pitch} \quad &= \theta = \text{ rotation about } \boldsymbol{E}. \\ \text{roll} \quad &= \Phi = \text{ rotation about } \boldsymbol{N} \end{aligned}$$

When the mobile phone is not in the normal position, we need to adjust the tilt to align the coordinate systems of the phone and earth. The rotation matrices of the yaw $(\boldsymbol{R}_\psi)$, pitch $(\boldsymbol{R}_\theta)$, and roll $(\boldsymbol{R}_\phi)$ are given by -

$$\boldsymbol{R}_\psi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{R}_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

$$\boldsymbol{R}_{\phi} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

The transformation of the acceleration data from the mobile phone coordinate system (a_x, a_y, a_z) to the ECEF coordinates (A_x, A_y, A_z) is given by

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \boldsymbol{R}_{(\psi,\theta,\phi)}[a_x \quad a_y \quad a_z]^T$$

Similarly,

For LIONet test and training

Measurement of the accelerometer and gyroscope in x, y and z serve as variables for the ZUPT algorithm.

Column names/variables for LIONet algorithm are as follows:

'imu.csv'(input file):

attitude roll(radians) attitude_pitch(radians) attitude_yaw(radians)
rotation_rate_x(radians/s)
rotation_rate_y(radians/s)
rotation_rate_z(radians/s) gravity_x(G)
gravity_y(G) gravity_z(G)

user_acc_x(G) user_acc_y(G) user_acc_z(G)

magnetic_field_x(microteslas)magnetic_field_y (microteslas) magnetic_field_z(microteslas)

'vi.csv'(output file):

translation.x translation.y translation.z

rotation.x rotation.y rotation.z rotation.w

3.2 Pre-Processing

Data preprocessing is the precursor to any prediction or error calculation algorithm. It converts the data into a state more suitable for the implementation and processing phases that eases the interpretation of the data characteristics. It includes filling in missing entries with workable values. The data in the .csv files was interpolated for null entries in the gyroscope columns using simple spreadsheet manipulation to obtain a continuous stream of data.

3.3 Selection of features for Error Analysis

Gyroscope and Accelerometer bias errors were one of the considered parameters obtained on implementing ZUPT to make inferences. These parameters indicate the amount of error the algorithm predicts as time progresses. It serves as a determinant of systematic error.

Another parameter considered for ZUPT were the covariance graphs in x, y and z. These graphs show how closely the errors in x, y and z are occurring. A graph with high correlation signifies lesser statistical error. Since small values of MSE signify lower statistical error, it was considered a parameter for LIONet.

3.4 Working of ZUPT and LIONet

ZUPT

Mathematically, there are two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ defined as follows:

$\mathcal{H}_0$: mobile phone is moving
$\mathcal{H}_1$: mobile phone is stationary.

The detector performance is then specified by the false-alarm probability,

$$P_{\text{FA}} = \Pr\{\mathcal{H}_1 \mid \mathcal{H}_0\}$$

and the probability of detection

$$P_D = \Pr\{\mathcal{H}_1 \mid \mathcal{H}_1\}$$

Further, Neyman-Pearson theorem can give us the function $L(\mathbf{z}_n)$ is referred to as the likelihood ratio. We will limit ourselves to the case when the combined multisensory navigation of the mobile phone may be described as follows:

$$\mathbf{y}_k = \mathbf{s}_k(\theta) + \mathbf{v}_k$$

where

$$\mathbf{s}_k(\theta) = \begin{bmatrix} \mathbf{s}_k^a(\theta) \\ \mathbf{s}_k^\omega(\theta) \end{bmatrix} \quad \text{and} \quad \mathbf{v}_k = \begin{bmatrix} \mathbf{v}_k^a \\ \mathbf{v}_k^\omega \end{bmatrix}$$

Here, $\mathbf{s}_k^a(\theta) \in \mathbb{R}^3$ and $\mathbf{s}_k^\omega(\theta) \in \mathbb{R}^3$ denote the accelerometer and the gyroscope readings, respectively, θ denotes the set of unknown signal parameters. Moreover, $\mathbf{v}_k^a \in \mathbb{R}^3$ and $\mathbf{v}_k^\omega \in \mathbb{R}^3$ denote the noise associated with the accelerometer-and gyroscope-sensor assembly, respectively. We will assume that the noise is independent identically distributed zero-mean, Gaussian with covariance matrix

$$\mathbf{C} = \mathbb{E}\{\mathbf{v}_k \mathbf{v}_k^T\} = \begin{bmatrix} \sigma_a^2 \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_\omega^2 \mathbf{I}_3 \end{bmatrix}$$

where $\mathbf{I}_3(\mathbf{0}_3)$ denotes an identity (zero) matrix of size 3×3, and T denotes the transpose operation. Further, $\mathbb{E}\{\cdot\}$ denotes the expectation operation, and $\sigma_a^2 \in \mathbb{R}^1$ and $\sigma_\omega^2 \in \mathbb{R}^1$ denote the accelerometer and gyroscope noise variance, respectively.

Modeling the acceleration and angular rates, i.e., $\mathbf{s}_k(\theta)$ experienced by the mobile phone under the hypothesis $\mathcal{H}_0$ in a consistent way that reflects a diverse set of gaits, is difficult. However, under the hypothesis $\mathcal{H}_1$, the following conditions hold true.

1 The acceleration measured by the accelerometers assembly is only due to the ground's standoff to the gravitation acceleration, whose magnitude is known.

2 The attitude of the mobile phone is constant, that is, that the angular velocity is zero.

In mathematical terms,

$$\mathcal{H}_0: \exists k \in \Omega_n \text{ s.t. } \mathbf{s}_k^a(\theta) \neq g\mathbf{u}_n \text{ or } \mathbf{s}_k^\omega(\theta) \neq \mathbf{0}_{3,1}$$
$$\mathcal{H}_1: \forall k \in \Omega_n \text{ then } \mathbf{s}_k^a(\theta) = g\mathbf{u}_n \text{ and } \mathbf{s}_k^\omega(\theta) = \mathbf{0}_{3,1}$$

Here, $\mathbf{u}_n \in \Omega_u$, where $\Omega_u = \{\mathbf{u} \in \mathbb{R}^3 : \| \mathbf{u} \| = 1\}$ and $g \in \mathbb{R}^1$
is the acceleration due to gravity. Also, $\| \mathbf{u} \| = \sqrt{\mathbf{u}^T\mathbf{u}}$ and $\Omega_n = \{\ell \in \mathbb{N} : n \leq \ell < N-1\}$. Hence,

$$\mathcal{H}_0: \theta \equiv \{\mathbf{s}_k\}_{k=n}^{n+N-1}$$
$$\mathcal{H}_1: \theta \equiv \mathbf{u}_n$$

That is, under $\mathcal{H}_0$, the signal $\mathbf{s}_k(\theta)$ is fully not known, whereas under $\mathcal{H}_1$, the direction of the acceleration is unknown. Since we cannot decipher θ, we cannot write the probability density functions of the observations. However, with the equations above, the measurements are given by –

$$p(\mathbf{z}_n;\theta,\mathcal{H}_i) = \prod_{k\in\Omega_n} p(\mathbf{y}_k;\theta,\mathcal{H}_i) = \prod_{k\in\Omega_n} p(\mathbf{y}_k^a;\theta,\mathcal{H}_i)p(\mathbf{y}_k^\omega;\theta,\mathcal{H}_i)$$

where

$$p(\mathbf{y}_k^a;\theta,\mathcal{H}_i) = \frac{1}{(2\pi\sigma_a^2)^{3/2}}\exp\left(-\frac{1}{2\sigma_a^2}\|\mathbf{y}_k^a - \mathbf{s}_k^a(\theta)\|^2\right)$$

and

$$p(\mathbf{y}_k^\omega;\theta,\mathcal{H}_i) = \frac{1}{(2\pi\sigma_\omega^2)^{3/2}}\exp\left(-\frac{1}{2\sigma_\omega^2}\|\mathbf{y}_k^\omega - \mathbf{s}_k^\omega(\theta)\|^2\right)$$

As the pdfs are dependent on θ which is partially not known (signal $\mathbf{s}_k(\theta)$), we cannot apply the Likelihood Ratio Test. However, by substituting it with its maximum likelihood estimate, we can derive a GLRT. The GLRT decides on $\mathcal{H}_1$, if

$$L_G(\mathbf{z}_n) = \frac{p(\mathbf{z}_n;\hat{\theta}^1,\mathcal{H}_1)}{p(\mathbf{z}_n;\hat{\theta}^0,\mathcal{H}_0)} > \gamma$$

where $\hat{\theta}^1$ is the MLE of the unknown parameters assuming $\mathcal{H}_1$ is true, and $\hat{\theta}^0$ is the MLE of the unknown parameters assuming $\mathcal{H}_0$ is true. In our case, under $\mathcal{H}_0$, the signal is unknown and $\hat{\theta}^0 = \{\mathbf{y}_k\}_{k=n}^{n+N-1}$. Thus,

$$p(\mathbf{z}_n;\hat{\theta}^0,\mathcal{H}_0) = \frac{1}{(2\pi\sigma_a^2)^{3N/2}}\cdot\frac{1}{(2\pi\sigma_\omega^2)^{3N/2}}$$

Under $\mathcal{H}_1$, we can obtain the MLE of the unknown signal parameter $\mathbf{u}_n$ by maximizing

$$p(\mathbf{z}_n;\mathbf{u}_n,\mathcal{H}_1) = \frac{1}{(2\pi\sigma_a^2)^{3N/2}}\exp\left(-\frac{1}{2\sigma_a^2}\sum_{k\in\Omega_n}\|\mathbf{y}_k^a - g\mathbf{u}_n\|^2\right)\cdot\frac{1}{(2\pi\sigma_\omega^2)^{3N/2}}\exp\left(-\frac{1}{2\sigma_\omega^2}\sum_{k\in\Omega_n}\|\mathbf{y}_k^\omega\|^2\right)$$

with respect to $\mathbf{u}_n \in \Omega_u$. That is,

$$\begin{aligned}\hat{\mathbf{u}}_n &= \arg\min_{\mathbf{u}\in\Omega_u}\left(p(\mathbf{z}_n;\mathbf{u},\mathcal{H}_1)\right)\\ &= \arg\min_{\mathbf{u}\in\Omega_u}\left(\sum_{k\in\Omega_n}\|\mathbf{y}_k^a - g\mathbf{u}\|^2\right)\\ &= \arg\min_{\mathbf{u}\in\Omega_u}\left(\sum_{k\in\Omega_n}\|\mathbf{y}_k^a\|^2 - 2g(\mathbf{y}_k^a)^T\mathbf{u} + g^2\,\|\,\mathbf{u}\,\|^2\right)\\ &= \arg\min_{\mathbf{u}\in\Omega_u}\left((\bar{\mathbf{y}}_n^a)^T\mathbf{u}\right) = \frac{\bar{\mathbf{y}}_n^a}{\|\bar{\mathbf{y}}_n^a\|}\end{aligned}$$

where

$$\bar{\mathbf{y}}_n^a = \frac{1}{N}\sum_{k\in\Omega_n}\mathbf{y}_k^a$$

This follows from the Cauchy's Schwarz inequality $\mathbf{a}^{\hat{T}}\mathbf{b} \leq \|\,\mathbf{a}\,\|\|\,\mathbf{b}\,\|$, where the equality is observed only if $\mathbf{a} = c\mathbf{b}; c$ is a real constant. Substituting $\hat{\mathbf{u}}_n$:

$$p(\mathbf{z}_n;\hat{\theta}^1,\mathcal{H}_1) = p(\mathbf{z}_n;\hat{\mathbf{u}}_n,\mathcal{H}_1)$$
$$= \frac{1}{(2\pi\sigma_a^2)^{3N/2}}\cdot\frac{1}{(2\pi\sigma_\omega^2)^{3N/2}} \cdot\exp\left(-\frac{1}{2\sigma_a^2}\sum_{k\in\Omega_n}\left\|\mathbf{y}_k^a - g\frac{\bar{\mathbf{y}}_n^a}{\|\bar{\mathbf{y}}_n^a\|}\right\|^2\right) \cdot\exp\left(-\frac{1}{2\sigma_\omega^2}\sum_{k\in\Omega_n}\|\mathbf{y}_k^\omega\|^2\right)$$

Next, if we combine the GLRT becomes:

decide on $\mathcal{H}_1$ if

$$L_G(\mathbf{z}_n) = \exp\left(-\frac{1}{2\sigma_a^2}\sum_{k\in\Omega_n}\left\|\mathbf{y}_k^a - g\frac{\bar{\mathbf{y}}_n^a}{\|\bar{\mathbf{y}}_n^a\|}\right\|^2\right) \cdot\exp\left(-\frac{1}{2\sigma_\omega^2}\sum_{k\in\Omega_n}\|\mathbf{y}_k^\omega\|^2\right) > \gamma$$

Let,

$$T(\mathbf{z}_n) = -\frac{2}{N}\ln L_G(\mathbf{z}_n)$$

We can then state the GLRT as: choose $\mathcal{H}_1$ if

$$T(\mathbf{z}_n) = \frac{1}{N}\sum_{k\in\Omega_n}\left(\frac{1}{\sigma_a^2}\left\|\mathbf{y}_k^a - g\frac{\bar{\mathbf{y}}_n^a}{\|\bar{\mathbf{y}}_n^a\|}\right\|^2 + \frac{1}{\sigma_\omega^2}\|\mathbf{y}_k^\omega\|^2\right) < \gamma'$$

where $\gamma' = -(2/N)\ln(\gamma)$.

The ZUPT Aided INS Program Package

The program package is an adaptation of an open-source program package by OpenShoe Foundation, and all the codes that have been used have their respective copyrights and proper licenses in the respective files.

The main program package

Outputs

7 graphs the details of which are discussed in the upcoming sections

Inputs

The accelerometer and gyroscope data vector

Function that run the zero-velocity aided INS Kalman filter algorithm.

Outputs

@param[out] x_h Matrix with the estimated navigation states. Each row holds the [position, velocity, attitude, (biases, if turned on),(scale factors, if turned on)] for time instant k, k=1,...N.

@param[out] cov Matrix with the diagonal elements of the state covariance matrices.

Inputs

@param[in] u The IMU data vector.

@param[in] zupt Vector with the decisions of the zero-velocity.

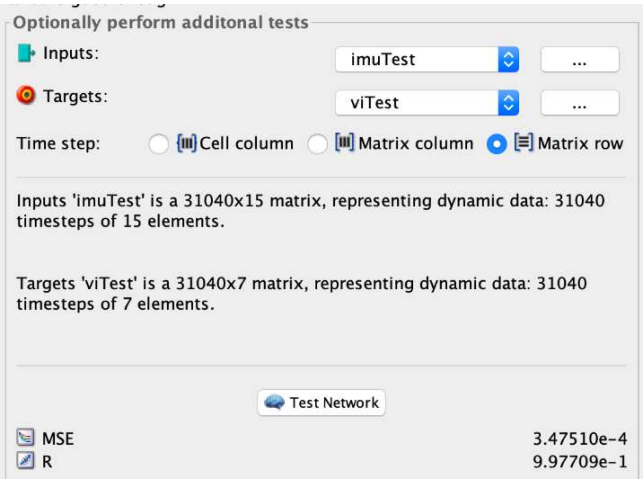
L-IONet

The readings 3-Dimensional accelerations $a_i \epsilon R^3$ and 3-Dimensional angular rates $w_i \epsilon R^3$at

time step i were continuously taken and divided into independent sequences of n frames.

The generated data is fed into Recurrent Neural Networks such as LSTM (in our case) to update the value of h (hidden value) depending on the database set $x_i = (a_i, w_i)$ which we get from our sensors.

$$h_{i+1} = LSTM(h_i, x_i)$$



But this process keeps on storing all the history of each step. Hence, resulting in huge amounts of data storage and high amounts of computations making the device slow.

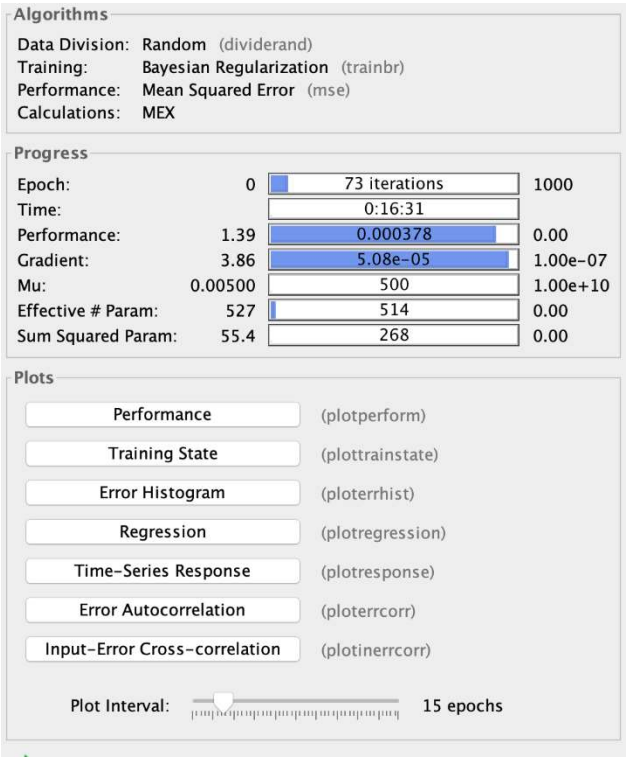
A modification of Lightweight Inertial Odometry Neural Networks (L-IONet) using LSTM models as framework along with Convolutional Neural Networks (CNN) enabled us to dilate the parameters of RNN model resulting in convergence of computational data as the dependence increases on recent state making L-IONet more optimal for usage in low end devices.

$$Z = \tan h(W_{f,k} * x) \odot \sigma(W_{g,k} * x)$$

$$\Delta l, \Delta\psi = FC(h_n)$$

$$L_n^x = L_0^x + \Delta l \cos(\psi_0 + \Delta\psi)$$

$$L_n^y = L_0^y + \Delta l \sin(\psi_0 + \Delta\psi)$$



The code can be explained in brief through following points:

1) The script implemented assumes that the following variables that we have used are defined:
   a. imu1 - input time series.
   b. vi1 - feedback time series.
   c. imu test - test input time series
   d. vi test - test feedback time series

2) Next a Training Function was chosen. There are many training functions out of which few are listed below, which were considered for implementation:

A). The Levenberg-Marquardt ('trainlm' is training function) algorithm is an iterative technique that locates the minimum of a function that is expressed as the sum of squares of nonlinear functions.

B). Bayesian regularized ('trainbr' is training function) is more robust than standard back-

propagation nets and can reduce or eliminate the need for lengthy cross-validation. Bayesian regularization is a mathematical process that converts a nonlinear regression[1] into a "well-posed" statistical problem in the manner of a ridge regression.

C). Scaled conjugate gradient ('trainscg' is training function) method is an algorithm for the numerical solution of particular systems of linear equations, namely those whose matrix is positive-definite.

Out of these, we settled for Bayesian Regularization backpropagation.

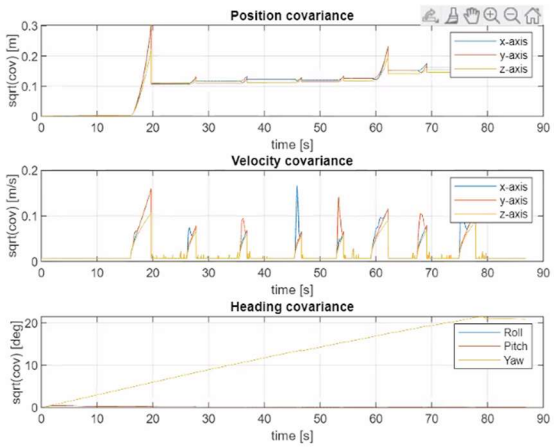
3) Then a Nonlinear Autoregressive Network with External Input was created.
4) After this, the Data was prepared for Training and Simulation. The function PREPARETS prepares timeseries data for a specific network, and shifts time by the minimum amount to fill input states and layer states. By using PREPARETS the original time series data remains unchanged, while easily customizing it for networks with differing numbers of delays, with open loop or closed loop feedback modes.
5) Closed Loop Network - We use this network to do multi-step prediction. The function CLOSELOOP replaces the feedback input with a direct connection from the output layer.
6) Step-Ahead Prediction Network - The original network returns predicted y(t+1) at the same time it is given y(t+1). For some applications such as decision making, it helps to get a prediction timestep early, i.e., predicting y(t+1) once y(t) is available, but before the actual y(t+1) occurs. The network can be made to return its output a timestep early by removing one delay so that its minimal tap delay is now 0 instead of 1. The new network returns the same outputs as the original network, but outputs are shifted left one timestep.
7) MSE, Regression was calculated and graphs of error autocorelation[2], regression and performance (which represents MSE) were plotted for LIONet processed data.

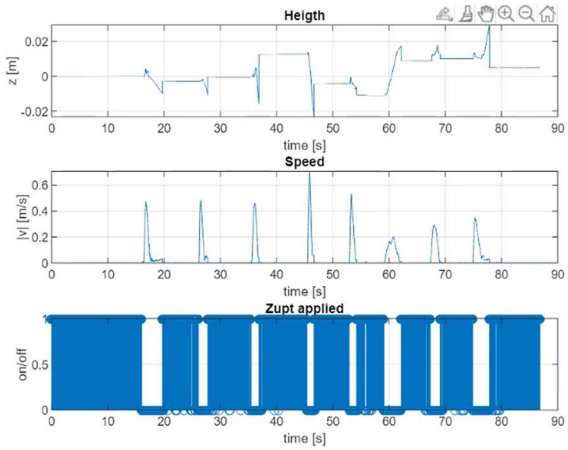
CONCLUSION

4.1 Inferences

All the graphs for ZUPT have a base of 0 rather than a constant value for an ideal case, indicating the ideal case is being subtracted and a relative result thereby generated.

1. [1] Regression: it is the measure of the relation between the mean value of one variable (e.g., output) and corresponding values of other variables (e.g., time and cost)
2. Mean square error (MSE): in statistics, the mean squared error measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. The mean squared error tells you how close a regression line is to a set of points.
3. [2] Error autocorrelation: error terms in a time series transfer from one period to another. In other words, the error for one time period a is correlated with the error for a subsequent time period b.





Statistical Error:

The covariance graphs show significant correlation for all three, x, y and z, thereby indicating that the error generation is not random and due to lower precision but corresponding to the movements made during hand held data collection. Therefore, we infer that precision errors are much lower than device and integration (Markov) induced errors.

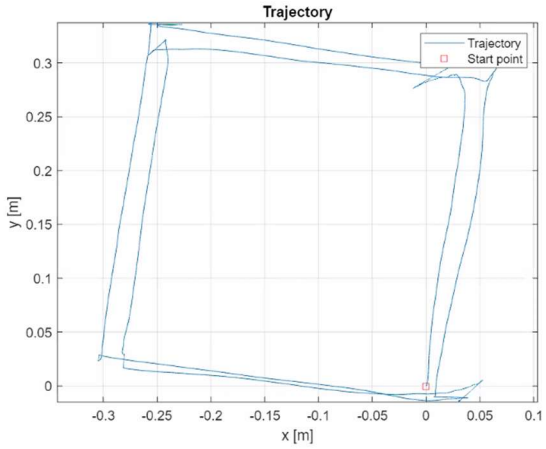
For LIONet MSE values come out to be very low, supporting our hypothesis that the statistical error is very low, as hypothesized.

Systematic Error:

The Gyroscope and Accelerometer bias error graph shows a steep increase in the initial onset of motion and becomes near parallel to the time axis with slight fluctuations about the mean signifying that the Kalman Filter has caught the additive error generated for data values in a standard smartphone IMU, which in most cases is coming out to be near constant and is successfully able to remove it. The large bias error value represents significant systematic error.

Miscellaneous Results:

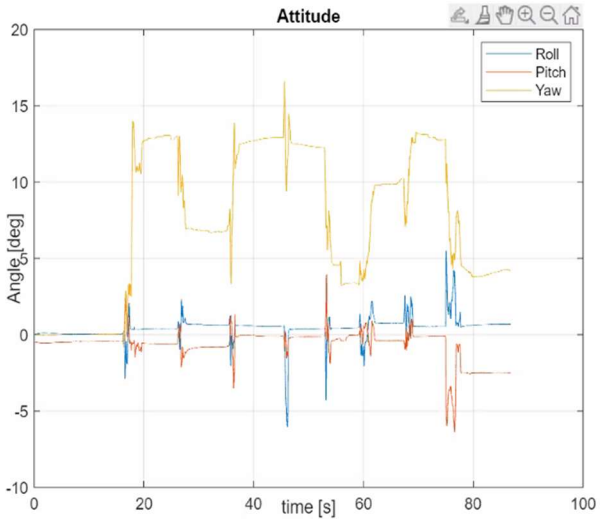
Height plot: For an ideal case, the plot should be parallel to the time axis, but in all cases, graphs obtained have variations which get significant when a person turns or completes a walk cycle, clearly representing an up and down motion of an unsteadily moving phone. The error gets magnified as a result of double integration of accelerometer data values for the z direction.



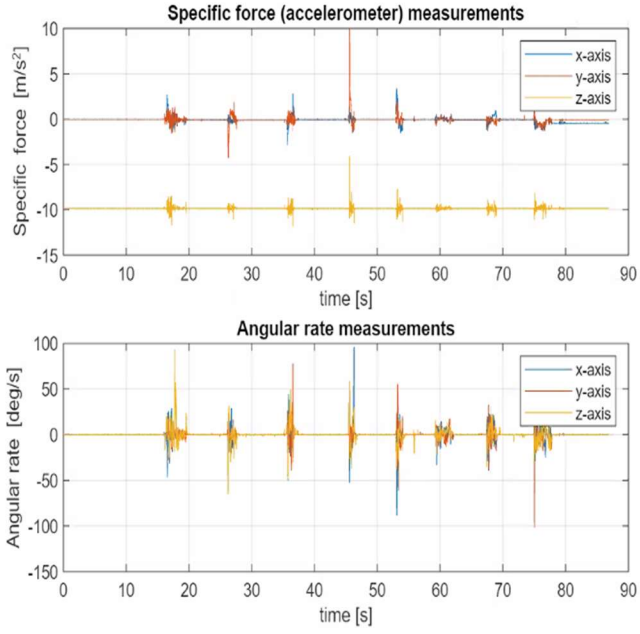
Speed: The speed plot shows a clear spike when a person turns, indicating a significant change in angular velocity.

ZUPT On/Off: The on/off graph for detection is successfully able to detect when a person's velocity magnitude becomes 0 for large and short

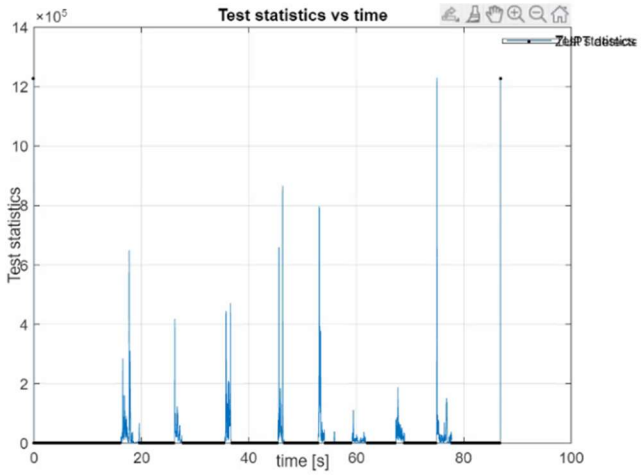
durations, that is, when a person completes a walk cycle or takes a turn.



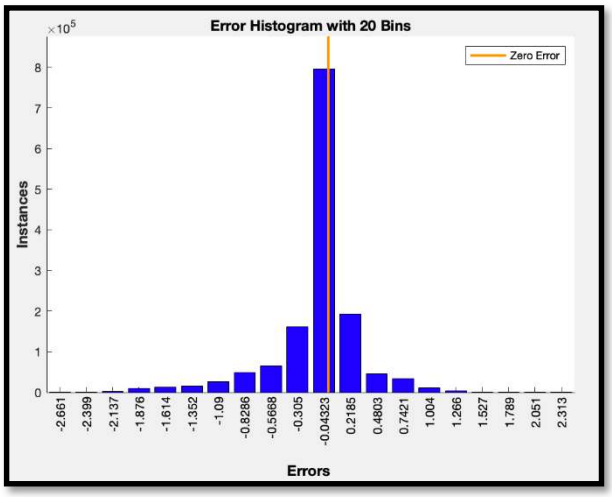
Yaw/Pitch/Roll: The yaw value changes more and unsteadily again indicating a walking movement over a sliding or linear motion.

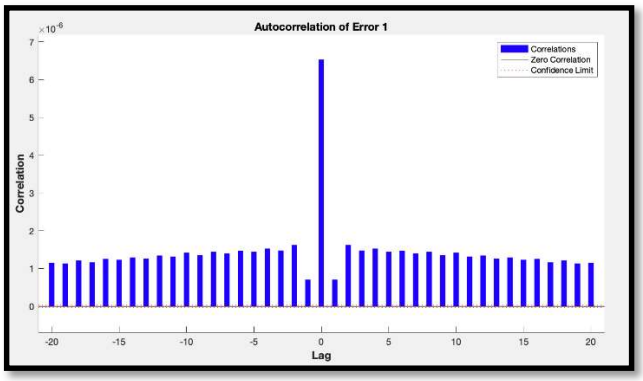


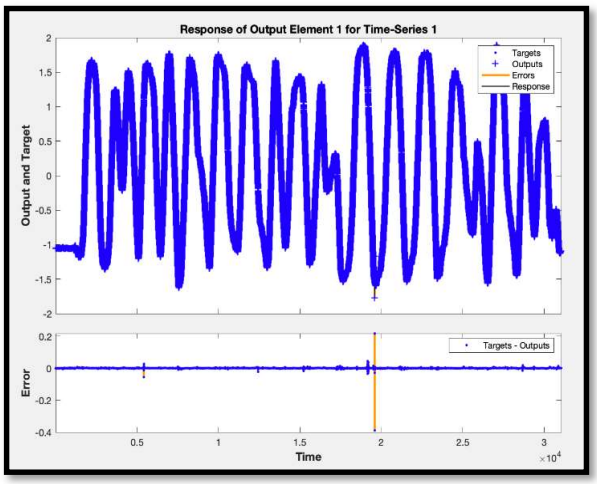
Angular Rate and Specific Force and Test Statistics: The plots again show spikes when a person takes a turn.

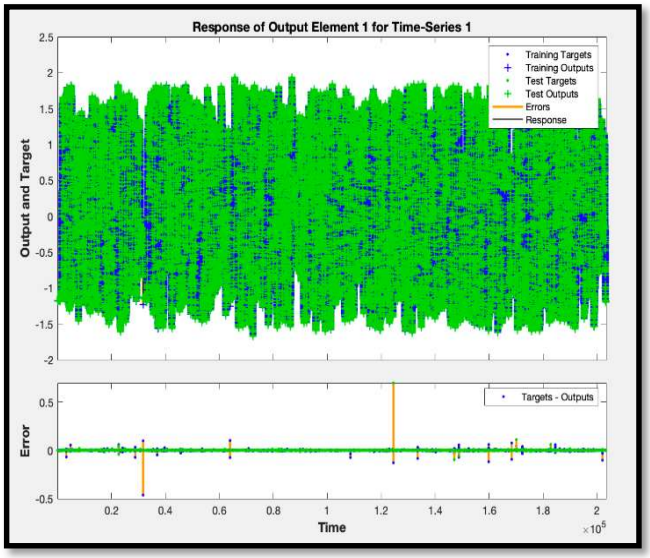


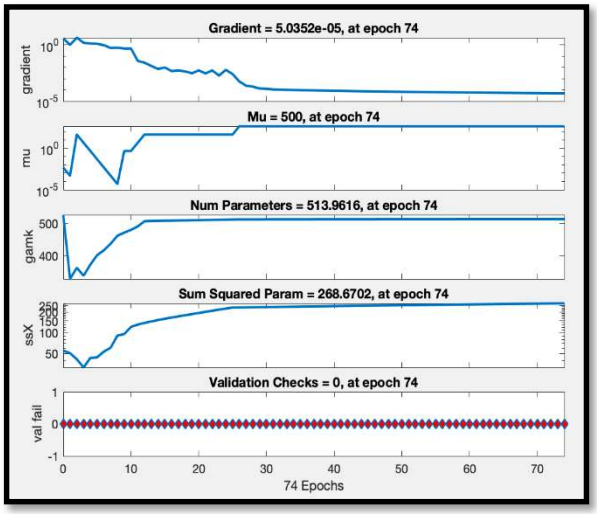
Graphs for L- IONet:

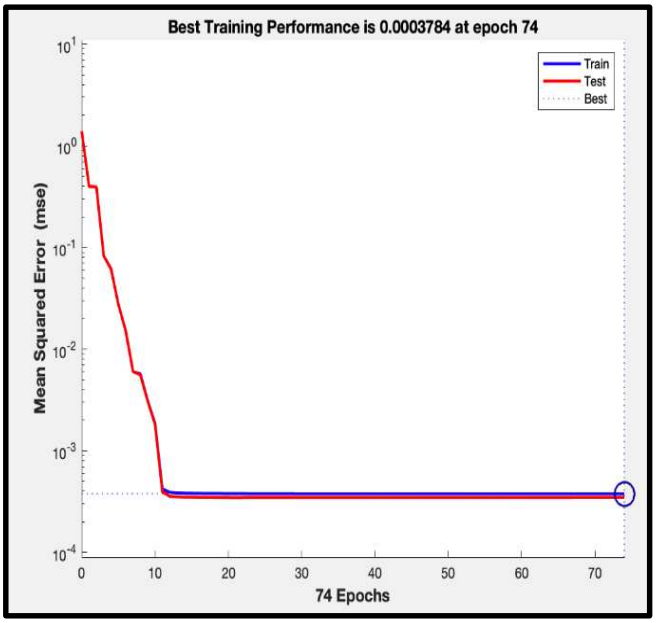


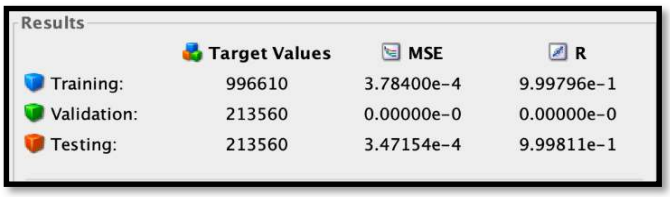












4.2 Future research

In this research work, we propose L-IONet and ZUPT, lightweight profound neural network structure to take in the inertial tracking from crude IMU information. They show serious execution over past profound inertial odometry models.

Future work could incorporate gathering information from additional testing circumstances, for instance, 3D following and tracking. Online normal benchmark and apparatuses for the examination of odometry models could be made. Data from more IoT gadgets for example, smartwatch, wristband, headphones, could be added to our dataset to expand the possible uses of this research.
A further expansion to current profound inertial odometry models is to embrace information refining to pack the deep neural organizations, which can diminish the number of boundaries and empower quicker preparing and on-gadget deduction.

Another future exploration could be to examine how to detail on an expanded easygoing convolutional model (for example, WaveNet style model) as a convenient framework to deal with an assortment of sensor information, for example, temperature and pressure and in other potential IoT application areas, for example, wellbeing/movement checking, sport examination, smart homes, and transportation.

REFERENCES

[1] Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, Niki Trigoni , "Deep Learning based Pedestrian Inertial Navigation: Methods, Dataset and On-Device Inference", IEEE INTERNET OF THINGS JOURNAL, VOL. X, NO. X, X 2019

[2] Yusheng Wang, Sina Askari, and Andrei M. Shkel, "Study on Mounting Position of IMU for Better Accuracy of ZUPT-Aided Pedestrian Inertial Navigation", 2019 IEEE

[3] Yusheng Wang , Andrei Chernyshoff , and Andrei M. Shkel, "Error Analysis of ZUPT-Aided Pedestrian Inertial Navigation", 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 24-27 September 2018, Nantes, France

[4] Zijun Zhou, Shuqin Yang, Zhisen Ni, Weixing Qian, Cuihong Gu and Zekun Cao, "Pedestrian Navigation Method Based on Machine Learning and Gait Feature Assistance", Published: 10 March 2020

[5] Jin Woo Song and Chan Gook Park, "Enhanced Pedestrian Navigation Based on Course Angle Error Estimation Using Cascaded Kalman Filters", Published: 21 April 2018

[6] Changhao Chen, Chris Xiaoxuan Lu, Johan Wahlstrom, Andrew Markham and Niki, **"**Trigoni Deep Neural Network Based Inertial Odometry Using Low-cost Inertial Measurement Units"

[7] Evaluation of zero-velocity detectors for foot mounted inertial navigation systems. (2010,September 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5646936/

[8] Zero-Velocity Detection—An Algorithm Evaluation. (2010, November 1). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/5523938/

[9] Poulose, Alwin & Steven Eyobu, Odongo & Han, Dong. (2019). An Indoor Position-Estimation Algorithm Using Smartphone IMU Sensor Data. IEEE Access. vol. 7. pp. 11165-11177. 10.1109/ACCESS.2019.2891942.