Group 2: Project FindMe - Tracking using IMU Sensor data

Rajib Sahoo Navneet Sharma Atanu Kar
Ashutosh More Abhay Kumar Dwivedi Mayank Devnani Siddhartha Bura
Amit Kumar Singh
22111088, 22111087 , 22111086 , 21111017, 22111001, 22111042, 21111059, 190117

Abstract

*Our project "***FindMe***" focuses on obtaining real time tracking of an individual based on Inertial Measurement Unit(IMU) sensors i.e Accelerometer, Magnetometer and Gyroscope data. In this project, we have obtained sensor data of these sensors (inbuilt in any smartphone) in real time through android apps like PhonePi+, Sensor streamer, HyperIMU,etc available in open source platform. These streamed data samples were used to track the movement of an individual. We have followed two separate approaches in order to get higher accuracy in tracking an individual. This project focuses on tracking an individual purely based on IMU sensor data wherever GPS based tracking is not possible akin to indoor tracking. This particular technique is still evolving and under research phase. Many techniques and approaches were suggested and tested by scholars but tracking with only IMU Sensor with very high accuracy is not achieved yet. Two approaches were taken while performing the experiments as part of our project and the experimental values were correlated with the ground values. The results were comparable with the ground truth, however accuracy needs to be improved further. These approaches and result are discussed in the relevant section in this paper.*

1 Introduction

Localization has two facets i.e. indoor and outdoor localization. Though Google Maps is one of the best options for outdoor navigation, it is highly limiting in indoor environments for reasons like difficulty in penetration through walls of the buildings and the sub meter accuracy sought in indoor navigation. The indoor localization is still a matter of research and various techniques have been proposed. Outdoor navigation with Google maps and other such apps is strictly dependent on internet and is not worthwhile once the data services become unavailable. There is a need to look for other options that are more real time and independent of internet or data services. Towards this direction our team has made use of the IMU sensor data and bring it to fore for more accurate and real time navigation which is independent of data service availability. In this project we have used IMU sensor data from platforms like mobile phones which are easy to carry, to a web page on the mobile phone itself which can be obtained from the browser. There are multiple apps which can help us to extract the IMU data from mobile phones and then with the help of some filters like Kalman filter this data can be approximated to near accurate levels such that we get results with minimum errors.

2 Related Work

- Abdul Rahman Alarifi, Abdul Malik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A. Al-Ammar, and Hend S. Al-Khalifa in their paper titled Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances, have discussed Internal and external factors SWOT analysis for UWB technologies which may affect this technology in indoor localization. UWB provides a high accuracy positioning in addition to many other features (e.g., license free, low power consumption, does not interfere with most of the existing radio systems, high level of multipath resolution, large bandwidth, and high data rate communication), UWB technology may affect GPS and aircraft navigation radio equipment and can also cause interference to the existing systems that operates in the ultra wide spectrum. In comparison to other technologies, UWB systems have emerged as one of the leading technologies for indoor positioning and have been used in many more applications than before. The system is dependent on database generation and careful cooperation among nodes.

- Zhice Yang , Zeyu Wang , Jiansong Zhang , Chenyu Huang and Qian Zhang in their paper titled Wearables Can Afford: Lightweight Indoor Positioning With Visible Light have explained how we can use the polarization of light , BCSK and probabilistic models for localization. The smartphones required in this case need not be high end. However, the constraint is in terms of the high electricity consumption due to densely deployed lamps. Though the accuracy achieved is sub-meter and LED as a light source is not necessary but the initial installation cost is very high.

- Zengshan Tian,Yuan Zhang, Mu Zhou and Yu Liu in their research paper titled Pedestrian dead reckoning for MARG navigation used a Huawei smartphone. The data used is IMU sensor based and has been obtained on smartphone platform. The proposed algorithm can act as a guidance towards Wi-Fi and MEMS based navigation systems. Though the claimed accuracy is high but the variation is in the range of nearly 08 meters which is high in an indoor environment.

3 Approaches adopted

We have worked on two different approaches towards our project.

- **Approach 1:** Use of PhonePi+ app initially and later on HyperIMU app to obtain acclerometer and gyroscope data samples while holding a smartphone along an open ground. The values were then filtered by applying Kalman filter and the resultant values were plotted to obtain real time positioning

- **Approach 2:** The second approach we have worked upon is use of Attitude and Heading Reference System (AHRS) alongwith IMUfusion. Data samples were streamed through the HyperIMU app and real time tracking was plotted on a graph.

4 Methodology

In this section, we provide an elaborate procedure based on how we setup and conducted the experiments.

Approach 1: Tracking based on HyperIMU app

1. *Tools/Approaches used*:

- We started with our experiments on tracking using PhonePi+ app installed in a smartphone. Phonepi+ app is an online available Playstore app which can be downloaded and it can stream data samples for eight sensors (accelerometer, gyroscope,orientation,proximity,etc). We obtained acceleromter and gyroscope values using Phonepi+ app and streamed it to server for plotting. We calculated Rotation Matrix by integrating Gyroscope data and removed the effect of gravity from streamed samples. Again rotation matrix was used to get acceleration data values in Global Coordinate System. However we faced synchronization issues with respect to the streamed data samples as the values from two sensors had time lag issue. To obviate this limitation we used HyperIMU app which allowed us to stream both sensor values simultaneously.

- **HyperIMU :** HyperIMU application [1] is used to get the measurement of acceleration in the Earth coordinate frame from its accelerometer and orientation data. HyperIMU sensor app was created to assist academics and developers in gathering all sensor data from a device for processing and analysis. Numerous sensor types embedded in mobile devices can be used to create an infinite number of applications across numerous industries. To the fullest extent possible, HyperIMU enables them. Some of the applications of HyperIMU app are streaming the geolocation data from all sensors, dependable sampling period, extremely flexible stream packet, TCP, UDP, JSON, CSV, and many other formats and protocols and Pharos Server's remote configuration and control.

- Using HyperIMU was a better option as it streamed data simultaneously and it provides linear acceleration and orientation values eliminating the requirement of removing effect of gravity and calculating rotation matrix using gyroscope data. A series of experiments were then conducted in an indoor as well as outdoor environment to obtain the sensor data. The values streamed were being plotted simultaneously on a laptop. The plotted results and actual trajectory followed was highly inaccurate. We then applied Kalman Filter to remove these inaccuracies, the results thereafter produced were much refined and it nullified the background noise to a considerable extent.

2. *Actual Approach:*
   - Linear acceleration and orientation values are collected in a CSV using the HyperIMU app.
   - The roll-pitch-yaw values obtained from orientation sensor are used to calculate the Rotation matrix given by the formula:

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}$$

   - The rotation matrix is then used to obtain acceleration in the global coordinate system.

$$a_{global} = \mathbf{R^T}\, a_{sensor}$$

   - Kalman filter is applied on the global acceleration values using *pykalman* library in Python.
   - Then the filtered global acceleration values are double integrated to get the position.

$$(x, y, z) = \iint_0^t a_{global}\, dt$$

   - The (x,y) results are plotted in real-time using *FuncAnimation* module from *matplotlib.animation*.
   - In the end, the plot shows an 'Outside' message if the position goes beyond a circular boundary.

3. *Accuracy:* Due to the double integration in calculation, the error in accelerometer and orientation values gets accumulated over time. Using Kalman filter, we were able to reduce significant errors. So the method works well till 20-25 seconds, after which the error becomes large that it surpasses the actual values and a change in actual values has little effect on the growing calculated value.

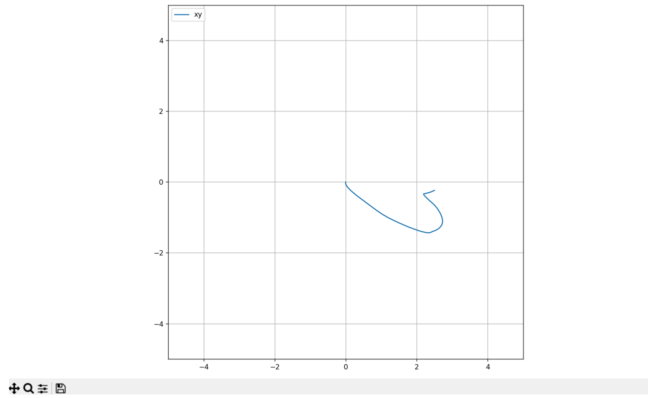

Figure 1: Screenshot of real-time tracking

Approach 2: Tracking based on AHRS and HyperIMU app

1. *Tools/Approaches used*:

- **AHRS Algorithm :** Sensors on three axes constitute an attitude and heading reference system (AHRS), which delivers roll, pitch, and yaw information for aircraft. That included solid-state or microelectromechanical systems (MEMS) gyroscopes, accelerometers, and magnetometers and are often referred to as MARG (Magnetic, Angular Rate, and Gravity) sensors. The older style gyroscopic navigation instruments are to be substituted by them. The on-board processing system of an AHRS, which provides attitude and heading information, is the fundamental distinction between an AHRS and an Inertial measuring unit (IMU). In contrast, an IMU feeds sensor data to a standalone module that analyzes attitude and heading. With sensor fusion, reference vectors such like gravity and the Earth's magnetic field are utilized to compensate drift from the gyroscopes' integration.An AHRS is a more cost-effective alternative to the conventional high-grade IMUs that merely include gyroscopes and rely on the gyroscopes' high bias stability since this generates a drift-free orientation. An AHRS may serve as a subsystem of an inertial navigation system in assisting determine attitude.

- **Kalman filter :** The majority of contemporary systems feature a large number of sensors that, using a chain of measurements, estimate hidden states. A GPS receiver, for example, provides location and velocity estimation, where location and velocity are the hidden states and measurements are the differential timing of the signals receiving.
- The provision of an accurate and precise estimation of the hidden states in the face of uncertainty is one of the primary challenges confronting tracking and control systems. The measurement uncertainty of GPS receivers is affected by a variety of external factors, including thermal noise, atmospheric effects, minute variations in satellite positions, receiver clock accuracy, and many more.One of the most essential and useful prediction algorithms is the Kalman Filter. The Kalman Filter derives hidden variable estimates based on unreliable and erroneous measurements. The Kalman Filter additionally assesses system state based on projections from the past.Target tracking (Radar), location and navigation systems, control systems, computer graphics, and numerous other applications employ the Kalman filter today.

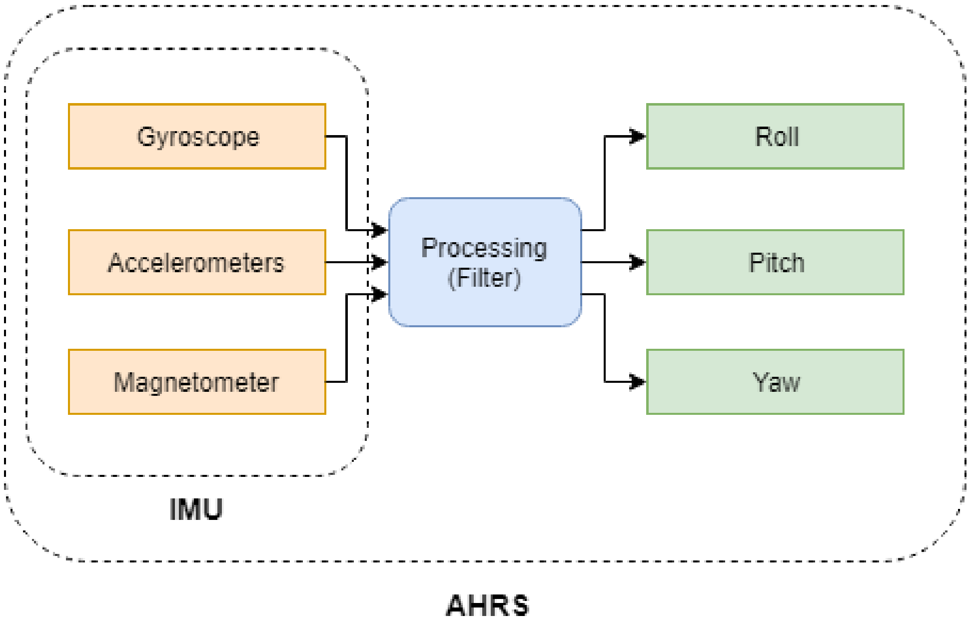

Figure 2: Attitude and Heading Reference System(AHRS)

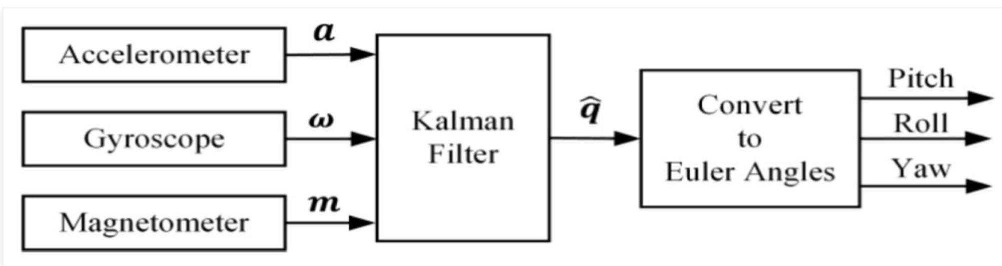

Figure 3. Block diagram of the first approach to Kalman filter design

Figure 3: Block diagram of Kalman filter design

2. *Actual Approach:*
   - The programme leverages Fusion to transform gyroscope and accelerometer data into a measurement of acceleration in the Earth coordinate frame. Fusion is a sensor fusion library for Inertial Measurement Units (IMUs), optimised for embedded systems.
   - The measurement of velocity is then calculated by integrating the acceleration data.
   - Using a zero-velocity detection technique, the measurement of velocity is drift-corrected before being integrated to provide the measurement of position.
   - Gyroscope, Accelerometer and Magnetometer values are fed into Processing filter to detect the Roll, Pitch and Yaw movements.
   - The roll axis, also referred as the longitudinal axis, commences at the gravity and is aimed forward onto the reference line of the fuselage. The term "bank" refers to an angular displacement about this axis.
   - The pitch axis, commonly known as the transverse or lateral axis, arises at the gravity and runs parallel to a line drawn from wingtip to wingtip to the right. We refer to motion about

this axis as pitch.

- The yaw axis arises at the gravity and is pointed downward, perpendicular to the wing reference line to the bottom. Yaw is the name for motion about this axis.

- The position of the co-ordinates X,Y are then rendered on a webpage plotting the path taken.

3. *Accuracy:* Since the IMU sensor values are prone to noise, calibration is needed to get the filtered values. Kalman filter is applied to reduce the noise inherited. Most of the noise is eliminated, but not completely.

5 Results

5.1 Approach 1:

We went on performing approach 1 first. We had selected PhonePi+ android application available on open source platform, in order to collect IMU Sensor data from the smartphone. As we have observed that using PhonePi+ it is very difficult to get multiple sensor values with same timestamp so we migrated to HyperIMU application available on google playstore. Firstly we have calculated rotion matrix to map local coordinate onto the global coordinate system and then using Kalman Filter we had eliminated the noise factor and the calculated the position by double integrating the acceleration data. The result obtained by this were tested on ground and an appreciable result was obtained with a lag of 3-4 sec while tracking an individual with a smartphone. The result obtained by our test can be seen in the below figure.

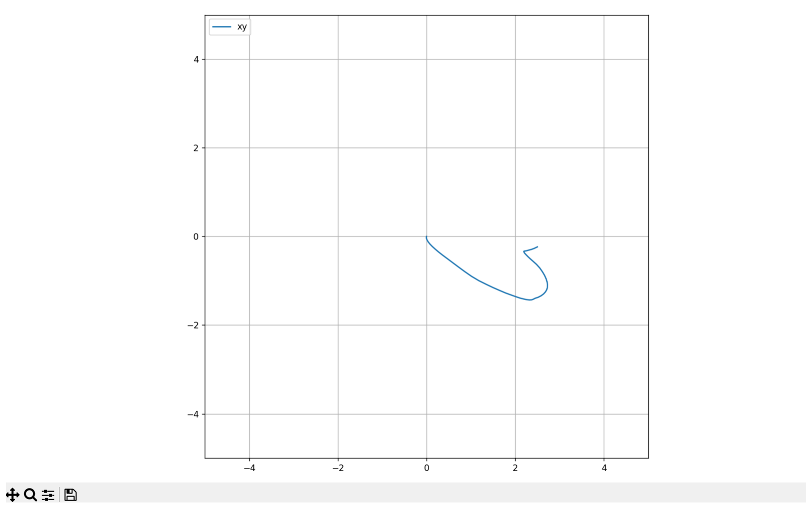

Figure 4: Results of approach 1

5.2 Approach 2:

In order improve the accuracy furthermore, we had adopted more research on the technique of tracking based on IMU sensor data. We homed on to a similar research work based on AHRS algorithm [2] for tracking using IMU sensor data. The approach was discussed in Approach 2. Results achieved by implementing this was comparable with Approach 1 and higher accuracy was observed. The experiment was carried out in an open ground. It was observed that the "L" shaped route actually followed by the individual holding the smartphone and result obtained by the our method was highly accurate as can be seen in below figure.

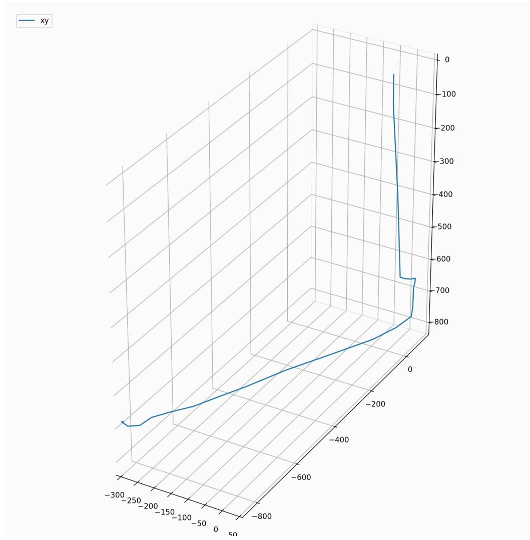

Figure 5: Tracking of an individual on L shaped route using Approach 2

To test the approaches, we performed experiments where we walked along variety of paths like circular, L-shaped, U-turn etc. The output closely resembled the travelled path on an L shaped path shown below:

Figure 6: Data Collection

In order to access the tracking in real time over a LAN via various devices we had configured an web server on XAMPP (Open source platform). A basic Web page was designed which renders after every seconds to update the data from the server. Thus we had achieved the tracking of an indidual with a smartphone remotely from an Anchor station.

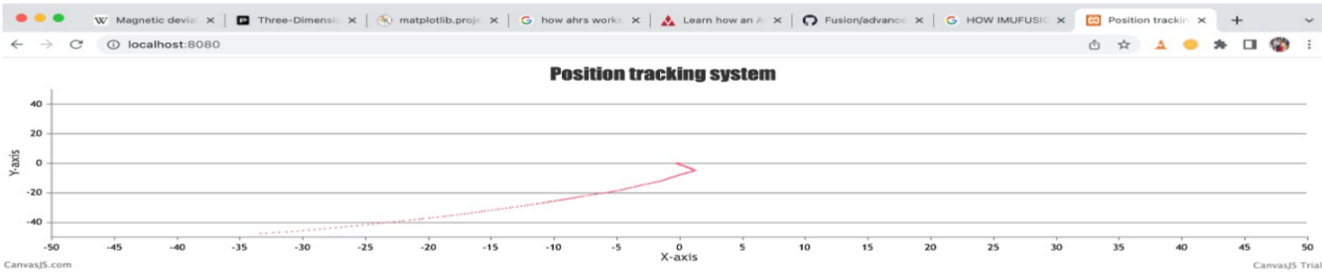

Figure 7: Webpage plotting

6 Discussion and Future Work

IMU Sensor data based tracking is highly prone to errors despite applying Kalman filter. In our project we have observed that the results obtained are satisfactory. However, tracking accuracy can further be improved by implementing it with GPS data for outdoor positioning and for indoor positioning we can resort to the method mentioned in this work[3].

We may overcome such challenges in indoor localization with the help of algorithms like simultaneous localization and mapping[4]. They are LIDAR based [5] and are in vogue for disciplines like Robotics.

7 Conclusion

IMU based apps alongwith application of certain fiters, in this case we used Kalman filter, were able to track path of an individual with adequate accuracy. Indoor tracking is more error prone as compared to outdoor tracking. Also the real time tracking yielded good results, however there is a requirement of efficient algorithm/filter to help achieve continuous tracking or tracking for a considerable period of time. The entire project source code is available on link [6]for references.

With the available technologies, instead of smartphones, smart sensor based tracking is likely to yield more efficient results along with exploitation of efficient algorithms.

References

[1] https://github.com/ianovir/HIMUServer. Hyper imu server configuration.

[2] Dariusz Tomaszewski, Jacek Rapiński, and Renata Pelc-Mieczkowska. Concept of ahrs algorithm designed for platform independent imu attitude alignment. *Reports on Geodesy and Geoinformatics*, 104, 2017.

[3] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A Al-Ammar, and Hend S Al-Khalifa. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(5):707, 2016.

[4] Zhice Yang, Zeyu Wang, Jiansong Zhang, Chenyu Huang, and Qian Zhang. Wearables can afford: Light-weight indoor positioning with visible light. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 317–330, 2015.

[5] Zengshan Tian, Yuan Zhang, Mu Zhou, and Yu Liu. Pedestrian dead reckoning for marg navigation using a smartphone. *EURASIP Journal on Advances in Signal Processing*, 2014(1):1–9, 2014.

[6] https://github.com/Siddhartha Bura/FindMe. Source code of findme project.