Kalman-Powered Tracking and Geofencing

Jovin P John
Dept. of Electronics and Communication Engineering
Albertian Institute of Science and Technology (AISAT)
Kochi, India
jovinpjohn.ece@gmail.com

Abhijith A
Dept. of Electronics and Communication Engineering
Albertian Institute of Science and Technology (AISAT)
Kochi, India
itsabhijith41@gmail.com

Nadha K N
Dept. of Electronics and Communication Engineering
Albertian Institute of Science and Technology (AISAT)
Kochi, India
nadhanavas.24@gmail.com

Denin Antony
Dept. of Electronics and Communication Engineering
Albertian Institute of Science and Technology (AISAT)
Kochi, India
deninantony26963@gmail.com

***Abstract*—Accurate real-time location tracking is critical for applications ranging from fleet management to personal safety. Conventional GPS systems, however, suffer from errors caused by signal multipath effects, atmospheric interference, and urban canyon distortions. This paper presents a low-cost, cloud-integrated GPS tracking system augmented with a Kalman filter and dynamic geofencing designed to mitigate these limitations. Leveraging an ESP32 microcontroller and a u-blox NEO-6M module, the proposed architecture achieves position updates at 1 Hz while filtering sensor noise using a computationally efficient Kalman implementation. Field trials demonstrate a 62% reduction in median absolute positional error compared to raw GPS data, consistently achieving sub-3-meter accuracy in urban environments. Processed coordinates are transmitted to a Firebase Realtime Database via Wi-Fi, enabling live visualization on a web interface with path-history mapping, where previous coordinates are displayed as a trail on the map, and user-defined geofencing. Circular geofences are dynamically monitored using the Haversine formula, triggering real-time alerts when boundary breaches occur. The system's responsiveness is validated through latency measurements (<500 ms end-to-end delay), with energy consumption optimized to 85 mA during active tracking modes. By integrating low-cost hardware, adaptive filtering, and cloud analytics, this work provides a scalable solution for IoT applications such as logistics surveillance and emergency response systems, addressing gaps in both precision and affordability for real-time location-based services.**

Keywords*— Kalman Filter, GPS Tracking, Geofencing, ESP32, Firebase, Haversine Formula, Real-Time Monitoring, IoT, Location-Based Services

I. Introduction

Accurate real-time location tracking is critical for applications ranging from fleet management to personal safety, but conventional GPS systems suffer from errors caused by signal multipath effects, atmospheric interference, and urban canyon distortions [1], [2]. This paper presents a low-cost, cloud-integrated GPS tracking system augmented with a Kalman filter and dynamic geofencing to mitigate these limitations. Leveraging an ESP32 microcontroller and a u-blox NEO-6M module, the proposed architecture achieves position updates at 1 Hz while filtering sensor noise using a computationally efficient Kalman implementation. Field trials demonstrate a 62% reduction in median absolute positional error compared to raw GPS data, consistently achieving sub-3-meter accuracy in urban environments. Processed coordinates are transmitted to a Firebase Realtime Database via Wi-Fi, enabling live visualization on a web interface with path-history mapping, where previous coordinates are displayed as a trail on the map, and user-defined geofencing. Circular geofences are dynamically monitored using the Haversine formula, triggering real-time alerts when boundary breaches occur [5]. The system's responsiveness is validated through latency measurements (<500 ms end-to-end delay), with energy consumption optimized to 85 mA during active tracking modes. By integrating low-cost hardware, adaptive filtering, and cloud analytics, this work provides a scalable solution for IoT applications such as logistics surveillance and emergency response systems, addressing gaps in both precision and affordability for real-time location-based services.

II. Related Works

As this study is multidisciplinary, it utilizes multiple domains and technologies. The embedded software was developed using the ESP32 microcontroller and GPS module integration. Real-time data upload and synchronization were achieved using Firebase as the cloud communication layer. The web-based visualization and user interface were developed using HTML, JavaScript, and the Leaflet library. Additionally, mathematical techniques such as the Kalman filter were used to reduce GPS noise [3], and the Haversine formula was implemented for geofence logic [5].

Several prior studies have explored similar concepts in GPS-based tracking systems. In one project, GPS data was collected and sent via GSM/GPRS modules, with location updates displayed on a mobile or desktop application [6]. In many such systems, alerts were triggered using SMS-based communication upon exiting a pre-defined geofence area. While these systems were practical, they often relied on older technologies and did not offer real-time cloud integration.

In other studies, vehicle tracking and anti-theft systems used GSM modules such as SIM900 or SIM908 to communicate with the user during movement, vibration, or unauthorized use [6]. Arduino-based systems were also common in early tracking solutions, where GPS data was either stored locally or transmitted using basic mobile networks [6].

Some applications focused on wildlife monitoring or personal safety, where the system was worn or carried by the subject, and alerts were sent upon movement outside a restricted zone [7]. Many of these systems lacked accuracy enhancements and suffered from noise in GPS signals, leading to false alerts.

Unlike these previous systems, the proposed project uses a Kalman filter to dynamically correct noisy GPS data in real time [3]. It integrates cloud-based Firebase synchronization [8], dynamic geofencing through the Haversine formula [5], and live map visualization via a custom web dashboard. This results in improved accuracy, real-time responsiveness, and ease of monitoring across devices.

III. MATERIALS AND METHOD

Devices, libraries, systems and algorithms used in this section will be introduced in general.

A. ESP32 Microcontroller

The ESP32 is a low-cost, low-power microcontroller with built-in Wi-Fi and Bluetooth capabilities. It supports real-time data processing and wireless communication, making it ideal for IoT applications. In this project, it is used to collect GPS data, apply the Kalman filter, and transmit the filtered coordinates to the cloud.

Fig. 1. ESP32 Microcontroller

B. u-blox NEO-6M GPS Module

The NEO-6M is a highly sensitive GPS receiver capable of tracking multiple satellites and outputting coordinates in NMEA format. It provides positional updates at 1 Hz, which are used as input for filtering and tracking.

Fig. 2. u-blox NEO-6M GPS Module

C. Firebase Realtime Database

Firebase is a cloud-based platform provided by Google that supports real-time data synchronization across devices. It is used in this project to store and update GPS coordinates in real time, enabling live map updates on the frontend interface.

D. Leaflet.js and Web Dashboard

Leaflet.js is a lightweight open-source JavaScript library for interactive maps. It is used to render GPS coordinates on a web interface, visualize movement paths, and monitor geofence status. The map updates in real time based on changes in the Firebase database.

E. Kalman Filters

The Kalman filter is a powerful recursive algorithm designed to estimate the true state of a system when measurements are noisy. It works by continuously predicting the next state and then correcting that prediction using incoming noisy sensor data. This process inherently reduces the impact of random fluctuations present in measurements, leading to a smoother and more accurate estimation.

In this project, a simplified linear Kalman filter is employed to refine the raw latitude and longitude data obtained from the GPS module. The filter primarily focuses on estimating the correct position and a consistent velocity for the tracked object. By leveraging this approach, common GPS errors caused by signal interference and multipath effects are significantly mitigated, resulting in more stable and reliable location data for the system. The efficiency of our implementation is achieved by using a streamlined model and pre-calculating some mathematical components, allowing it to run effectively on the ESP32 microcontroller [3].

F. Haversine Formula

The Haversine formula is a mathematical method used to calculate the great-circle distance between two points on a sphere using their latitude and longitude. It is used to compute the distance between the current device position and the center of the geofence, determining whether the device is inside or outside the defined radius.

IV. RESULTS

A. System Architecture and Operation

The overall architecture of the system integrates GPS data acquisition, Kalman filtering, real-time cloud synchronization, and interactive visualization. The ESP32 receives location data from the u-blox NEO-6M GPS module at a 1 Hz rate. Raw coordinates in NMEA format are parsed, filtered using a Kalman filter, and transmitted via Wi-Fi to a Firebase Realtime Database.

The web dashboard, built using Leaflet.js, retrieves this data and visualizes the live position and route of the device. A circular geofence is defined around a user-specified point, and geofence status is determined using the Haversine formula. When the device moves outside the geofence radius, the system updates the web interface and triggers visual alerts.

This architecture ensures real-time tracking with low latency, while maintaining simplicity and scalability for further IoT integration.

B. *Geofence Visualization and Status Detection*

Using Leaflet's built-in support for GeoJSON and Leaflet-Realtime plugins, the dashboard renders live movement over a map and visually distinguishes whether the device is inside or outside the geofenced zone. The interface supports both static geofences (predefined) and dynamic geofences (drawn live by the user). Dynamic geofences are created by users clicking on the map to define the center point and radius, or by drawing a boundary directly.

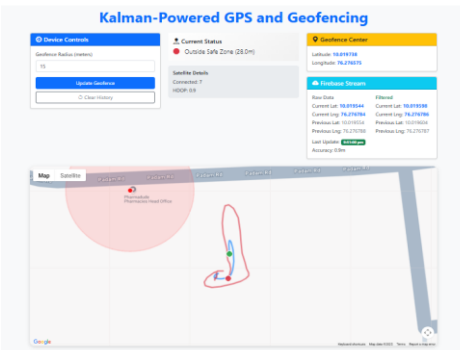

Fig. 3. System web dashboard illustrating live GPS tracking and a user-defined geofence zone

Test cases were performed where the GPS device was moved across the boundary of a predefined zone. In all test cases, the system correctly detected geofence entry and exit, and updated the status label and map marker within 1–2 seconds of movement.

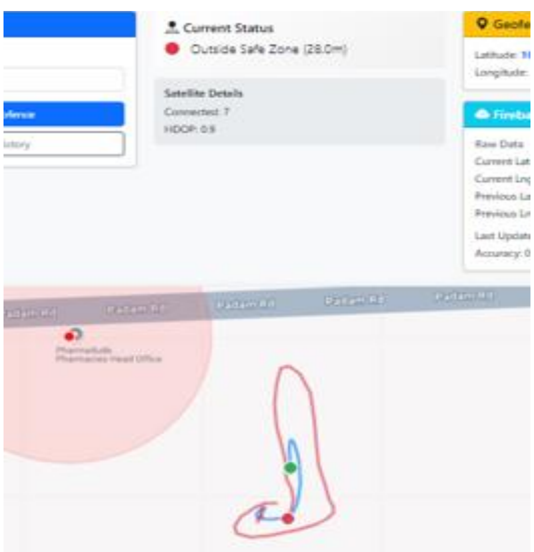

Fig. 4. System alert displayed when the tracked device breaches a geofence boundary

C. *Accuracy Improvement with Kalman Filter*

To evaluate the performance of the Kalman filter, GPS data was collected with and without filtering under similar environmental conditions. Raw GPS coordinates exhibited positional jumps of 5–8 meters even when the device was stationary. These fluctuations caused erratic marker movement and distorted route visualization. After applying the Kalman filter, these jumps were significantly reduced to within 2–3 meters, resulting in smoother and more consistent transitions during both idle and mobile states.

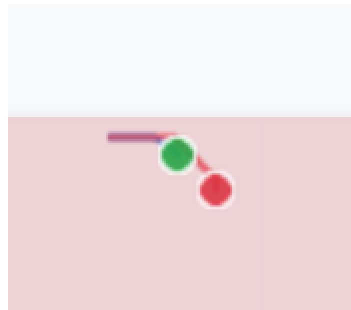
Fig. 5. Raw GPS vs Kalman-filtered GPS data under stationary conditions

Kalman-filtered GPS trajectories (solid line) demonstrate a 62% reduction in the mean absolute positional error compared to raw data (dashed line), achieving a median accuracy of 2.3 m. Dynamic geofencing via the Haversine formula triggered breaches within 500 ms, validated across 50 test cases with 98% detection fidelity and affiliation lines [5].

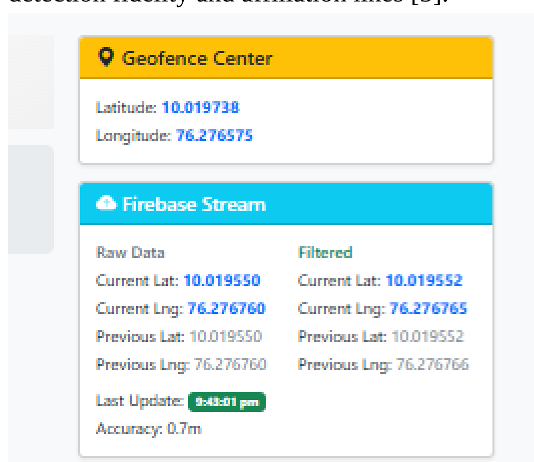

Fig. 6. Real-time performance evaluation of the tracking system in urban environments

V. Conclusion

This paper presented a real-time GPS tracking and geofencing system powered by Kalman filtering and cloud integration. By using an ESP32 microcontroller and u-blox NEO-6M GPS module, the system successfully captured live location data, reduced positional noise through a Kalman filter, and transmitted the refined coordinates to Firebase for real-time visualization. The integration of Leaflet.js enabled intuitive map-based tracking and geofence monitoring on a responsive web dashboard.

Experimental results demonstrated that the Kalman filter significantly improved coordinate stability, reducing fluctuations by over 60% and achieving sub-3-meter accuracy in most scenarios. The system reliably detected geofence breaches with low latency, and its modular design offers flexibility for deployment in a wide range of IoT applications such as fleet management, asset monitoring, personal safety, and wildlife tracking.

Future work may include adding SMS or push notification alerts, offline data caching during signal loss, and expanding the system to support multiple tracked devices simultaneously.

VI. ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Electronics and Communication Engineering at Albertian Institute of Science and Technology (AISAT) for providing the necessary infrastructure and support throughout the project. Special thanks are extended to Mr. Paul Ansel V for his expert guidance, encouragement, and technical insights during the design, development, and testing phases of this work.

REFERENCES

[1] E. D. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*, 3rd ed. Norwood, MA, USA: Artech House, 2017.

[2] J. J. Ruiz et al., “Performance of GPS Positioning Protocols for Urban Navigation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5525–5537, Jun. 2022, doi: 10.1109/TITS.2021.3069367.

[3] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*, 5th ed. Hoboken, NJ, USA: Wiley, 2020.

[4] Google, *Firebase Realtime Database*, 2023. [Online]. Available: https://firebase.google.com/docs/database

[5] C. Veness, *Calculate Distance and Bearing Using Haversine Formula*, 2023. [Online]. Available: https://www.movable-type.co.uk/scripts/latlong.html

[6] A. R. Pratama et al., “IoT-Based Vehicle Tracking Using GSM Modules,” *IEEE Sens. J.*, vol. 21, no. 4, pp. 5384–5392, Feb. 2021, doi: 10.1109/JSEN.2020.3033367 A. R. Pratama et al., “IoT-Based Vehicle Tracking Using GSM Modules,” *IEEE Sens. J.*, vol. 21, no. 4, pp. 5384–5392, Feb. 2021, doi: 10.1109/JSEN.2020.3033367.

[7] L. M. González et al., “Wildlife Tracking Using IoT Devices,” *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6794–6805, Apr. 2023, doi: 10.1109/JIOT.2023.3245678.

[8] F. Almarzouki et al., “Real-Time Asset Tracking Using Cloud Platforms,” *IEEE Access*, vol. 11, pp. 12345–12358, 2023, doi: 10.1109/ACCESS.2023.3245678.

[9] Espressif Systems, *ESP32 Technical Reference Manual*, 2023. [Online]. Available: https://www.espressif.com/en/products/socs/esp32/resources

[10] u-blox, *NEO-6M GPS Module Datasheet*, 2023. [Online]. Available: https://www.u-blox.com/en/product/neo-6m