Real-Time Precision Navigation Design for Autonomous Vehicle with EGI System

Chin-Chia Hsu [1], Yen-En Huang [2], Kai-Wei Chiang [3], Meng-Lun Tsai [4], Chien-Hsun Chu [5]

[1] Dept. of Geomatics, National Cheng Kung University, Tainan, Taiwan - chinchia120@gmail.com
[2] Dept. of Geomatics, National Cheng Kung University, Tainan, Taiwan, - jimmy4345@gmail.com
[3] Dept. of Geomatics, National Cheng Kung University, Tainan, Taiwan - kwchiang@geomatics.ncku.edu.tw
[4] Dept. of Geomatics, National Cheng Kung University, Tainan, Taiwan - taurusbryant@geomatics.ncku.edu.tw
[5] Dept. of Geomatics, National Cheng Kung University, Tainan, Taiwan - 11303040@gs.ncku.edu.tw

Keywords: EGI Inertial Navigation System, HD Maps, LiDAR, Autoware, Autonomous Vehicle.

Abstract

This research presents an innovative system framework, and a multi-sensor integration algorithm aimed at improving the navigation accuracy of autonomous vehicle (AV). This study addresses the shortcomings of the LiDAR-centric approach used in Autoware, a popular open-source platform for self-driving cars. This article presents FalcoNav.AI, a novel INS-centric method, integrating inertial navigation, satellite positioning, Light Detection and Ranging (LiDAR), and High-Definition (HD) maps to enhance navigational performance. An Extended Kalman Filter (EKF) is employed for efficient data fusion and processing of Inertial Navigation System (INS). The performance of system is evaluated in various environments, including open sky, Global Navigation Satellite System (GNSS) challenge and GNSS denied areas, showing significant improvements in navigation accuracy and reliability. The key components of the system include affordable Velodyne VLP-16 and a custom-built Embedded GNSS/INS (EGI) Inertial Navigation System (EGI-370). Experiments demonstrate that the system achieves "Where-in-Lane" level accuracy, highlighting its potential for wide application in autonomous vehicle. This innovation represents a significant advance towards more dependable and precise navigation in a wide range of driving conditions.

1. Introduction

1.1 Overview of Autonomous Vehicles

The global technological landscape has undergone a profound transformation with the emergence of Autonomous Vehicle (AV) technology, marking a paradigm shift in human-transportation interaction. This transformative evolution began in earnest during the 1980s and 1990s, laying the foundation for one of the most significant advancements in modern technology. The Society of Automotive Engineers (SAE) International's standards classify autonomous driving into six levels, ranging from no automation (Level 0) to full automation (Level 5), as illustrated in Figure 1. Level 1 involves basic driver assistance features, such as adaptive cruise control, while Level 3 allows for conditional automation, where the vehicle can handle most of the driving tasks but requires human intervention in complex scenarios. Level 5 represents full automation, where no human intervention is required under any conditions. This classification underscores the escalating complexity and sophistication required in AV design and development to achieve higher levels of automation.

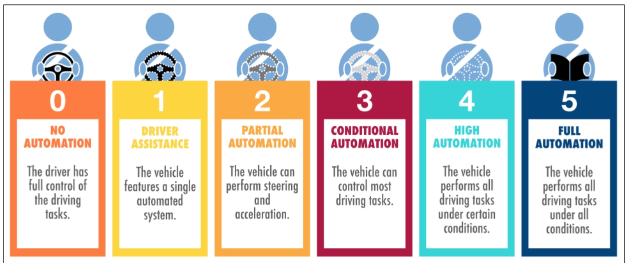

Figure 1. Level of autonomous driving.

As illustrated in Figure 2, navigational accuracy in AV can be categorized into four distinct levels: "which road level" (accuracy above 5 meters), "which lane level" (1.5 meters), "where in lane level" (0.5 meters), and "active control level" (0.1 meters). These levels of accuracy are essential for enabling AV to operate safely in complex environments, such as urban areas with dense traffic or GNSS-denied zones like tunnels and underground parking lots. As AV technology progresses toward and beyond Level 3 automation, the demand for robust positioning systems with centimetre-level accuracy becomes increasingly critical. Such precision is vital for ensuring reliable vehicle operation, particularly in scenarios requiring hazard avoidance or lane-keeping in dynamic environments.

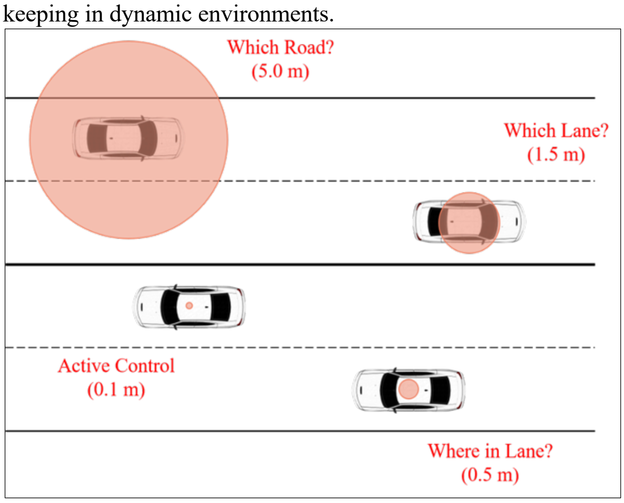

Figure 2. Navigation accuracy classification.

In recent years, the demand for advanced navigation technology has grown significantly across various industries. The Global Navigation Satellite System (GNSS) has become a cornerstone of modern navigation, satisfying the needs of most users. However, GNSS relies on signal transmission and reception, which can be easily disrupted by obstacles such as buildings, trees, and tunnels. In urban environments, GNSS signals are often affected by multipath effects, where signals reflect off surfaces, leading to inaccuracies in positioning. Additionally, atmospheric conditions, such as ionospheric delays, can further degrade GNSS performance. These limitations mean that GNSS-based navigation systems may experience interruptions in challenging environments, particularly in GNSS-denied areas like tunnels or urban canyons, where satellite signals are completely blocked. To address these challenges, the Inertial Navigation System (INS) is often integrated with GNSS, providing continuous and reliable navigation in environments where GNSS signals are unavailable or degraded.

The Inertial Measurement Unit (IMU), a core component of INS, is typically classified into three performance categories: navigation-grade, tactical-grade, and consumer-grade. Navigation-grade IMU, often used in aerospace and military applications, offering the highest accuracy but comes at a significant cost. Tactical-grade IMU strikes a balance between performance and affordability, making it suitable for commercial applications like autonomous vehicles. Consumer-grade IMU, while cost-effective, is limited by higher error rates and is typically used in low-precision applications such as consumer electronics. Lower-performance IMU, such as consumer-grade MEMS (Micro-Electro-Mechanical Systems) gyroscopes, is more prone to accumulating errors, leading to rapid drifts in position and orientation. For example, consumer-grade MEMS gyroscopes often exhibit a bias instability of approximately 70 degrees per hour. In contrast, navigation-grade IMU offers significantly higher accuracy but is often prohibitively expensive for widespread use.

The integration of GNSS and IMU begins with the hardware setup, that precise time synchronization between the GNSS receiver and IMU samples is a critical challenge. Precise time alignment is essential because even minor timing discrepancies can lead to significant errors in the estimation of position and velocity, compromising the overall accuracy of the navigation system. Additionally, correcting bias and scale factor errors in IMU data through rigorous calibration is crucial for improving overall navigation performance. In this study, precise parameter estimation in error models is emphasized, with the Allan variance test serving as an efficient tool for estimating INS error model parameters. The Extended Kalman Filter (EKF) is widely used to merge GNSS and IMU data, and this study adopts a loosely coupled (LC) framework to enhance the robustness and accuracy of the integrated navigation solution. The EKF effectively combines the high-frequency, short-term accuracy of IMU data with the long-term stability of GNSS measurements, resulting in a robust and reliable navigation system.

1.2 Autonomous Vehicle Platform

AV systems are designed to operate without human intervention, utilizing an array of sensors, cameras, LiDAR, RADAR, and Artificial Intelligence (AI) to perceive their environment and make decisions. These systems manage vast amounts of data to navigate roads, recognize traffic signals, and adapt to various driving conditions, providing a level of precision and response time that surpasses human abilities. The architecture of an AV system is built on four key components: perception, localization, planning, and control, as illustrated in Figure 3.

The perception component relies on sensors such as LiDAR, cameras, and RADAR to detect and classify objects in the environment. Localization ensures the vehicle knows its precise position on the road, often using a combination of GNSS, INS, and HD maps. The planning component uses this information to generate safe and efficient routes, while the control component translates these plans into physical actions, such as steering and braking, to navigate the vehicle. These components work together to create a robust and reliable autonomous driving system.

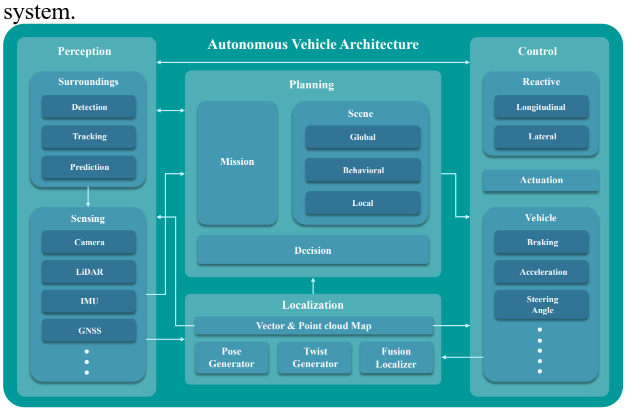

Figure 3. Four pillars AV architecture.

Sensor fusion is a critical aspect of AV systems, as it combines data from multiple sensors to create a more accurate and reliable understanding of the environment. For example, while LiDAR provides high-resolution 3D maps, cameras offer rich visual information, and GNSS/INS systems ensure precise positioning. By integrating these data sources, the system can compensate for the limitations of individual sensors, such as LiDAR's susceptibility to adverse weather or GNSS's vulnerability to signal blockages.

AI plays a pivotal role in AV systems, enabling tasks such as object detection, path planning, and decision-making. Machine learning algorithms, trained on vast datasets, allow the system to recognize and classify objects in real-time, predict the behaviour of other road users, and make safe driving decisions in complex scenarios.

This robust architectural design has driven numerous car manufacturers and tech companies to develop various AV platforms. Current open-source AV platforms, such as Autoware and Apollo, employ a multi-sensor fusion approach, integrating different sensors, including GNSS, IMU, LiDAR, RADAR, cameras, and ultrasonic sensors, to achieve higher levels of autonomous driving. For this study, Autoware was selected due to its open-source nature, regular updates, and user-friendly features, making it an ideal platform for research and development. Its modular architecture allows for seamless integration of custom sensors and algorithms, enabling researchers to test and refine new approaches in a flexible and scalable environment.

1.3 Motivation, Objectives and Contribution

While Autoware strives to meet high standards in autonomous driving, its current design has specific limitations, especially in achieving advanced autonomous navigation levels. Autoware faces challenges due to its heavy reliance on LiDAR and the integration of other sensors like GNSS and IMU, which diminish its effectiveness in environments with signal obstructions. Additionally, the initialization method in Autoware, particularly during point cloud matching, can lead to inefficiencies within its LiDAR-centric framework. The overall cost of the current AV system sensors is also high, limiting its scalability for widespread adoption.

To address these challenges, this study proposes a multi-sensor fusion strategy that integrates INS, GNSS, LiDAR, and HD maps with cost-effective sensors Velodyne VLP-16 and a custom-built EGI Inertial Navigation System EGI-370. The primary focus is on INS, which provides continuous, high-frequency data in any environment, ensuring the AV consistently receives reliable information. FalcoNav.AI approach aims to enhance the real-world application performance, particularly in challenging environments where existing systems struggle. The contributions of this study as following:

1.3.1 Custom Low-Cost EGI and LiDAR Integration: This study integrates a custom-built EGI system and low-cost LiDAR to create a cost-effective hardware framework. The EGI system ensures accurate positioning, while LiDAR provides precise environmental perception. Together, they deliver a robust, affordable solution for autonomous navigation, compatible with FalcoNav.AI and suitable for real-world deployment.

1.3.2 Refinement of Multi-Sensor Fusion Algorithm: This study develops an LC algorithm that integrates INS, GNSS, LiDAR, and HD maps. FalcoNav.AI employs an INS-centric structure to address the limitations of LiDAR-centric approach in Autoware, improving navigation accuracy and reliability in challenging environments.

1.3.3 Realistic Test Field Design: This study creates a carefully designed test field that simulates real-world conditions, including varying GNSS signal strengths and HD maps availability. This setup enables comprehensive and realistic testing, ensuring the navigation system performs reliably in diverse scenarios for autonomous vehicles.

2. Hardware Framework

2.1 Hardware Integration

The hardware integration is meticulously designed to meet the advanced navigation requirements of FalcoNav.AI. The AV platform operates within a Robot Operating System (ROS) environment on a portable Industrial Personal Computer (IPC). Central to this setup is the EGI-370 module, which merges accurate GNSS receiver data with dynamic IMU inputs, ensuring continuous and reliable positioning even in environments with poor GNSS signal quality, such as urban canyons or tunnels.

The EGI-370, embedded within the IPC, serves as the system's computational hub. It processes raw data using a sophisticated encoder-decoder, controlled by a Microcontroller Unit (MCU) and a Real-Time Clock (RTC), and communicates via the EGI ROS driver. This integration with ROS enables seamless data flow and real-time processing, essential for AV navigation. Additionally, the ROS driver for the Velodyne VLP-16 integrates detailed environmental point cloud data, enhancing the system's perception capabilities.

The system employs standard interfaces, including SMA, RS232-USB, and RJ45, to ensure efficient data transmission and synchronization between the EGI-370, LiDAR, and other components. These interfaces enable seamless communication, critical for real-time navigation and control. Figure 4 illustrates the hardware integration architecture, highlighting the connections between the EGI-370, LiDAR, and other components.

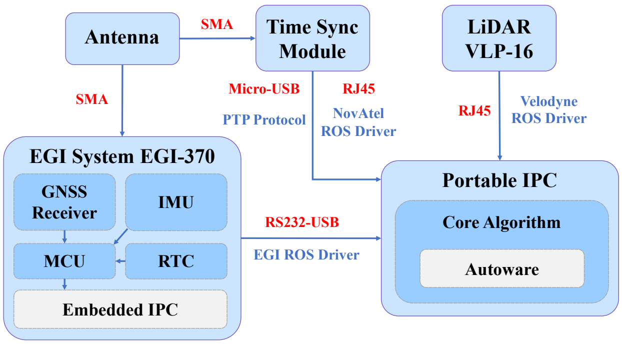

Figure 4. Hardware integration architecture.

2.2 Time Synchronization Module

In precision-critical industrial environments, the Precision Time Protocol (PTP) is essential for synchronization. This protocol is particularly vital for portable Industrial Personal Computer (IPC), which play a key role in data acquisition and processing. IPC depends on precise timing to synchronize various networked devices and sensors, and PTP provides a solution that meets industrial standards for accuracy and reliability. In the proposed system, the core function of PTP is to accurately measure time differences between the master clock and slave clock, as well as to evaluate communication delays.

Operating over an Ethernet framework, the PTP protocol integrates timestamp messages within network packets. Synchronization among peripheral devices is achieved through a systematic exchange of these messages, following a master-slave hierarchy. The protocol ensures time alignment between the master time source, the time synchronization module, and subordinate slave clocks. This precision is critical for maintaining consistency across the network, as illustrated in Figure 5.

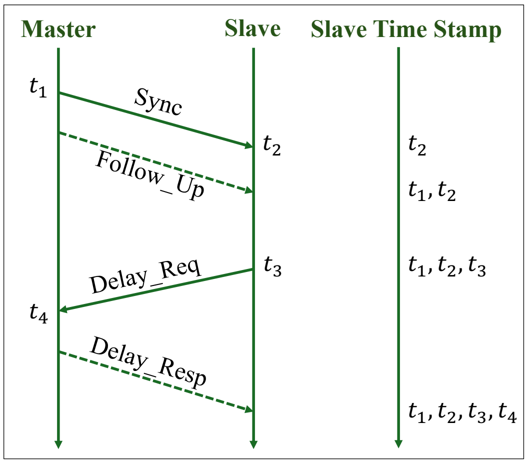

Figure 5. PTP master-slave hierarchy.

The master clock sends a $Sync$ message and records the time t_1 when the message is sent. After the $Sync$ message arrives, the slave clock records the time t_2. The master clock then sends a $Follow\_Up$ message containing the precise time t_1, allowing the slave clock to record both t_1 and t_2. Next, the slave clock sends a $Delay\_Req$ message and records the time t_3 when the message is sent. The master clock receives the $Delay\_Req$ message at time t_4 and responds with a $Delay\_Resp$ message containing the timestamp t_4. Using the timestamps t_1, t_2, t_3, and t_4, the slave clock calculates the offset between the master and slave clocks, as well as the network transmission delay.

$$\begin{cases} delay + offset = t_2 - t_1 \\ delay - offset = t_4 - t_3 \end{cases} \quad (1)$$

The objective is to minimize time discrepancies between the master and slave clocks to within tens of nanoseconds, ensuring precise synchronization. By calculating delay and offset, the system can maintain long-term consistency between the master and slave clocks.

$$delay = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \quad (2)$$

$$offset = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \quad (3)$$

3. Methodology

3.1 Integrated Algorithm

The proposed core algorithm employs an LC-based fusion approach to integrate INS, GNSS, LiDAR, and HD maps, specifically designed for FalcoNav.AI. The process begins with GNSS, which provides position and velocity data to initialize the IMU in both static and dynamic modes. The INS/GNSS integration is then refined using an EKF and motion constraints to enhance accuracy. Direct Georeferencing (DG) further improves precision by transforming IMU data from the body frame (b-frame) to the LiDAR frame (l-frame), refining the initial estimate for Normal Distribution Transform (NDT) matching and improving localization accuracy, as shown in Figure 6.

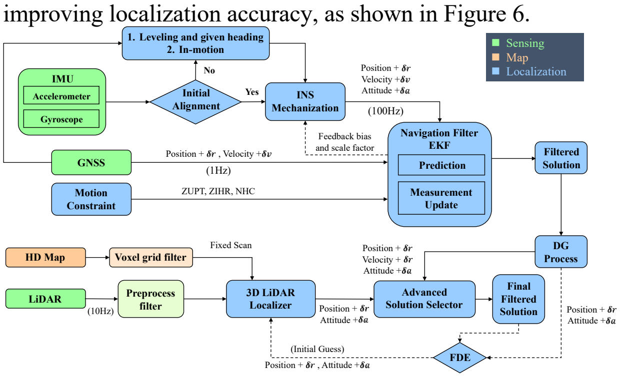

Figure 6. Flowchart of FalcoNav.AI.

To optimize NDT matching, LiDAR-scanned point clouds undergo preprocessing to remove data points that may degrade performance. The HD maps is downsampled to meet ROS environment constraints, ensuring efficient NDT processing. An advanced solution selector then finalizes the filtered solution, generating accurate navigation information.

The flowchart outlines the core fusion algorithm, structured into two main steps. First, it estimates the filter solution through LC-INS/GNSS integration within the EGI system, incorporating motion constraints. Second, it applies this filtered solution for LiDAR-NDT matching using HD maps. The flowchart visually represents different components with green for sensing, flesh for maps, and blue for localization, highlighting the algorithm's seamless integration into the architecture of FalcoNav and its compatibility with advanced autonomous vehicle systems.

3.2 LC-GNSS/INS Integration

Using an EKF model for INS/GNSS integration generally ensures stable results and delivers robust performance across various conditions. The EKF state vector consists of 21 states, enabling precise estimation of the navigation solution, which is expressed as follows:

$$x_k = \begin{bmatrix} \delta r & \delta v & \delta \varphi & \delta b_a & \delta b_g & \delta s_a & \delta s_g \end{bmatrix}^T_{21\times 1} \quad (4)$$

where x_k = error state vector at time k
δr = position error state vector
δv = velocity error state vector
$\delta \varphi$ = altitude error state vector
δb_a = bias error state vector of accelerometer
δb_g = bias error state vector of gyroscope
δs_a = scale factor error state vector of accelerometer
δs_g = scale factor error state vector of gyroscope

In an integrated navigation system, inertial data integration is highly nonlinear, with current states derived from IMU observations using mechanization equations. To implement INS/GNSS navigation, the system model must be linearized to satisfy KF assumptions. The EKF represents this system in a discrete-time form, estimating error states δx as they evolve rather than directly computing navigation states:

$$\delta x_k = \Phi_{k-1,k} \delta x_{k-1} + w_{k-1} \quad (5)$$

where δx_k = estimate errors state at time k
$\Phi_{k,k-1}$ = state transition matrix from time k-1 to k
w_k = process noise at time k-1

Most measurement equations in the integrated navigation system are nonlinear, such as GNSS and vehicle velocity measurement models; thus, it is necessary to linearize the equation, which can be expressed as:

$$\delta z_k = H_k \delta x_k + v_k \quad (6)$$

where: δz_k = measurements error state vector at time k
H_k = design matrix of measurements at time k
v_k = measurements noise at time k

3.3 Motion Constrains

Motion constraints are essential in navigation systems, particularly for ground vehicles where precision and stability matter. By leveraging predictable motion patterns governed by physical laws, these constraints enhance state estimation accuracy. They establish movement rules that serve as reference points for correction algorithms, helping to reduce drift in inertial navigation systems.

Key motion constraints include Zero Velocity Update (ZUPT), which corrects velocity errors during stationary periods; Zero Integrated Heading Rate (ZIHR), which mitigates heading drift when no rotational motion occurs; and the Non-Holonomic Constraint (NHC), which improves path estimation by enforcing a vehicle's non-lateral movement characteristics. These constraints enhance accuracy and stability in navigation systems.

3.3.1 Zero Velocity Update: ZUPT operates on the principle that vehicles undergo stationary periods, such as stopping at traffic signals in urban environments. By detecting these moments, ZUPT corrects velocity errors by resetting velocity estimates to zero, which can reduce error accumulation and improving the overall accuracy of the navigation system over time effectively.

3.3.2 Zero Integrated Heading Rate: ZIHR leverages periods of no rotational movement to minimize heading drift which is a common error that accumulates over time. By assuming that the heading remains constant during stationary phases, ZIHR updates the heading angle using the INS's recorded value at the beginning of the static period, which enhance navigation accuracy effectively.

3.3.3 Non-Holonomic Constraint: NHC is essential in vehicular navigation, reflecting movement limits due to vehicle dynamics. Unless airborne or slipping, a vehicle's lateral velocity remains minimal. This constraint is especially relevant for four-wheeled vehicles which maintain stable attitude angles and conservative turning rates.

3.4 Point Cloud Preprocess

In point cloud preprocessing, LiDAR sensors generate data representing object surfaces, but real-world point clouds often contain errors. Preprocessing enhances accuracy by filtering out noise, aligning the data, and optimizing its structure for more effective analysis.

This step enhances data reliability for techniques like NDT and other computational tasks. The next sections will explore key preprocessing methods, such as error removal, alignment, and optimization for diverse applications. Figure 7 illustrates the results of point cloud preprocessing.

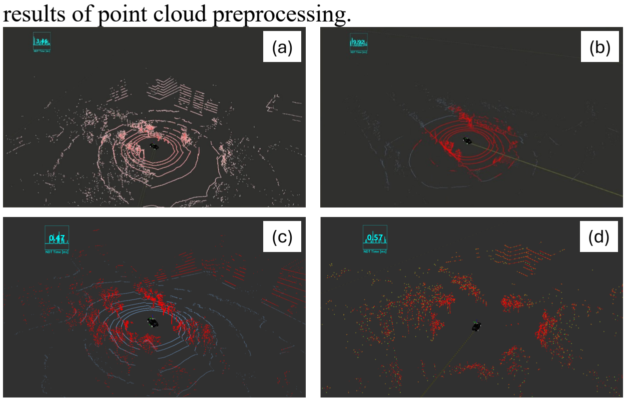

Figure 7. Point cloud preprocess. (a) Outlier filter; (b) ROI filter; (c) Ground point filter; (d) Downsampled filter.

3.4.1 Outlier filter: Outlier filtering in point cloud preprocessing removes anomalous points caused by sensor inaccuracies, environmental factors, or reflections. These outliers introduce noise and errors that affect data quality. By eliminating them, preprocessing enhances the accuracy and reliability of the point cloud for further analysis and applications.

3.4.2 ROI filter: In LiDAR systems, defining a precise ROI mitigates issues like inaccurate laser pulse reflections. Setting minimum and maximum range thresholds removes distant, sparse, and unreliable points, as well as excessively close ones that may cause distortions. This ensures that only relevant and reliable points are processed.

3.4.3 Ground point filter: Ground point filtering removes ground points from point cloud data to improve obstacle detection and clustering accuracy. By focusing on obstacles rather than the road surface, it provides a clearer environmental representation. This process reduces unnecessary data, simplifies analysis, and enhances the detection of critical objects like pedestrians and vehicles while improving computational efficiency.

3.4.4 Downsampled filer: In ROS environments, where point cloud data size is limited, downsampling slightly reduces HD maps accuracy but ensures efficient real-time localization. Using the float data type helps manage storage while maintaining sufficient precision for tasks like NDT matching. The adoption of downsampling filtering in this research, applied to both HD maps and LiDAR-scanned point cloud data, addresses the challenge of managing large datasets in robotics and autonomous systems.

3.5 Direct Georeferencing for LiDAR-IMU Calibration

DG is a mapping and surveying technique that determines a sensor's precise position and orientation relative to the Earth's surface. DG integrates GNSS and INS to establish the sensor's location and orientation within a uniform reference frame. Figure 8 illustrates the geometric relationship between navigational sensors in this land vehicle system.

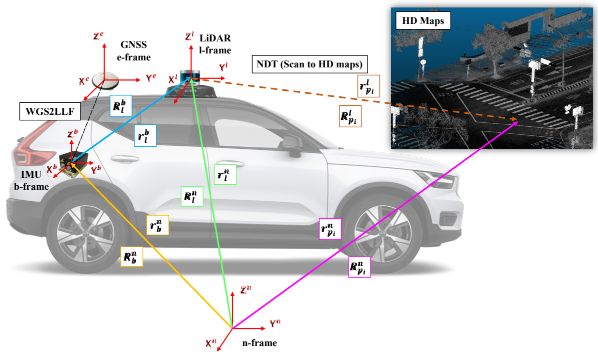

Figure 8. HD maps-based calibration architecture.

In LiDAR applications, DG transforms point cloud data from its native coordinate system to a mapping coordinate system, typically aligned with East-North-Up or North-East-Down orientations. Accurate navigation state estimation depends on converting LiDAR scan coordinates from the LiDAR frame (l-frame) to the navigation frame (n-frame).

$$r_l^n = r_b^n + R_b^n r_l^b \tag{7}$$

$$r_{p_i}^n = r_l^n + R_b^n (R_l^b \times r_{p_i}^l) \tag{8}$$

where: r_l^n = translation vector from n-frame to l-frame
r_b^n = translation vector from n-frame to b-frame
r_l^b = translation vector from b-frame to l-frame
$r_{p_i}^n$ = translation vector from n-frame to HD maps
$r_{p_i}^l$ = translation vector from l-frame to HD maps
R_b^n= rotation matrix from n-frame to b-frame
R_l^b= rotation matrix from b-frame to l-frame

This process determines lever arms r_l^b and boresight angles R_l^b. Calibration uses a high-definition point cloud map, integrating mounting parameters into the georeferencing equation at each epoch.

https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-633-2025 | © Author(s) 2025. CC BY 4.0 License.

3.6 3D-Normal Distribution Transform

NDT matching methods are divided into Point-to-Distribution (P2D-NDT) and Distribution-to-Distribution (D2D-NDT). P2D-NDT, the more common method, was developed for 2D scans and later adapted to 3D. It aligns scan data by matching points from the current frame to a reference distribution, enabling efficient registration through numerical optimization.

In P2D-NDT, the scanned area is divided into uniform cells. For cells containing enough points, the average position and covariance matrix are calculated to create a probabilistic model that estimates the likelihood of a point’s position.

$$\vec{\mu} = \frac{1}{m}\sum_{k=1}^{m} \vec{x}_k \tag{9}$$

$$\Sigma = \frac{1}{m-1}\sum_{k=1}^{m}(\vec{x}_k - \vec{\mu})(\vec{x}_k - \vec{\mu})^T \tag{10}$$

where: $\vec{x}_k$ = scan points contained in the cell
$\vec{\mu}$ = mean vector of specific cell
Σ = covariance matrix of specific cell

The process creates a Probability Density Function (PDF) for each cell to represent the distribution of points within it. This PDF models how surface points are generated in the cell. Assuming the reference scan points follow a normal distribution, the probability of observing a point at a specific position is calculated as follows:

$$p(\vec{x}) = \exp\left(-\frac{(\vec{x} - \vec{\mu}_k)^T \Sigma^{-1} (\vec{x} - \vec{\mu}_k)}{2}\right) \tag{11}$$

where: $p(\vec{x})$ = $\vec{x}_k$’s distribution probability within specific cell

The likelihood of a point belonging to the k^{th} cell is determined using a Gaussian-based scoring function, contributing to the NDT score. The source point cloud is transformed with a pose vector $\vec{p}$, and alignment is evaluated by summing the scores of all transformed points. Newton's method then optimizes $\vec{p}$ through an iterative update formula.

$$H\Delta\vec{p} = -\vec{g} \tag{12}$$

where: H = Hessian matrix
$\Delta\vec{p}$ = increment of $\vec{p}$
$\vec{g}$ = gradient vector of the function

4. Field Testing

4.1 Experiment Setup

This study utilizes a ground vehicle to test FalcoNav.AI in various scenarios. The equipment setup includes a NovAtel PwrPak 7D-E2, iMAR iNAV-RQH-10018, EGI-370, NovAtel GPS-703-GGG, and a Velodyne VLP-16, as shown in Figure 9.

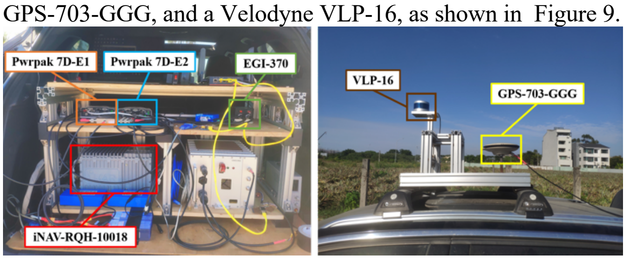

Figure 9. Experiment setup.

This research compares the LiDAR-centric Autoware algorithm with the proposed INS-centric FalcoNav.AI approach, as shows in Table 1. By leveraging the characteristics of inertial navigation, which is less affected by environment and supports high-frequency sampling, the proposed method achieves more stable localization results across various environments. The equipment specifications are provided in Table 2 and Table 3.

	Reference	Autoware	FalcoNav.AI
GNSS	NovAtel	EGI-370	EGI-370
IMU	iMAR	EGI-370	EGI-370
LiDAR		Velodyne	Velodyne
HD maps	X	V	V
Software	Inertial Explorer	Autoware Localization Module	Proposed Algorithm

Table 1. Comparison of difference algorithm.

iMAR iNAV-RQH-10018	
Gyroscopes	
Bias instability	$< 0.002\ (deg/hour)$
Angular random walk	$< 0.0015\ (deg/\sqrt{hour})$
Accelerometers	
Bias instability	$< 10\ (\mu g)$
Velocity random walk	$< 8\ (\mu g/\sqrt{Hz})$

Table 2. Specification of iMAR iNAV-RQH-10018.

EGI-370	
Gyroscopes	
Bias instability	$< 0.8\ (deg/hour)$
Angular random walk	$< 0.06\ (deg/\sqrt{hour})$
Accelerometers	
Bias instability	$10\ (\mu g)$
Velocity random walk	$< 0.25\ (s/\sqrt{hour})$

Table 3. Specification of EGI-370.

4.2 Scenario Description

This study presents two evaluation scenarios, as shown in Figure 10. The first scenario is located around National Cheng Kung University (NCKU) campus in Tainan, Taiwan, where the route passes through an underground parking lot which is a GNSS-denied area. The second scenario is at in Tainan Shalun, where GNSS signals are influenced by multipath effects caused by roadside trees and buildings. Both scenarios include open sky environments and GNSS challenging conditions, assessing the feasibility and stability of the proposed algorithm and hardware structure.

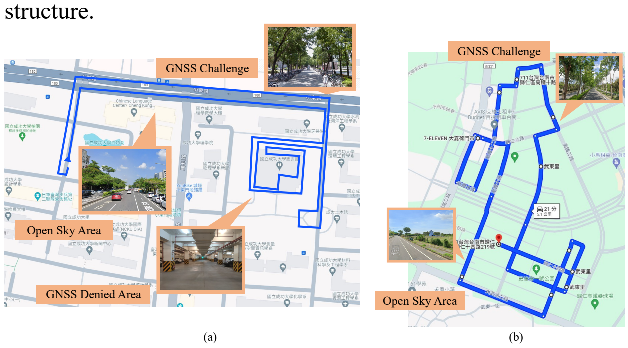

Figure 10. Experiment routes.
(a) Around NCKU campus; (b) Tainan Shalun.

5. Result and Discussion

5.1 Scenario 1: Around NCKU Campus

In the analysis of NCKU campus, Autoware algorithm experiences a blunder when entering the underground parking lot. In contrast, FalcoNav.AI algorithm remains close to the reference trajectory. Figure 11 illustrates a trajectory comparison around NCKU, which the red trajectory represents the reference, the blue trajectory represents Autoware algorithm, and the purple trajectory represents FalcoNav.AI algorithm.

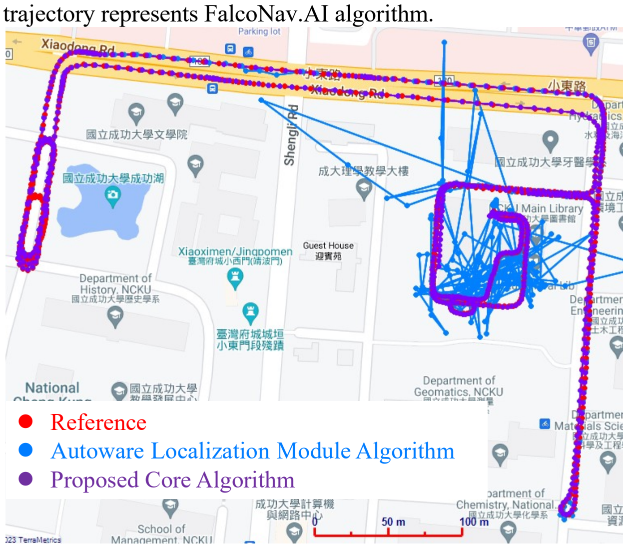

Figure 11. Trajectory Comparison around NCKU campus.

Although both the Autoware and proposed algorithms use the same point cloud matching method, they differ in how they provide the initial guess. Figure 12 focuses on areas of successful and failed NDT matching. (a) shows the open sky area around NCKU campus, where correct matching occurs due to excellent GNSS reception. (b) displays an irregular and unstable trajectory caused by GNSS multipath effects from trees and buildings along the roadside. (c), Autoware algorithm shows significant error as the route enters an underground parking lot, where GNSS reception is unavailable. This leads to NDT matching failure due to a poor initial guess, with errors ranging from tens to hundreds of meters.

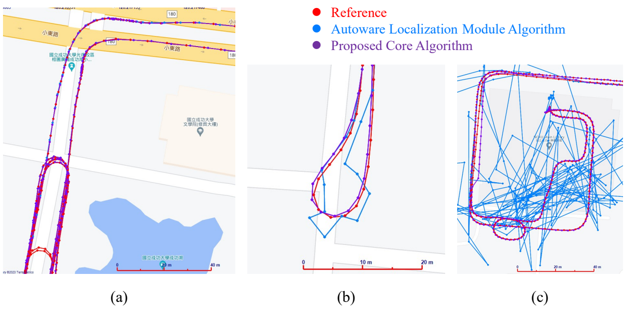

Figure 12. Comparison of NDT matching around NCKU campus. (a) Matching successfully; (b) (c) Matching failed.

Table 4 presents a comparative analysis of the positioning accuracy between Autoware algorithm and FalcoNav.AI, especially with the integration of the EGI-370, demonstrates high performance. It offers a more accurate initial guess for point cloud matching compared to Autoware algorithm, which relies on GNSS assistance. FalcoNav.AI achieves high accuracy with RMSE of 0.495 meters in horizontal and RMSE of 0.555 meters in 3D.

Methods	RMSE (m)		
	U	2D	3D
Autoware	7.844	48.399	49.031
FalcoNav.AI	0.240	0.495	0.550

Table 4. Localization results around NCKU campus.

5.2 Scenario 2: Tainan Shalun

In the analysis of the Tainan Shalun area, Autoware algorithm experiences noticeable deviations when navigating through many roadside trees or buildings area. In contrast, the proposed core algorithm maintains a trajectory closely aligned with the reference path, demonstrating better localization accuracy. Figure 13 illustrates a trajectory comparison around Tainan Shalun, with the red trajectory represents the reference, the blue trajectory represents Autoware algorithm, and the purple trajectory represents FalcoNav.AI algorithm.

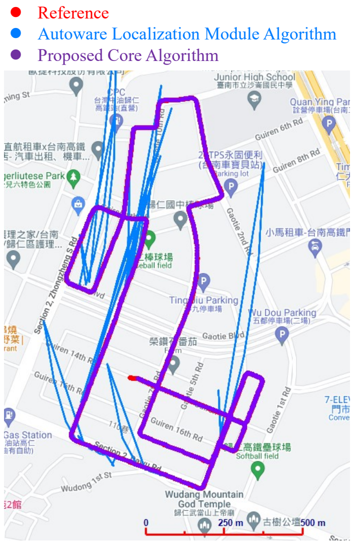

Figure 13. Trajectory comparison in Tainan Shalun.

Figure 14 focuses on the performance of NDT matching in both successful and challenging areas. In (a), the open sky area in Tainan Shalun allows for successful matching due to strong GNSS reception, resulting in accurate trajectories for both algorithms. In contrast, (b) depicts a route that passes through areas with dense roadside trees and surrounding buildings, causing GNSS multipath effects that lead to irregular trajectories for Autoware algorithm. However, FalcoNav.AI still demonstrates a smoother and more stable trajectory, closely following the reference path despite the challenging environment.

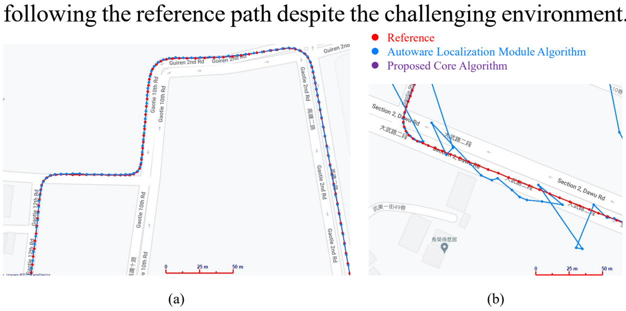

Figure 14. Comparison of NDT matching in Tainan Shalun. (a) Matching successfully; (b) Matching failed.

Table 5 provides a comparative evaluation of the positioning accuracy between Autoware and FalcoNav.AI algorithm. The results clearly indicate that FalcoNav.AI outperforms Autoware algorithm, delivering significantly improved localization accuracy. While Autoware algorithm exhibits higher RMSE values, particularly in the 2D and 3D dimensions, FalcoNav.AI maintains superior performance with minimal errors. Eventually, FalcoNav.AI achieves an RMSE of 0.475 meters in 2D, and 0.501 meters in 3D.

Methods	RMSE (m)		
	U	2D	3D
Autoware	4.134	94.783	94.873
FalcoNav.AI	0.161	0.475	0.501

Table 5. Localization results in Tainan Shalun.

6. Conclusions

This study develops FalcoNav.AI, a robust multi-sensor navigation system based on INS, integrating GNSS, LiDAR, and HD maps to improve accuracy. Unlike the LiDAR-centric approach in Autoware, the proposed system uses INS as the core sensor, providing high-frequency data unaffected by external conditions. This reduces errors common in LiDAR-unfriendly or poor GNSS environments.

Testing includes open-sky, GNSS-challenged, and GNSS-denied scenarios to evaluate the real-time performance of the custom-built EGI-370 and the LC-GNSS/INS algorithm. The proposed algorithm delivers better accuracy, especially in challenging environments like underground parking lots and GNSS multipath areas. By using the EGI-370 as the core sensor, it addresses NDT matching issues in Autoware and minimizes the impact of weak GNSS signals. Test results confirm the system achieves "Where-in-Lane" level accuracy, demonstrating its reliability and robustness in real-world conditions.

Acknowledgements

The authors extend their gratitude to Gaia Technology for invaluable technical assistance. We also wish to express their appreciation for the financial support received from the Ministry of the Interior within the Executive Yuan in Taiwan.

References

Bimbraw, K. (2015, July). Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*.

SAE International (2021). "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles".

Stephenson, S., Meng, X., Moore, T., Baxendale, A., & Ford, T. (2011, November). Accuracy requirements and benchmarking position solutions for intelligent transportation location based services. In *Proceedings of the 8th international symposium on location-based services.*

Zong, W., Zhang, C., Wang, Z., Zhu, J., & Chen, Q. (2018). Architecture design and implementation of an autonomous vehicle. IEEE access, 6, 21956-21970.

T. G., Pervez, N., Ibrahim, U., Houts, S. E., Pandey, G., Alla, N. K., & Hsia, A. (2019). “Standalone and RTK GNSS on 30,000 km of North American Highways”. In *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation ION GNSS+ 2019*.

AutowareFoundation (2021). Autoware-documentation. https://github.com/autowarefoundation/autoware-documentation

ApolloAuto (2017). Apollo. https://github.com/ApolloAuto/apollo

Raju, V. M., Gupta, V., & Lomate, S. (2019). Performance of open AV platforms: Autoware and Apollo. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*.

Reshetyuk, Y. (2009). *Self-calibration and direct georeferencing in terrestrial laser scanning* (Doctoral dissertation, KTH).

Shin, E. H. (2005). Estimation techniques for low-cost inertial navigation.

Chiang, K., Chiu, Y., Srinara, S. et al. Performance of LiDAR-SLAM-based PNT with initial poses based on NDT scan matching algorithm. Satell Navig 4, 3 (2023).

Biber, P., & Straßer, W. (2003, October). The normal distributions transform: A new approach to laser scan matching. *In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453)*.

Magnusson, M. (2009). The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection (Doctoral dissertation, Örebro universitet).

K.W. Chiang, G.J. Tsai, H.W. Chang, C. Joly, N. EI-Sheimy, Seamless navigation and mapping using an INS/GNSS/grid-based SLAM semi-tightly coupled integration scheme.

Srinara, S., Chiu, Y. T., Tsai, M. L., & Chiang, K. W. (2022). High-definition point cloud map-based 3D LiDAR-IMU calibration for self-driving applications. The *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43*, 271-277.

Huang, F., Wen, W., Zhang, J., & Hsu, L. T. (2022). Point wise or feature wise? A benchmark comparison of publicly available LiDAR odometry algorithms in urban canyons. *IEEE Intelligent Transportation Systems Magazine, 14*(6), 155-173.

Huang, Y. E., Tsai, S., Liu, H. Y., Chiang, K. W., Tsai, M. L., Lee, P. L., & El-Sheimy, N. (2023). The Development and Validation of a Tactical Grade EGI System for Land Vehicular Navigation Applications. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 48*, 821-828.

Yin, Y., Zhang, J., Guo, M., Ning, X., Wang, Y., Lu, J. (2023). Sensor Fusion of GNSS and IMU Data for Robust Localization via Smoothed Error State Kalman Filter. *Sensors, 23*(7), 3676.ghn bftdvrceswxaqz`13.