

Demonstrating the Efficiency of RKO-LIO on a DJI UAV Platform Utilizing Drone Built-in IMU for GPS-Denied Mapping

Mohammed Ali Mohammed S. Bajhaw
Mechatronics Engineering
Department, Firat University
Elazig, Turkey,
mohammedbajhaw2020@gmail.com

Anas Mahyoub Naji Saeed Alqadhi
Mechatronics Engineering
Department, Firat University,
Elazig, Turkey,
anas.m.qd@gmail.com

Munef el Muhammed
Mechatronics Engineering
Department, Firat University
Elazig, Turkey,
munafa662@gmail.com

Aysegul Ucar
Mechatronics Engineering
Department, Firat University
Elazig, Turkey,
agulucar@firat.edu.tr

Abstract— *This work presents a hardware and software integration framework for Unmanned Aerial Vehicles (UAVs) operating in GPS-denied environments. We detail the integration of a DJI Matrice M350 RTK drone with an NVIDIA Jetson Orin NX edge computing platform using Robot Operating System (ROS 2). To minimize payload cost, weight, and mechanical complexity, the system relies exclusively on a Velodyne VLP-16 LiDAR whose data is fused with telemetry from the drone's built-in high-precision Inertial Measurement Unit (IMU) nominally streaming at 200 Hz, removing the need for expensive external IMUs. We deploy RKO-LIO, a recent scan-to-map method that replaces sensor-specific noise modeling with a simplified IMU motion model and an adaptive gravity-orientation regularizer. From a 458.49-meter outdoor flight reaching 38.28 m altitude we report three findings. First, the map produced onboard during flight reached an average structural error of 1.57% against laser-rangefinder measurements of six facade segments, even though latency spikes caused 41.3% of the LiDAR scans to be skipped. Second, the method's default 0.5 m association threshold is conservative for aerial operation: the inter-beam spacing of a 16-beam LiDAR exceeds it beyond 14 m range, where 52.8% of our returns lie, and the registration rejects 2.0% of point correspondences at 0.5 m against 0.1% at 5.0 m. Relaxing it reduces the final landing-point position error, validated against a physical ground measurement, from 1.58 m to 0.31 m (0.343% to 0.067% of path length). Third, we characterize the timestamp fidelity of the DJI built-in IMU, delivered in bursts with 12.2% of samples arriving after gaps exceeding 15 ms.*

Keywords— *UAV, Edge Computing, ROS 2, RKO-LIO, Hardware Integration, Sensor Fusion, GPS-denied Navigation, DJI E-Port*

I. INTRODUCTION

In recent years, the deployment of Unmanned Aerial Vehicles (UAVs) has expanded across various domains, including search and rescue, industrial inspection, and 3D mapping. The majority of these applications rely heavily on the Global Positioning System (GPS) for localization and navigation. However, in complex environments such as dense urban canyons or GPS-denied zones surrounding buildings,

GPS signals degrade or vanish entirely, necessitating alternative sensing systems for reliable operation [1].

Simultaneous Localization and Mapping (SLAM) algorithms provide an ideal solution. Among the available technologies, LiDAR (Light Detection and Ranging) has emerged as a powerful tool for constructing accurate 3D maps due to its illumination invariance and precise distance measurement capabilities. Nevertheless, relying solely on LiDAR presents challenges, notably motion skew distortion and poor performance in feature-poor environments [2-5].

To overcome these limitations, LiDAR data is typically fused with Inertial Measurement Unit (IMU) data in what is known as Lidar-Inertial Odometry (LIO). While popular algorithms such as LOAM, LIO-SAM and FAST-LIO2 [2,3,6] have proven highly successful, they suffer from accumulated drifts when encountering environments with dynamic moving objects. Furthermore, these methods often require expensive, high-frequency external IMUs (e.g., Pixhawk Cube Orange) to maintain stability. Recent works have been focusing new LiDAR-inertial SLAM methods and navigations [7,8] using ROS2 and reinforcement learning in GPS denied environments.

In this context, this paper presents three main contributions: (1) A robust and cost-effective hardware integration that merges a DJI M350 RTK drone with an NVIDIA Jetson Orin NX. By relying entirely on the drone's internal IMU, nominally streaming at 200 Hz, via the DJI Payload SDK rather than an external unit, we remove the cost, weight and calibration burden of a dedicated inertial unit. (2) A quantitative characterization of the telemetry path itself: we show that the built-in IMU reaches ROS 2 in bursts rather than at a uniform 200 Hz, with 12.2% of samples arriving after gaps exceeding 15 ms, three times the nominal period. (3) The practical evaluation of RKO-LIO [5] on an aerial platform, where the onboard map reached an average facade-measurement error of 1.57% and where we find that its default association threshold is conservative for the sampling geometry of a 16-beam LiDAR at altitude. Relaxing it lowers the final landing-point position error from 1.58 m to 0.31 m over a 458.49-meter flight. This study differs from our earlier indoor work [9] in platform, environment and inertial source: [9] used an external Pixhawk IMU indoors, whereas the present system uses no external IMU and operates outdoors at up to 38.28 m altitude.

II. HARDWARE INTEGRATION AND ARCHITECTURE

Achieving real-time remote-controlled mapping requires a hardware system capable of providing high computational power while adhering to the strict weight, power consumption, and cost constraints imposed by UAVs.

A. UAV Platform

The DJI Matrice M350 RTK was selected as the primary platform due to its high reliability in industrial applications, its ability to carry a payload of up to 2.7 kg, and its resistance to harsh weather conditions (IP55 rating) (Fig.1). The drone provides an E-Port interface that allows direct integration with custom payloads, facilitating both power and data transfer.



Fig. 1. The complete 3D mapping payload mounted on the DJI Matrice M350 RTK UAV

B. Edge Computing Platform

To analyze the massive data generated by the LiDAR sensor and run complex ROS 2 SLAM algorithms, the system is equipped with an NVIDIA Jetson Orin NX module (16GB RAM) (Fig.2). This module is an ideal choice as it provides an excellent balance between efficient power consumption and high performance (up to 100 TOPS), in addition to its built-in support for JetPack SDK, CUDA, and TensorRT libraries.

C. LiDAR Sensor

We utilized a Velodyne VLP-16 LiDAR sensor, providing 360-degree coverage and generating a precise point cloud at a high recovery rate of 10 Hz (Fig.2). The sensor is connected to the Jetson Orin NX via an Ethernet cable to transmit data packets at high speeds, while being powered through a dedicated voltage step-down circuit. Table I shows properties of the Jetson Orin NX.

D. Built-in IMU Integration

Instead of adding an external high-frequency IMU sensor (like a MicroStrain or Pixhawk Cube Orange)—which increases calibration complexity, hardware cost, and payload weight—a novel mechanism was devised to extract data from the high-precision built-in IMU located inside the DJI M350 drone. The inertial data is carried from the drone's flight controller to the Jetson unit over the native telemetry link, so no separate clock synchronization between an external IMU and the flight state is required. The interface advertises a 200 Hz stream; our timestamp analysis in Section V-D shows that samples in fact arrive in bursts rather than at a uniform 5 ms interval. This transport characteristic must therefore be

accounted for during inertial integration and scan deskewing, while still removing the need for an external IMU.



Fig. 2. Main payload components: (a) the NVIDIA Jetson Orin NX; (b) VLP16 Lidar sensor.

TABLE I. COMPUTATIONAL RESOURCE PROFILING ON NVIDIA JETSON ORIN NX

Metric	Value
Computing Platform	NVIDIA Jetson Orin NX (16GB)
OS / Middleware	Ubuntu 22.04 / ROS 2 Humble
LiDAR Processing Frequency	10 Hz
Nominal IMU Data Rate	200 Hz
Average CPU Utilization	25.79% (Peak: 31.9%)
Average RAM Usage	~1.99 GB / 1,997.8 MB (Peak: 2.01 GB / 2,011.7 MB)
Core Operating Temperatures	CPU Average: 56.47°C / GPU Average: 53.94°C

E. Communication Architecture

For communication, the USB Bulk protocol was allocated for high-bandwidth data, connecting the E-Port output to the Jetson's USB-A port to ensure stable and fast data transfer. Furthermore, the UART protocol was used as a vital, low-latency data link for direct drone control, receiving the IMU data, and extracting instantaneous takeoff coordinates at a baud rate of 230400 bps. This configuration allows the system to acquire the built-in IMU data at its nominal 200 Hz rate, providing the dense motion constraints required for accurate Lidar-Inertial Odometry without external sensors.

III. SOFTWARE FRAMEWORK AND MIDDLEWARE

To link the physical components and achieve synchronization, a software architecture based on ROS 2 (Humble version) was built, providing a robust infrastructure for inter-process communication [10].

A strict Transform (TF) Tree structure was defined to ensure the success of the mapping algorithms. The central point ('base_link') represents the drone's center of mass (where the built-in IMU is located). We defined an accurate rigid transformation matrix between the 'base_link' and the LiDAR frame ('velodyne') based on physical millimeter measurements of the mounting location. This static description ensures that vibration or deviation motion sensed by the built-in IMU is accurately and mathematically compensated on the LiDAR point cloud during the odometry calculations.

IV. LIDAR-INERTIAL ODOMETRY WITHOUT SENSOR-SPECIFIC MODELING

The primary goal of localization techniques is to continuously estimate the change in the robot's 6-DOF pose (translation and rotation) with high precision.

A. Limitations of Traditional SLAM Algorithms

Common algorithmic frameworks like LOAM and LIO-SAM rely on extracting edge and planar features and matching them to correct drift. However, in real-world environments, the resulting point cloud often contains visual noise caused by "dynamic objects" (like walking people, swaying trees, or moving vehicles) or "distorted features" (like shiny metal pipes that specularly reflect laser beams). Traditional algorithms often treat these erroneous points as static environmental features, leading to accumulated drift, ghosting artifacts, and overall map distortion.

B. Principles of RKO-LIO and its Adaptive Regularization

To solve this deficiency, we adopted the RKO-LIO algorithm [5]. As demonstrated by Malladi et al. [5] and applied to a UAV platform in our earlier indoor study [9], RKO-LIO replaces sensor-specific noise modeling with a physics-based kinematic motion model and an adaptive gravity-orientation regularization. The method explicitly avoids both the iterated Kalman filtering used by FAST-LIO2 [6] and the pose-graph optimization used by LIO-SAM [2], which is what allows it to operate without a characterized IMU noise model. Accelerometer and gyroscope samples between two consecutive LiDAR frames are averaged and integrated with a constant-acceleration model; the resulting relative pose both deskews the scan and initializes the registration.

Registration minimizes the unweighted mean squared point-to-point residual against a VDB voxel map [11], following the point-to-point scheme of KISS-ICP [4], whose scan-alignment module RKO-LIO adopts. Outliers, including returns from dynamic objects such as moving vehicles and swaying foliage, are rejected by a hard association gate: a point contributes to the optimization only if its nearest map neighbor lies within a fixed distance, and is discarded entirely otherwise. There is no continuous down-weighting function. The novel component of the method is an adaptive gravity-orientation term added to the cost and weighted by $1/\beta$, where $\beta = \beta_0(1 + \sigma_a^2)$ and σ_a is the standard deviation of the accelerometer magnitude over the frame interval. When the platform vibrates or accelerates aggressively, σ_a grows and the gravity prior is automatically de-weighted in favor of the LiDAR measurement.

Fig. 3 shows why this fixed distance matters for a 16-beam LiDAR such as the VLP-16, whose adjacent beams are 2° apart vertically. On a surface at range R they land $R \cdot \tan(2^\circ)$ apart (dark-blue line), which exceeds the 0.5 m default gate (dashed red line) beyond $R = 14.3$ m, so the gate becomes restrictive for returns farther away. The light-blue bars show that 52.8% of the returns of our flight lie beyond that range. This is a geometric argument whose measured effect is given in Section V-C; a point rejected by the gate is left out of the pose correction only, while the full scan is still inserted into the local map.

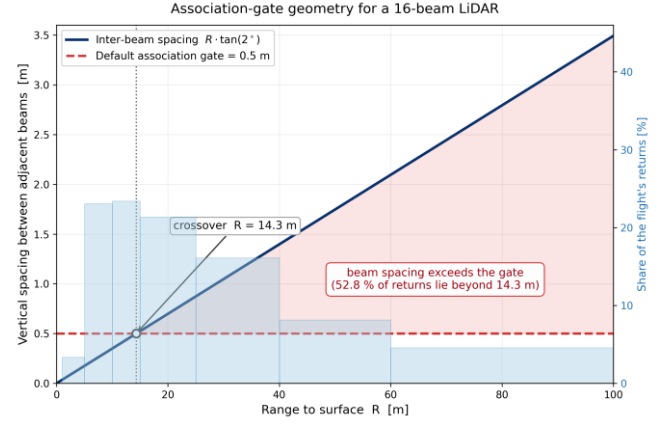


Fig. 3. Association-gate geometry for a 16-beam LiDAR: spacing of adjacent beams versus the 0.5 m default gate, and share of the flight's returns per range interval.

V. FIELD EXPERIMENTS AND VALIDATION

To evaluate the robustness, mapping accuracy, and performance of the integrated hardware system and the RKO-LIO algorithm, a rigorous field experiment was conducted in a dynamic outdoor environment surrounding the university campus buildings (Figs. 4, 6).

A. Dynamic Outdoor Flight Experiment and Facade Trajectory Analysis

The UAV payload was integrated with the DJI Matrice M350 RTK drone and flown in a mixed outdoor environment. The flight trajectory covered a total path length of 458.49 meters over a flight duration of 484.21 seconds (8 minutes and 4 seconds). This environment presented severe dynamic interference, containing swaying tree branches, complex facade geometries, and active moving vehicles in the parking lot. The drone executed 6-DOF maneuvers, recording 4,802 LiDAR scans at an average speed of 0.95 m/s and a peak velocity of 3.95 m/s. The complete offline reconstruction yields one pose per scan, whereas the onboard run produced 2,819 poses and ended 29.15 s before touchdown. The onboard map is the one used for the facade validation below, and the complete reconstruction is used for the parameter study of Section V-C.

Fig. 4 compares a satellite view of the site (left) with the top-down view of the onboard point cloud (right), colored by height: the main building, its neighbors and the rows of trees are recognizable in both, while the roof interior is empty because it was barely scanned.

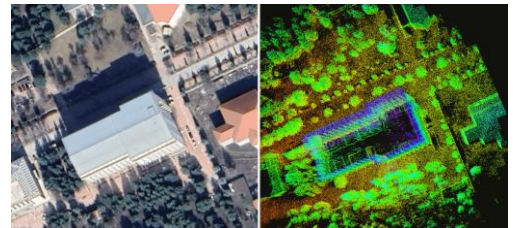


Fig. 4. Aerial view of the university buildings from Google Earth (satellite reference) compared side-by-side with the RKO-LIO reconstructed 3D global point cloud map.

For the onboard trajectory of Fig. 5, the spatial mapping coverage was evaluated, yielding a 2D bounding-box area of 5,673.91 m² with a total 3D bounding-box volume of 190,343 m³. The onboard run reached a maximum altitude Z_{max} of 33.55 meters above the takeoff point (38.28 m in the complete reconstruction), with a maximum vertical speed V_z of 3.92 m/s. Flight kinematics analysis revealed an average acceleration of 1.23 m/s², a peak acceleration of 6.46 m/s² (gravity-compensated, taken directly from the IMU), and a velocity standard deviation σ_v of 0.63 m/s, indicating that the platform was exercised over a wide range of dynamics during the mapping run.

Fig. 5 shows the trajectory estimated onboard (2,819 poses over 455.1 s), colored by altitude: a loop around the building, about 100 m in X and 58 m in Y, mostly between 0.5 and 14 m altitude, with two steep climbs (yellow-orange) to 32.2 m and 33.5 m. The green marker is the take-off point (frame origin); the red marker, at (9.79, -0.75, 3.69) m, is the last onboard pose, reached 29.15 s before touchdown, and thus not the landing point evaluated in Section V-C.

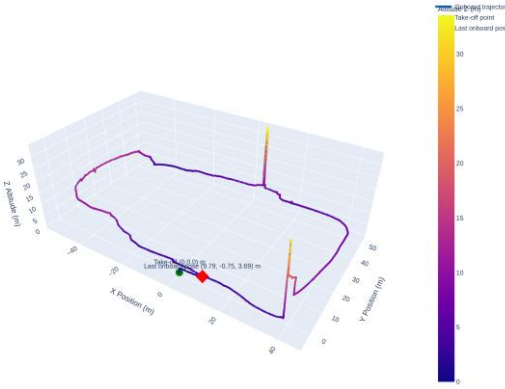


Fig. 5. 3D trajectory estimated onboard (2,819 poses), colored by altitude. Green: take-off point; red: last onboard pose.

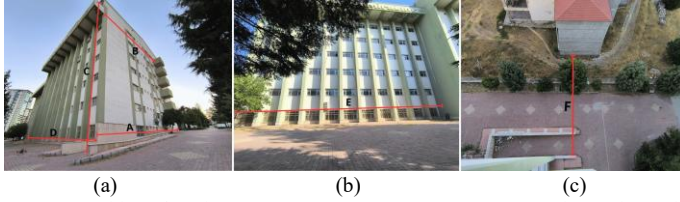


Fig. 6. Building facade measurement segments A-F annotated on physical photographs of the university buildings, used for comparing real-world measurements with RKO-LIO point cloud estimations.

TABLE II. COMPARISON OF PHYSICAL GROUND TRUTH AND RKO-LIO DISTANCE ESTIMATIONS

Segment	Real-World Distance (m)	RKO-LIO Map Distance (m)	Absolute Error (m)	Relative Error (%)
A	38.87	38.50	0.37	0.95%
B	26.21	26.50	0.29	1.11%
C	27.59	28.10	0.51	1.85%
D	36.41	37.31	0.90	2.47%
E	39.91	40.35	0.44	1.10%
F	24.10	24.57	0.47	1.95%

To validate the geometric accuracy of the resulting 3D point cloud, six outer building facade segments (labeled A through F, Fig. 6) were measured using high-precision laser rangefinders

to establish the physical ground-truth distances. These were compared directly with the corresponding distances extracted from the RKO-LIO reconstructed 3D point cloud model. The comparative results are presented in Table II. The absolute errors for segments B, C, D, E, and F range between 0.29 m and 0.90 m, representing relative errors below 2.5%. Segment A, representing a larger 38.87-meter span, registered a 0.37-meter absolute error, a relative error of 0.95%. The six facade-segment measurements in Table II yield an average relative structural error of 1.57% and a maximum of 2.47% across all segments.

In Fig. 6 the six segments are drawn as red lines on photographs of the building: A, D and E along the lower part of the facade, B across its upper part, C along a vertical edge, and F on the ground in a top-down view.

B. Computational Resource Profiling on Edge Hardware

A critical constraint for real-time UAV mapping is the edge processing capability. We profiled the computational resource consumption of the entire software stack—including the psdk_ros2 node, LiDAR pre-processing, and the RKO-LIO optimization—running on the onboard NVIDIA Jetson Orin NX (16GB) during the 458.49-meter flight.

Table I summarizes the measurements recorded from the hardware sensors: the entire mapping pipeline consumed an average of 25.79% CPU (peak 31.9%) and 1,997.81 MB of RAM (about 1.99 GB, 12.19% of the 16 GB total; peak 2,011.7 MB, 12.3%).

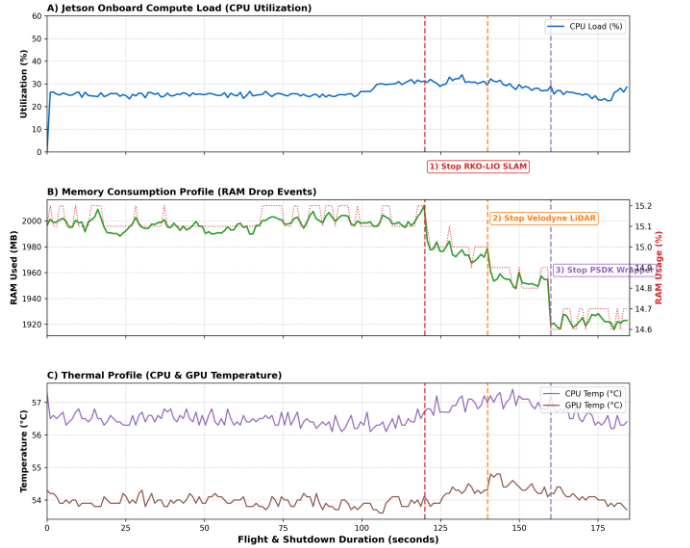


Fig. 7. Onboard hardware resource utilization profile on the Jetson Orin NX showing CPU, RAM and temperature during the final portion of the mission and the subsequent sequential component shutdown.

The CPU figure is averaged over the eight cores and was recorded while a full-rate sensor bag (≈ 18 MB/s) was being written concurrently, so deployment load is strictly lower; it should not be read as real-time margin (see Section V-D). Under active flight dynamics and a 15W operational profile, the CPU and GPU temperatures stabilized at an average of 56.47°C (peak 57.3°C) and 53.94°C (peak 54.3°C) respectively, ensuring thermal safety and system stability (Fig. 7).

C. Parameter Sensitivity and Validation Against Ground Truth

The landing point of the aircraft was measured on site and expressed in the take-off frame as $(X, Y, Z) = (8.46, -1.20, -0.16)$ m. Because the trajectory is estimated without any satellite or motion-capture reference, this single point is the only external datum available, and we therefore report the final position error rather than an absolute trajectory error. Table III sweeps the two parameters that govern the registration, namely the association gate and the regularization weight β_0 , with everything else, including the input data, held identical.

Relaxing the gate from the 0.5 m default to 1.0 m or beyond lowers the error by roughly a factor of five. The last column of Table III gives the share of point correspondences rejected by the gate, counted inside the registration for the four gate settings with DD on: it falls from 14.5% at 0.25 m to 2.0% at 0.5 m, 0.4% at 1.0 m and 0.1% at 5.0 m, in step with the error. In Table III, DD denotes the optional double-downsampling step described in Section IV-B. This result follows from the vertical angular resolution of the sensor, approximately 2° for a 16-beam scanner, and therefore depends on flight altitude and on the resulting point density; it should not be read as a claim that a 0.5 m gate is inappropriate for ground-level platforms, for denser scanners, or for other registration schemes.

TABLE III. FINAL POSITION ERROR AGAINST THE MEASURED LANDING POINT

Gate [m]	DD	3-D error [m]	% of path	Rejected [%]
0.25	on	3.489	0.710	14.5
0.50 (default)	on	1.583	0.343	2.0
0.50	off	1.886	0.410	—
1.00	on	0.489	0.107	0.4
5.00	on	0.307	0.067	0.1
5.00	off	0.343	0.075	—
$\beta_0 = 0$	on	1.082	0.236	—
$\beta_0 = 1000$	on	1.256	0.272	—

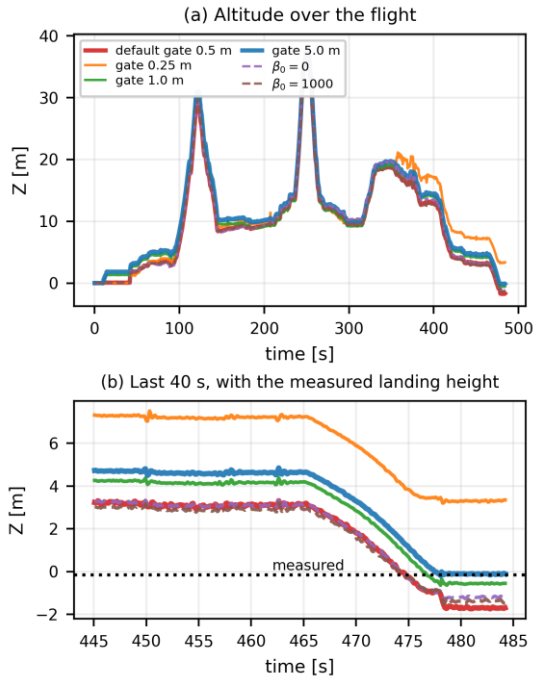


Fig. 8. Estimated altitude for the settings of Table III: (a) whole flight; (b) last 40 s, with the measured landing height (dotted line).

Fig. 8 plots the estimated altitude for the settings of Table III (DD variants omitted). Up to about 350 s all curves stay within about 2.2 m of each other, at altitudes up to 38 m; from about 400 s the 0.25 m gate separates and ends 3.3 m above the take-off altitude. The zoom on the last 40 s adds the measured landing height (-0.16 m, dotted): the 5.0 m gate ends closest to it (-0.11 m), then the 1.0 m gate (-0.56 m), while the default gate ends 1.5 m below it (-1.70 m).

Given a ground-truth uncertainty of about ± 0.15 m the three best rows are not separable from one another; the ordering relative to the default, however, is unambiguous. The systematic $+0.23$ m offset common to every row in the Y axis corresponds to a 1.56° error in the assumed take-off heading and is a property of the measurement frame, not of the odometry; the rotation-invariant horizontal magnitude is 8.545 m measured against 8.401 m estimated.

D. Real-Time Behavior and Frame Loss

The flight-log timestamps allow the in-flight behavior of the pipeline to be reconstructed. The onboard pipeline processed 2,819 of the 4,802 LiDAR scans (41.3% skipped): the mean cost per scan was 80.5 ms, within the 100 ms sensor period, but latency spikes (95th percentile 179.8 ms) combined with a subscriber queue one element deep caused the next arriving scan to be discarded. Nevertheless, the onboard map underlying Table II and Figs. 4 and 6 reached a 1.57% average structural error, so the skipped scans did not prevent accurate mapping in this flight.

The inertial stream itself was also examined. Although the DJI telemetry is advertised at 200 Hz and its mean inter-sample interval of 5.01 ms agrees with that figure, the 96,669 samples arrive in bursts rather than uniformly: 66.2% of the intervals are shorter than 3 ms (median 2.35 ms), while 12.2% of samples follow a gap exceeding 15 ms, three times the nominal period, the longest being 41.3 ms. This is a property of the PSDK transport path rather than of the inertial sensor, and it should be taken into account by any odometry method that assumes uniformly sampled inertial input.

VI. CONCLUSION AND FUTURE WORK

This paper presented a cost-effective UAV mapping payload and a field validation of RKO-LIO driven by the drone's built-in IMU. On the hardware side, relying entirely on the drone's built-in IMU, nominally streaming at 200 Hz, via the DJI E-Port and designing efficient UART and DC-DC power circuits removes the need for an additional external IMU and simplifies payload integration. On the software side, practical results across a dynamic 458.49-meter outdoor UAV flight trajectory characterized the behavior of the RKO-LIO algorithm on an aerial platform. Its default 0.5 m association gate proved conservative for the sampling geometry of a 16-beam LiDAR at altitude; relaxing it reduced the final landing-point position error from 1.58 m to 0.31 m, i.e. 0.067% of the trajectory. Structural accuracy on the facade segments remained at an average relative error of 1.57% (maximum 2.47%). Post-flight analysis of the recorded timestamps also showed that latency spikes caused the onboard run to skip 41.3% of the LiDAR

scans. This did not affect the primary objective of the study: the onboard map remained accurate, as the facade measurements above show, and clearly resolved the surrounding campus buildings. The payload ran within a modest resource envelope, averaging 25.79% CPU and 1.99 GB RAM (peak 2.01 GB), which supports the practical feasibility of the integration.

Limitations. These results come from a single flight, and no continuous reference trajectory was available (the aircraft RTK solution was not logged, and the site had no motion-capture coverage). Absolute trajectory error and relative pose error [12] therefore cannot be computed; we report the final position error against a physically measured landing point, with an uncertainty of approximately ± 0.15 m, and do not label it ATE. The ordering of the tested settings is robust, but this single reference point does not resolve the optimal gate value. The odometry pipeline consumes no satellite positioning: GNSS was used solely by the DJI flight controller for stabilization and waypoint holding, and no GNSS measurement enters the mapping. The approach is therefore GPS-denied by construction, although the present flight took place at an outdoor site where GNSS happened to be available, so this case is addressed by design rather than demonstrated under actual signal deprivation.

Future work aims to utilize the real-time map produced within the Jetson unit to integrate advanced environmental analysis algorithms, transforming the drone into a remote-controlled mapping platform capable of onboard AI-driven data processing without satellite positioning. Recent learning-based navigation methods for dynamic environments may provide a future direction for integrating autonomous navigation capabilities with the proposed UAV mapping system [8].

ACKNOWLEDGMENT

This research was funded by The Scientific and Technological Research Council of Türkiye, TÜBİTAK, grant number 123E406 and was funded by FIRAT University Scientific Research Projects Unit, FUBAP, grant number MF.24.80 and MF.25.155.

REFERENCES

[1] Y. Wu, D. Liu, J. Ma, and X. He, "UAV Localization in GPS-Denied Environments: A Review," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, doi: 10.1109/TSMC.2026.3708505.

[2] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 5135-5142, doi: 10.1109/IROS45743.2020.9341176.

[3] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, vol. 2, pp. 1-9, 2014, doi: 10.15607/RSS.2014.X.009.

[4] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," in *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029-1036, Feb. 2023, doi: 10.1109/LRA.2023.3236571.

[5] M. V. R. Malladi, T. Guadagnino, L. Lobefaro, and C. Stachniss, "Robust Approach for LiDAR-Inertial Odometry Without Sensor-Specific Modeling," in *IEEE Robotics and Automation Letters*, vol. 11, no. 6, pp. 7420-7427, June 2026, doi: 10.1109/LRA.2026.3685966.

[6] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast Direct LiDAR-Inertial Odometry," in *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053-2073, Aug. 2022, doi: 10.1109/TRO.2022.3141876.

[7] M. El Muhammed, A. M. N. S. Alqadhi, M. A. M. S. Bajhaw, and A. Ucar, "MSR-SLAM: A Multi-Stage Map-Refinement Framework for Reliable LiDAR-Inertial SLAM in GPS-Denied Environments," *Applied Sciences*, 2026.

[8] A. M. N. S. Alqadhi, M. El Muhammed, M. A. M. S. Bajhaw, and A. Ucar, "PV-STAM: Velocity-Aware Attention for Mapless Deep Reinforcement Learning Navigation in Dynamic Environments," *Applied Sciences*, vol. 16, no. 18, art. no. 9083, 2026, doi: 10.3390/app16189083.

[9] A. M. N. S. Alqadhi, M. A. M. S. Bajhaw, M. El Muhammed, and A. Ucar, "Drift-Resilient Indoor 3D Mapping in GPS-Denied Environments Using RKO-LIO with Velodyne-Pixhawk Integration in ROS 2," 2026 8th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (ICHORA), Ankara, Türkiye, 2026, pp. 1-5, doi: 10.1109/ICHORA69329.2026.11537209.

[10] S. Macenski et al., "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, art. no. eabm6074, 2022, doi: 10.1126/scirobotics.abm6074.

[11] K. Museth, "VDB: High-Resolution Sparse Volumes with Dynamic Topology," *ACM Transactions on Graphics*, vol. 32, no. 3, art. no. 27, 2013, doi: 10.1145/2487228.2487235.

[12] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 2012, pp. 573-580, doi: 10.1109/IROS.2012.6385773.