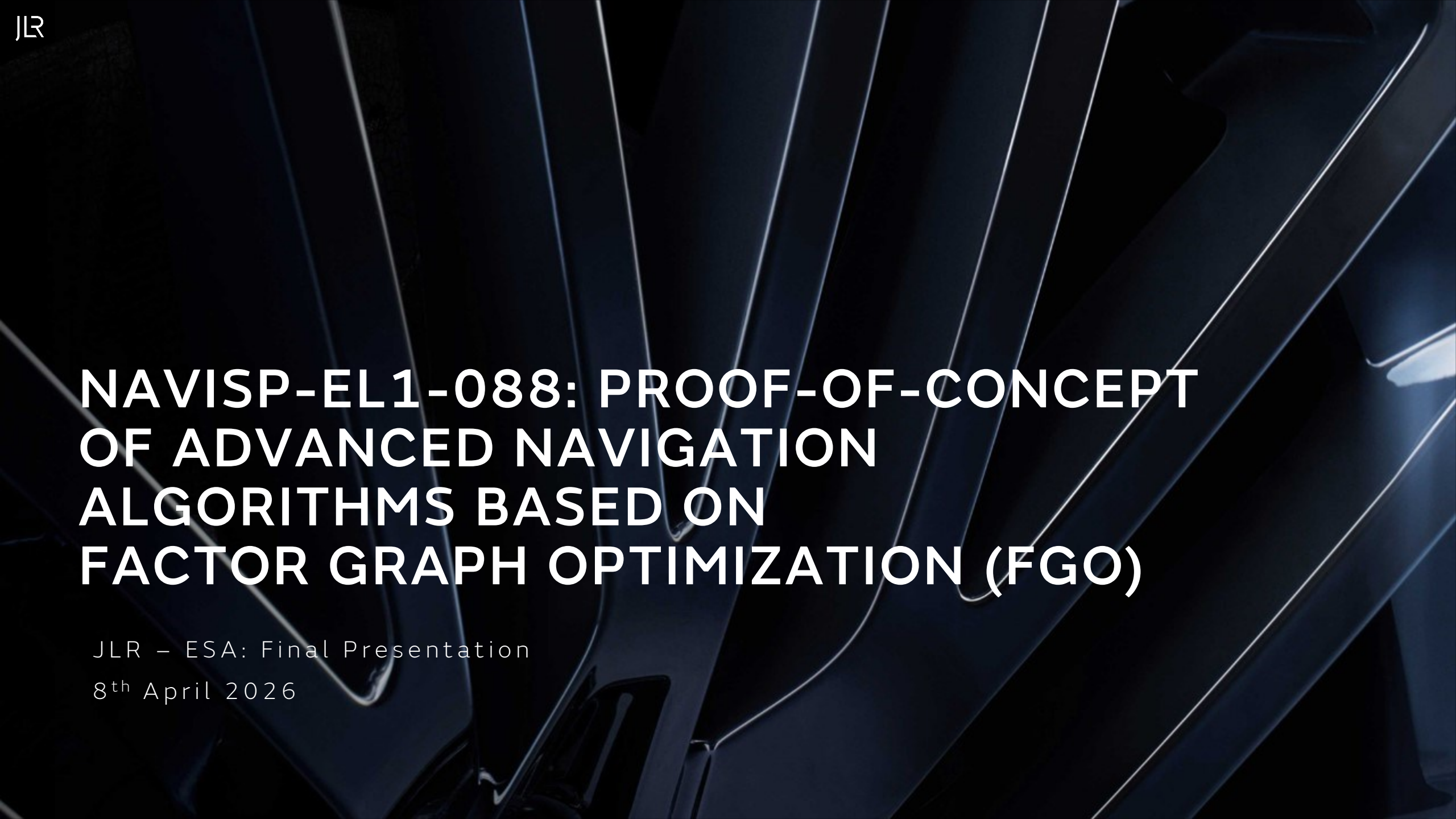


NAVISP-EL1-088: PROOF-OF-CONCEPT OF ADVANCED NAVIGATION ALGORITHMS BASED ON FACTOR GRAPH OPTIMIZATION (FGO)

JLR – ESA: Final Presentation

8th April 2026

Overview

Introduction to the Project

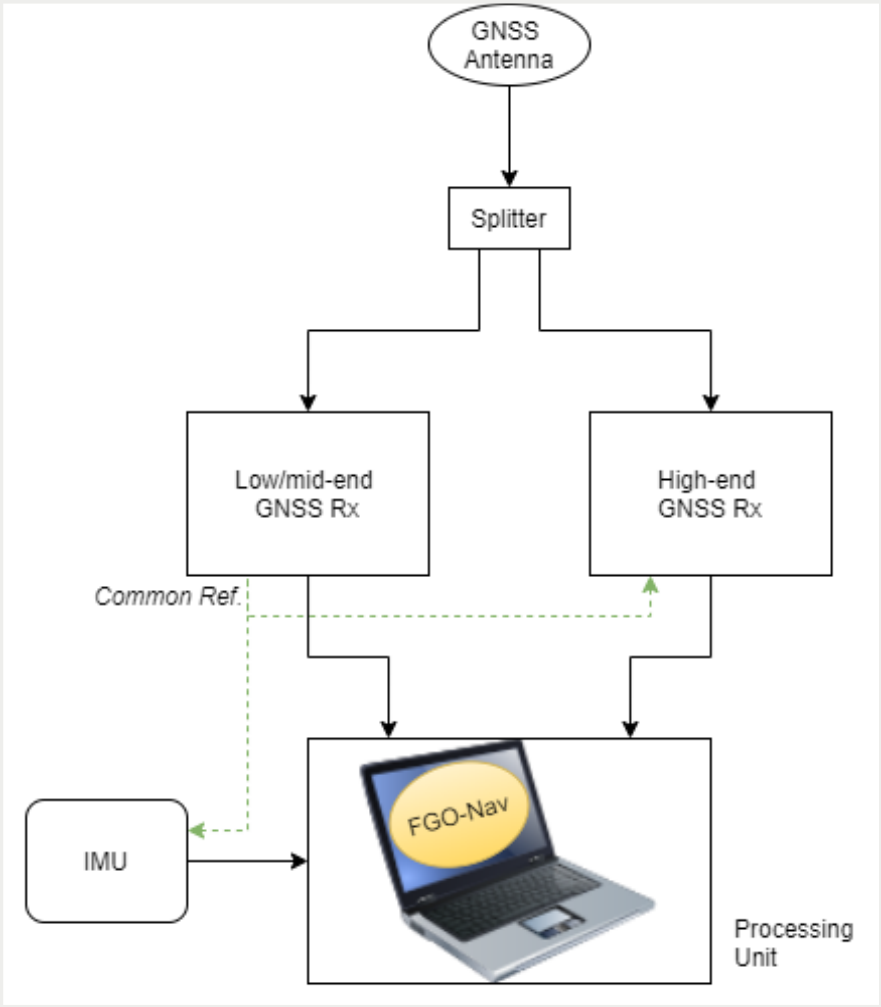
Vehicle Localisation in GNSS-degraded Environments

- Self-localisation and positioning are foundational to safe autonomous navigation, influencing path planning, decision-making, and vehicle control.
- Urban environments (“urban canyons”) present major challenges for localisation due to signal blockage from tall buildings, multipath reflections and degraded GNSS quality, frequent loss of satellite visibility, etc.
- Erroneous or unreliable GNSS measurements can lead to navigation errors, safety risks, and degraded performance of Advanced Driver Assistance Systems (ADAS).
- There is a growing need for robust, next-generation localisation algorithms capable of maintaining accuracy even in GNSS-challenged conditions.

JLR is evolving ADAS systems requiring increased accuracy and robustness for localisation

FGO-Nav Unit Design

- Fully engineered **FGO-Nav Unit** implementing a complete factor-graph-based navigation pipeline.
- Modular multi-sensor architecture supporting GNSS, IMU and extendable to LiDAR/Camera/Radar.
- Integrated **Sensor Abstraction Layer (SAL)** enabling hardware-agnostic sensor integration with reproducible offline replay environment to mimic real-time behaviour.
- Embedded **KPI computation suite** for performance analysis and validation.
- **User-friendly GUI** for configuration, visualisation, and switching between estimation modes (FGO/ WLS / EKF).
- Validated through multi-environment real-world datasets, ensuring robustness in GNSS-challenged conditions.
- Delivered as a scalable, configurable platform ready for automotive, robotics, and ESA research workflows.

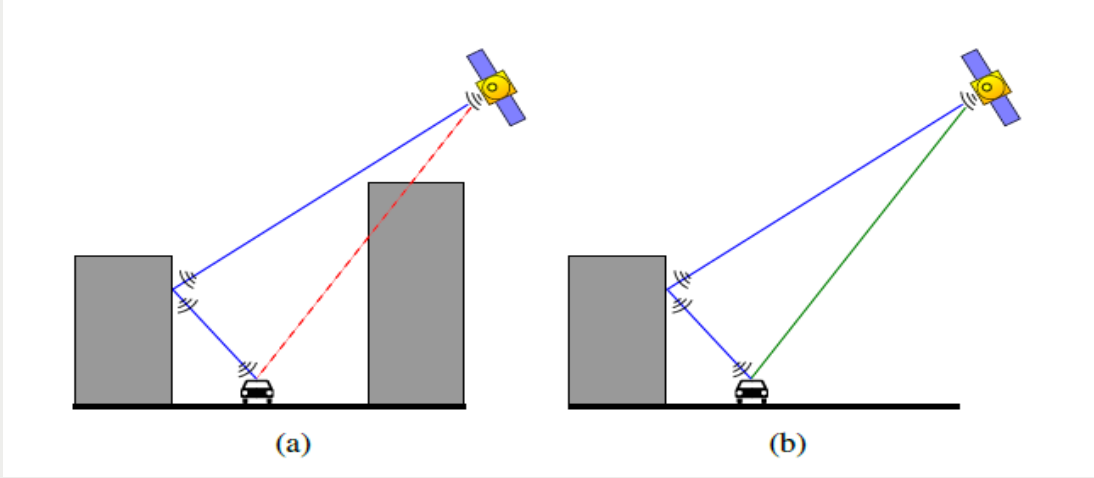


JLR fully addressed the call requirements


Use cases Trade-off & Selection

Challenges in Vehicle Localisation

- Weak or multipath GNSS signals in dense urban environments
- Sensor outliers caused by reflections, occlusions, and dynamic objects
- GNSS-denied areas such as tunnels, underground parking, and factory zones
- Positional drift and missing data frames during GNSS outages
- Traditional algorithms (WLS, EKF) struggle with delayed or occluded signals
- Sensor fusion complexity across GNSS, IMU, LiDAR, and cameras
- Need for robustness against noise and rapidly changing environments



The non-line-of-sight (a) and the multipath (b) problem are two common challenges for urban satellite-based positioning.

Image source: https://ieeexplore.ieee.org/document/6629480

GNSS dropouts and sensor noise in real-world environments require robust, redundant perception and fusion for safe autonomous driving

Challenges JLR addressed by designing FGO-Nav Unit

GNSS-Denied Environments

- Reliable positioning when GNSS signals are unavailable or degraded
- Critical for autonomous navigation in tunnels, parking structures, and dense industrial zones

Urban Canyons

- Robust against multipath, occlusions, and sensor outliers
- Outperforms EKF/WLS with smoother trajectories and higher accuracy

Multi-Sensor Fusion

- Combines GNSS, LiDAR, cameras, IMUs for redundancy and robustness
- Handles delayed/asynchronous data seamlessly
- Superior accuracy in complex, nonlinear systems compared to filtering methods

Navigation

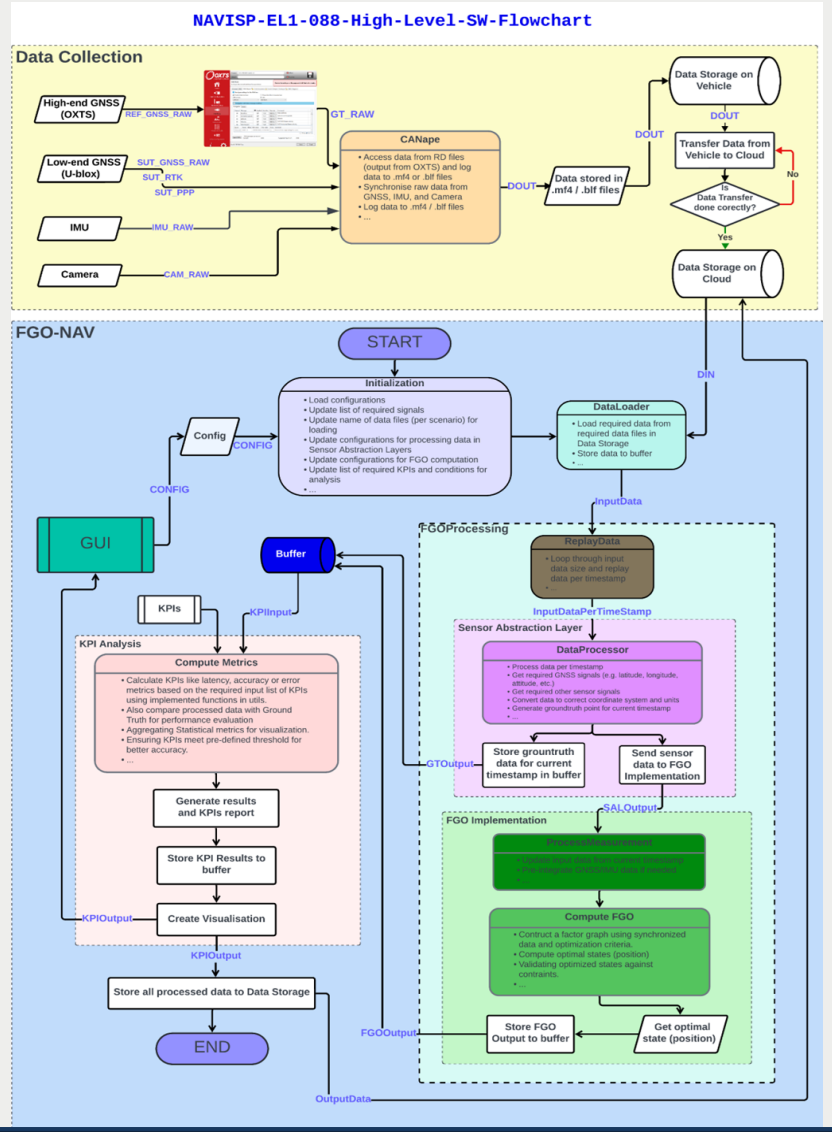
- Optimizes vehicle trajectory using temporal and spatial relationships
- Maintains accuracy during GNSS outages and sensor degradation
- Proven to outperform EKF in real-time navigation scenarios

JLR development of FGO solution aimed to test its application in core challenges facing the automotive market


Software Design

Software Development

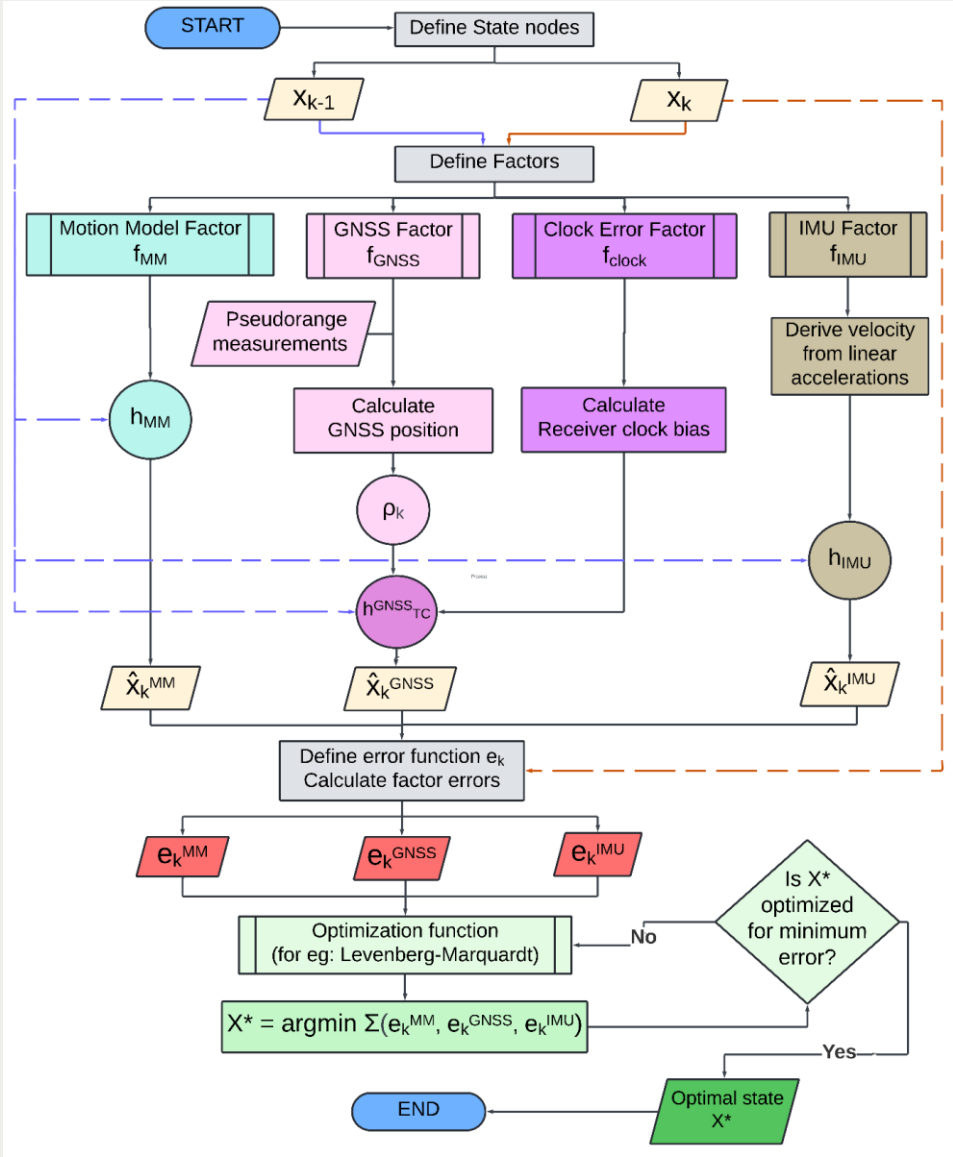
- The Software Design was developed in a containerized C++ environment that uses Docker to ensure platform-independent development and deployment.
- It is hosted on ESF Gitlab using CI/CD pipelines for static and dynamic analysis.
- Software Development lifecycle best practices were followed in the design of the algorithms.



Software flow charts were used to detail the design of FGO-Nav Unit and pseudocode was developed to ensure compliance to requirements

Factor Graph Optimization (FGO)

- **Step 1**: Load latest GNSS and IMU measurements from their respective buffers.
- **Step 2:** Construct a Factor Graph that includes state (position, velocity) nodes with relevant factors (GNSS Factor, IMU Factor, Motion Model Factor, etc.) at predefined time steps.
- **Step 3**: If more than one node exists in the optimization window, run the selected optimizer (**Gauss-Newton or Levenberg-Marquardt**) to solve the nonlinear least-squares problem by minimizing the sum of squared residuals from all factors associated with nodes in the window.
  It then iteratively updates the state estimates until convergence is reached, or the maximum number of iterations is exceeded.
- **Step 4**: Keep track of all updated states in state history. Repeat from Step 1.



Factor Graph Optimization finds the most probable configuration of variables that best satisfies all constraints


Test Campaign

Test Strategy Overview

The overall test strategy is a **comprehensive, phased, and data-driven approach** for robust system validation:

Phase 1: Data Collection Campaign

- In-vehicle testing in suburban, light urban, and deep urban environments
- Collect raw GNSS and IMU data using standard hardware
- Enables reuse of datasets for consistent test execution

Phase 2: FGO-Nav Unit Testing

- Lab-based replay of Phase 1 datasets
- Deterministic and repeatable testing environment
- Isolates algorithm performance from external variables

Phase 3: Defect Resolution & Regression Testing

- Bug fixing and re-validation of affected components
- Ensures system stability and reliability

Performance Evaluation

- Based on KPIs: position accuracy (CDF error), latency, computational load, energy consumption

Benchmarking

- Compare FGO-Nav against COTS GNSS receivers and open datasets (e.g., UrbanNav) - Assess its performance against conventional localisation algorithms – WLS and EKF

The test strategy uses a comprehensive, phased, and data-driven approach to validate system performance

Test Scenarios

Three different test scenarios/environments:

- **Suburban:** low congestion, varying traffic and weather conditions
- **Light urban:** areas with moderate building density and traffic, signal reflections from tall buildings and traffic variations
- **Deep urban:** dense high-rise structures, frequent signal obstructions, and heavy traffic.

Test Strategy:

- unit tests on individual components
- integration tests using KPIs
- stress tests under high data loads.

Test Execution:

- ensure data integrity by logging results to portable storage for post-analysis
- log report capturing KPIs like MSE, CDF error, latency, and energy consumption
- Plot graphs and create documentations

Table 3 System Test Campaign

S. No	Test scenario	Scope	Testbed Configuration Hardware	Testbed Configuration Software	Test strategy	Test Execution	Test Deliverable
1.	Suburban Environment	• The test shall include validating the precision of data fusion between GNSS and IMU in a suburban environment. [REQ-SOW-06] • Validate and assess the stability of the system under varying signal conditions typical of suburban areas. [Ex. Vehicle speed, impact of weather condition fog/rain]	• Processor: Intel i7 or equivalent • GNSS antenna – Novatel GNSS-804 • GNSS antenna splitter 4-way • Low-end-GNSS – CAM-M8C-0 and/or ZED F9P • High-end GNSS – RT3000 v4 IMU – OxTS • IMU – Mti-1/ADIS16460 • SSD EVO 2.5, 2TB SSD • Logging PC VP7571-vector • CAN interface – VN1670 Box with cables • 8 port USB Hub – S79-USBHUB-3P • Camera with mount – 0675-001	• FGO custom implementation • OxTS software • Drivers for GNSS and IMU • User selects FGO config – History length • User selects sensor config – GNSS high end, GNSS low end, IMU, Camera – Single/dual frequency [REQ-SOW-08] • Verify import export and modify functionality in GUI config • User selects SAL config – Data format and conversion	Unit test: Validate individual components (GNSS, IMU, FGO algorithm) separately. Verify FGO_Nav system response when no GNSS or no IMU or both not present – Reliability of the system Integration Test: Test predicted position by measuring KPI metrics [attach metric details] Stress Test: Evaluate system performance under high data load and varying suburban conditions.	• Verify the hardware and software environment. • Conduct tests in a suburban area, capturing data under different conditions (e.g., near buildings, open spaces). • Visualise GUI and observe CDF error plots, reference and computed position, computational burden of FGO-Nav. [REQ-SOW-11] • Analyse the results to identify any issues or areas for improvement. • Write the processed data to portable disk space to ensure data integrity and ease of access and download the data from the portable disk space for further analysis.	• Detailed report including test scenarios, results, KPI like MSE, CDF error, latency, responsiveness, energy consumption and plots. • Documentation of any issues found during testing.
2.	Light Urban Environment	• Test Navigation precision accuracy in areas with moderate building and traffic density. • Evaluate FGO-Nav system	Testbed hardware same as suburban case	Testbed software same as suburban case	Unit test: Validate individual components (GNSS, IMU, FGO algorithm) separately. Verify FGO_Nav system response when no GNSS or no	• Conduct tests in light urban area, capturing data under medium level traffic and tall buildings. • Visualise GUI and observe CDF error plots, MSE reference and computed	• Detailed report including test scenarios, results, KPI like MSE, CDF error,

The test scenarios were selected to address challenging environments for GNSS transmission

Data Collection Campaign Overview

Area - Date	Route ID	Google Map	Duration	Route Description	Weather
Birmingham_08Oct25	R09	https://maps.app.goo.gl/xv8TQNCPbe1PXn1MA	30 min 2 sec	Urban outer city driving, light traffic	Dry, Clear
	R10	https://maps.app.goo.gl/BnCwrEbTzRQEdATV7	27 min 41 sec	Rural village drive A-Roads, little traffic	Dry, Clear
	R11	https://maps.app.goo.gl/DyM1L2oXs8Vc2X9B9	1 hr 1 min 46 sec	Motorway route, clear run	Dry, Clear
London_03Oct25	R12a	https://maps.app.goo.gl/UG5xAsQHwzb4417j9	49 min 20 sec	Busy motorway, average traffic	Rainy
	R12b				
	R13	https://maps.app.goo.gl/FrgWFZNRZV4rnUKQ9	1 hr 40 min 43 sec	Peak busy London Street driving	Rainy
	R14	https://maps.app.goo.gl/eWex32dwap5oRegb9	42 min 37 sec	A-road route, above average traffic	Rainy
Alton Tower_09Oct25	R16	https://maps.app.goo.gl/gXqMgForHzctCw349	1 hr 55 min 56 sec	Motorway route, clear run	Dry, Cloudy
	R17	https://maps.app.goo.gl/rhPqELksohUsjA8f8	1 hr 18 min 46 sec	A-road route, light tree cover, little traffic	Dry, Cloudy
	R18	https://maps.app.goo.gl/mmQsMscGQGCVuZYk9	57 min 30 sec	Suburban outer city driving, some traffic	Dry, Clear
	R19a	https://maps.app.goo.gl/8frSqyrWQVgzhYkr8	35 min 31 sec	First half: city drive with tunnels and high-rise buildings; second half: motorway, light traffic	Dry, Clear

The Recorded over 24 hours data in light urban, suburban, and deep urban areas

Test Campaign Summary

Activity	Description
Total Data Logged	24+ hours of fully synchronised GNSS/INS recordings across diverse UK routes.
Environment Coverage	Suburban, light-urban, motorway, deep-urban, tunnels, NLOS zones, high-rise corridors, and tree-shadowed segments.
Algorithms Evaluated	LC-FGO, TC-FGO, FGO-RTK, PPP-FGO, WLS, EKF, and GNSS/INS RTK Receiver.
Dataset Types	Synthetic datasets, public deep-urban datasets (UrbanNav), and JLR real-world multi-city datasets.
KPI Metrics Computed	RMSE, MAE, Max Error, latency and Computational usage.
Hardware & Timestamp Validation	Verified GNSS/INS hardware stability, PTP-based time synchronisation, and sensor alignment.
Data Continuity Assurance	Identified and re-driven routes to recover missing GNSS observables, ensuring ESA-aligned dataset completeness.
End-to-End Integrity Checks	Confirmed timestamp correctness, absence of data gaps, and alignment between reference (OxTS) and raw GNSS/IMU streams.

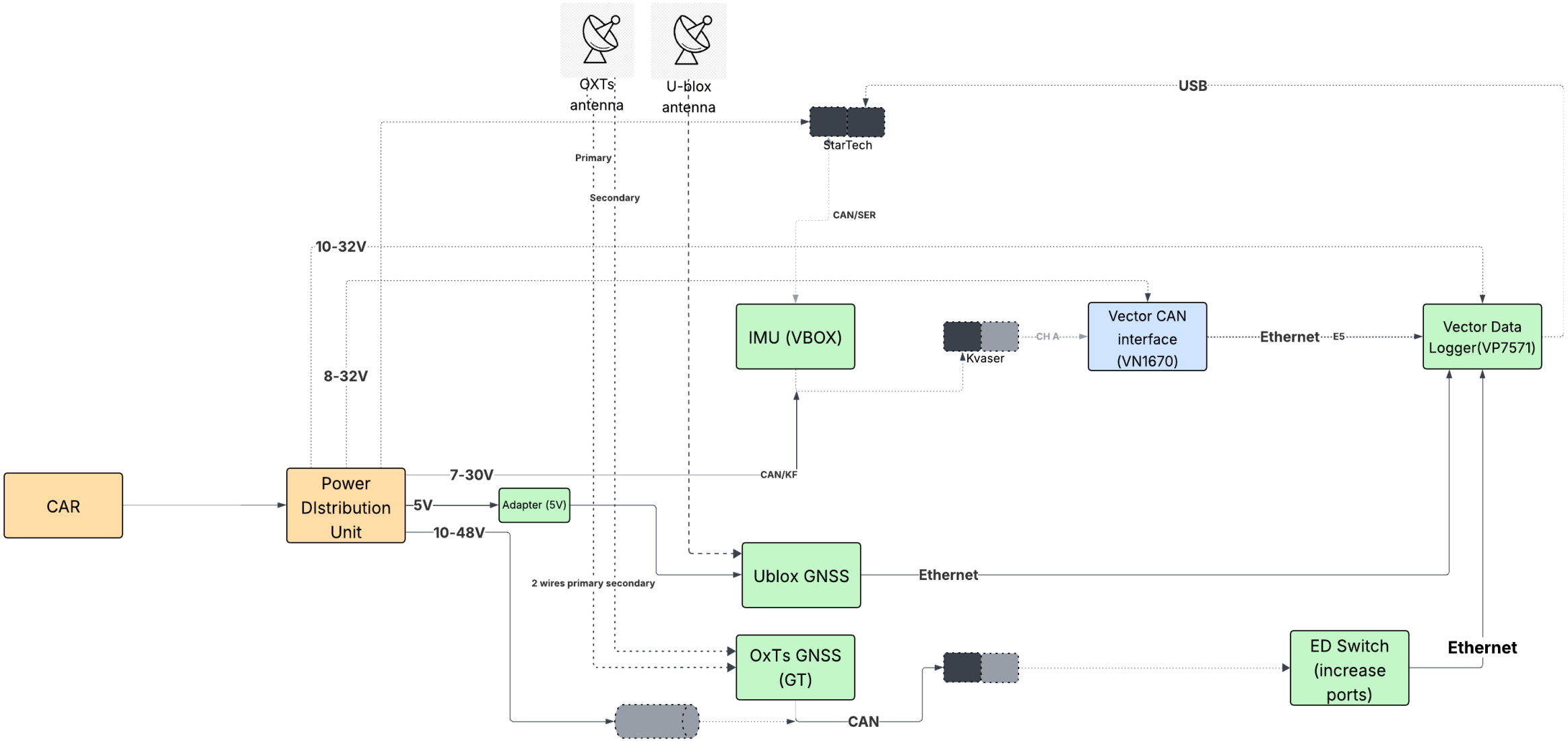

Hardware Testbed Design

Selected Hardware (HW)

Device	Model
GNSS Antenna (x2)	Novatel VEXXIS GNSS-804 GPS Antenna (roof mounted)
Splitter (x2)	GPS Antenna Splitter 4-way
Low-end GNSS	CAM-M8C-0 and/or ZED F9P
High-end GNSS	RT3000 v4 IMU
IMU	IMU05
SSD	870 EVO 2.5, 2TB SSD
Logging PC	VP7571
PC screen	1503I
Antenna	Wifi modem antenna
CAN Interface	VN1670 Box with cables
8 port USB Hub	S79-USBHUB-3P
Isolator switch	6006

Hardware selected to assess FGO-Nav Solution applicability to our car lines

HW Architecture




GUI Design & Demo

Graphical User Interface (GUI)

Technology used:

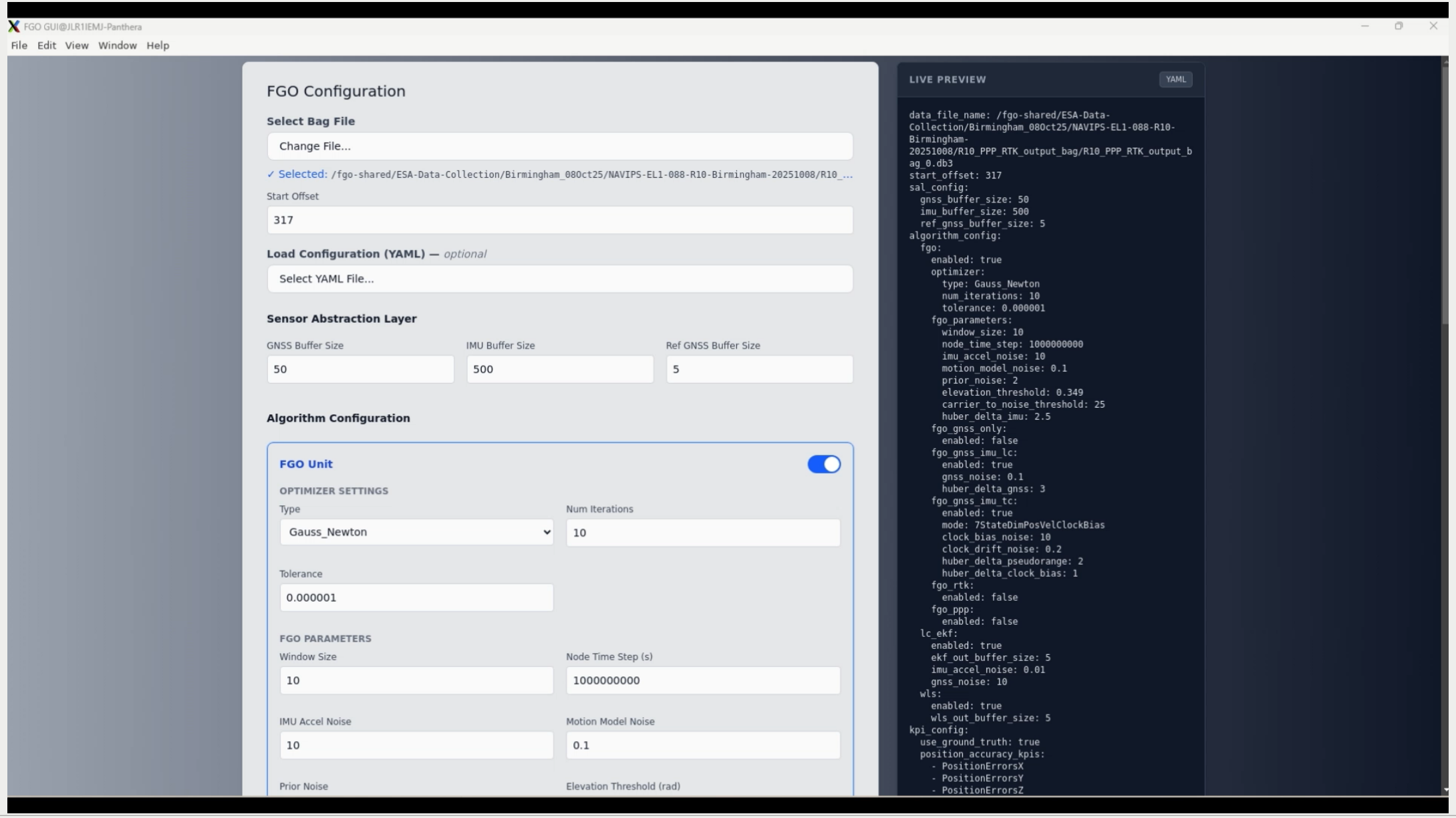
- **Electron**: Electron is a framework that lets you build cross-platform desktop apps using web technologies like HTML, CSS, and JavaScript.
- **React**: React is a JavaScript library for building fast, interactive user interfaces with reusable components.

Usage:

- Provides a user interface to select recorded datasets and configure processing settings before running FGO.
- Responsible for starting FGO processing and visualising KPI results for performance evaluation.
- Enables users to tune parameters across multiple FGO methods and compare results against benchmarking techniques.
- Supports a configurable FGO engine covering:
  - GNSS-only
  - Loosely Couple GNSS/INS Integration (LC-FGO)
  - Tightly Couple GNSS/INS Integration (TC-FGO)
  - Real-Time Kinematic (RTK-FGO)
  - Precise Point Positioning (PPP-FGO)

GUI provides a framework for configuring and tuning parameters of multiple FGO methods

Demo


FGO GUI@JLR1IEMJ-Panthera
File Edit View Window Help
FGO Configuration
Select Bag File
Change File...
✓ Selected: /fgo-shared/ESA-Data-Collection/Birmingham_08Oct25/NAVIPS-EL1-088-R10-Birmingham-20251008/R10_...
Start Offset
317
Load Configuration (YAML) — optional
Select YAML File...
Sensor Abstraction Layer
GNSS Buffer Size
50
IMU Buffer Size
500
Ref GNSS Buffer Size
5
Algorithm Configuration
FGO Unit
OPTIMIZER SETTINGS
Type
Gauss_Newton
Num Iterations
10
Tolerance
0.000001
FGO PARAMETERS
Window Size
10
Node Time Step (s)
1000000000
IMU Accel Noise
10
Motion Model Noise
0.1
Prior Noise
Elevation Threshold (rad)
LIVE PREVIEW
YAML
data_file_name: /fgo-shared/ESA-Data-Collection/Birmingham_08Oct25/NAVIPS-EL1-088-R10-Birmingham-20251008/R10_PPP_RTK_output_bag/R10_PPP_RTK_output_bag_0.db3
start_offset: 317
sal_config:
gnss_buffer_size: 50
imu_buffer_size: 500
ref_gnss_buffer_size: 5
algorithm_config:
fgo:
enabled: true
optimizer:
type: Gauss_Newton
num_iterations: 10
tolerance: 0.000001
fgo_parameters:
window_size: 10
node_time_step: 1000000000
imu_accel_noise: 10
motion_model_noise: 0.1
prior_noise: 2
elevation_threshold: 0.349
carrier_to_noise_threshold: 25
huber_delta_imu: 2.5
fgo_gnss_only:
enabled: false
fgo_gnss_imu_lc:
enabled: true
gnss_noise: 0.1
huber_delta_gnss: 3
fgo_gnss_imu_tc:
enabled: true
mode: 7StateDimPosVelClockBias
clock_bias_noise: 10
clock_drift_noise: 0.2
huber_delta_pseudorange: 2
huber_delta_clock_bias: 1
fgo_rtk:
enabled: false
fgo_ppp:
enabled: false
lc_ekf:
enabled: true
ekf_out_buffer_size: 5
imu_accel_noise: 0.01
gnss_noise: 10
wls:
enabled: true
wls_out_buffer_size: 5
kpi_config:
use_ground_truth: true
position_accuracy_kpis:
- PositionErrorsX
- PositionErrorsY
- PositionErrorsZ


Results & Performance Evaluation

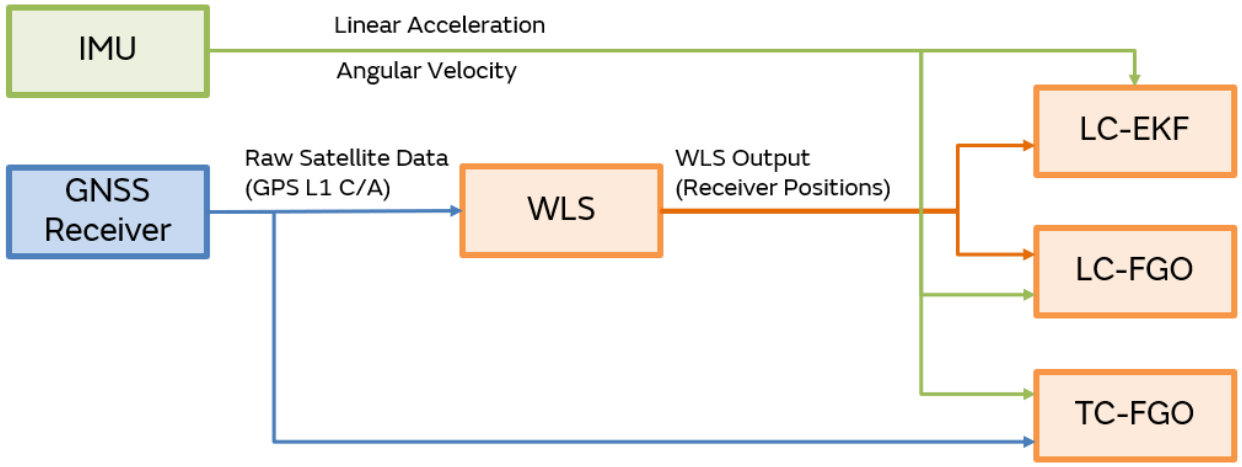
Software Test Approach

Benchmark Methods

- **Weighted Least Squares (WLS)**: Minimizes weighted sum of squared residuals between measured GPS L1 C/A pseudoranges and modelled ranges from estimated satellite positions.
- **Extended Kalman Filter (EKF)**: Uses WLS position estimates (from raw GNSS L1 C/A pseudoranges) and IMU data with Gaussian noise assumptions for process and measurement uncertainties.

Factor Graph Optimisation (FGO) Methods:

- Loosely Coupled FGO (LC-FGO)
  - Factors: GNSS, IMU, Motion Model, Prior
  - Input: Position estimates from WLS + IMU data
  - Output: 3D position & velocity per timestamp
- Tightly Coupled FGO (TC-FGO)
  - Factors: Pseudorange, Clock Bias, IMU, Motion Model, Prior
  - Input: Raw GNSS pseudoranges (L1 C/A) + IMU data
  - Output: 3D position, velocity & receiver clock bias per timestamp



Flowchart of input for Loosely Coupled and Tightly Coupled FGO Implementations

Test LC-FGO and TC-FGO against WLS and EKF for benchmarking

Light Urban Datasets

Table 1: KPIs results from testing synthetic data generated by MATLAB with standard deviation = 1m.

Methods	RMSE 2D [m]	RMSE 3D [m]	MAE 2D [m]	MAE 3D [m]	Max Error 2D [m]	Max Error 3D [m]
LC-FGO	0.26806	0.32020	0.15552	0.18207	1.63079	1.90515
TC-FGO	0.19966	0.21939	0.13053	0.14371	1.14770	1.23865
EKF	0.32455	0.46669	0.16239	0.21933	4.01531	5.54979
WLS	0.10725	0.11102	0.08644	0.09127	0.39445	0.39919

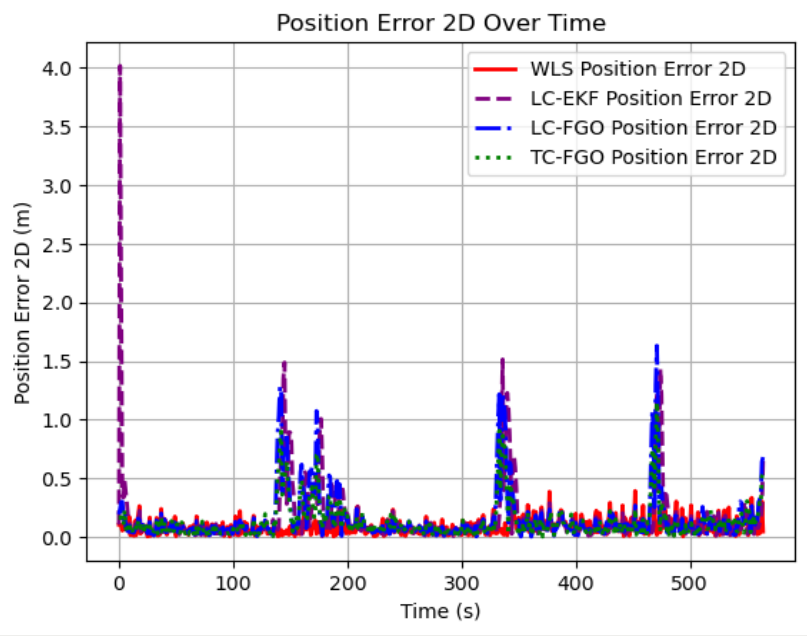
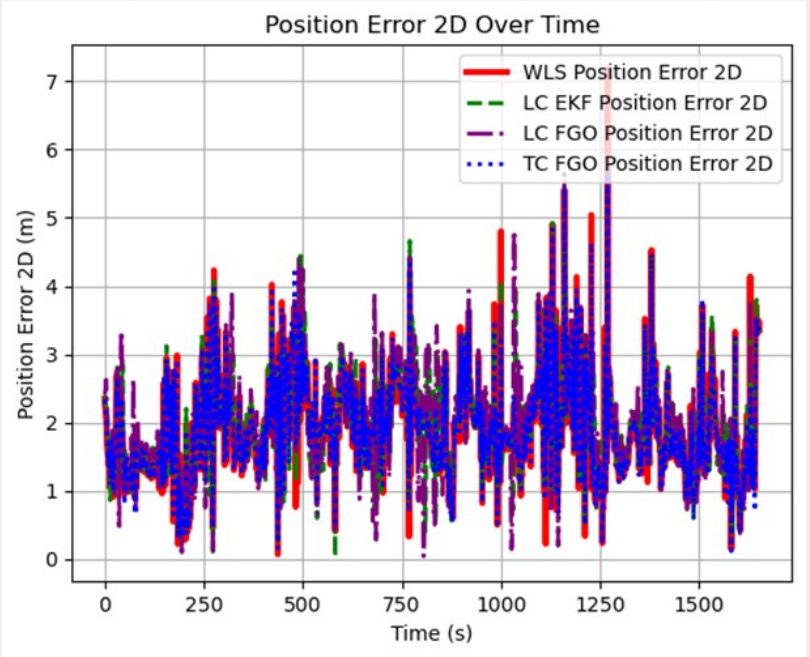


Table 2: KPI Results from testing Birmingham_08Oct25-R10 dataset.

KPIs in meters	WLS	LC-EKF	LC-FGO	TC-FGO
RMSE 2D	2.031375	2.113303	2.148283	2.015695
RMSE 3D	49.83129	49.81355	49.81142	49.65871
Max Error 2D	7.18135	5.665822	5.755536	5.877274
Max Error 3D	64.86579	64.25032	64.30326	63.43723
MAE 2D	1.909489	1.99103	2.015982	1.899582
MAE 3D	49.79253	49.77843	49.77688	49.62696



RMSE 2D of LC-FGO and TC-FGO for JLR collected Data for Birmingham Region < 2.1m

Suburban Datasets

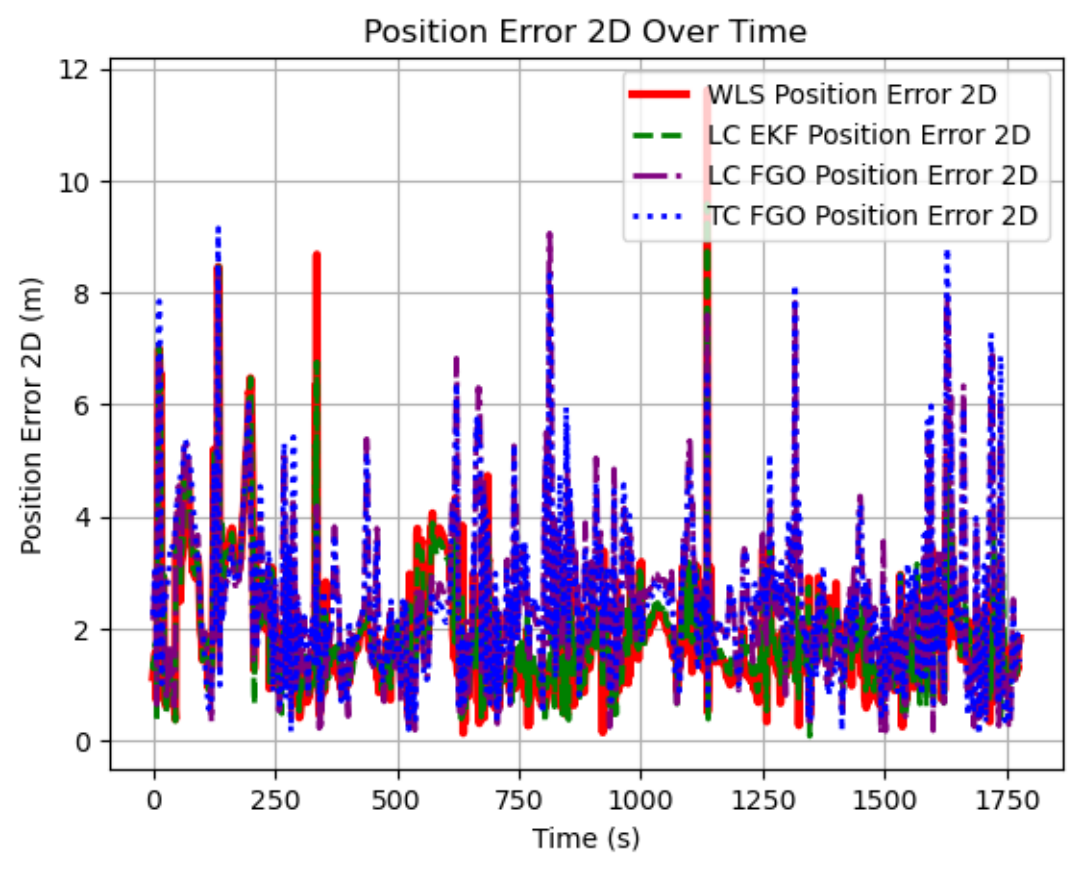


Table 3: KPI results of LC-FGO, TC-FGO, WLS, and LC-EKF, against reference GNSS, including RMSE, MAX, MAE values across 2D and 3D error components for Birmingham_08Oct25-R09 dataset.

KPIs in meters	WLS	LC-EKF	LC-FGO	TC-FGO
RMSE 2D	2.109777	2.125303	2.753646	2.836676
RMSE 3D	50.616040	50.630205	50.715595	50.058132
Max Error 2D	11.611375	9.589159	9.056732	9.145437
Max Error 3D	71.393319	66.844475	61.629978	68.746648
MAE 2D	1.849138	1.886781	2.428372	2.509678
MAE 3D	50.573740	50.589416	50.667919	49.976340

RMSE 2D of LC-FGO and TC-FGO for JLR collected Data is < 3m

Deep Urban Datasets

Table 4: KPI results of LC-FGO and EKF against reference GNSS, including RMSE values across 2D and 3D error components for UrbanNav dataset : UrbanNav-20210517-With-GNSS-Without-LiDAR-and-Camera.

KPIs in meters	LC-FGO	LC-EKF
RMSE X	5.65	6.99
RMSE Y	8.03	8.58
RMSE Z	6.14	7.53
RMSE 2D	9.82	11.07
RMSE 3D	11.59	13.39

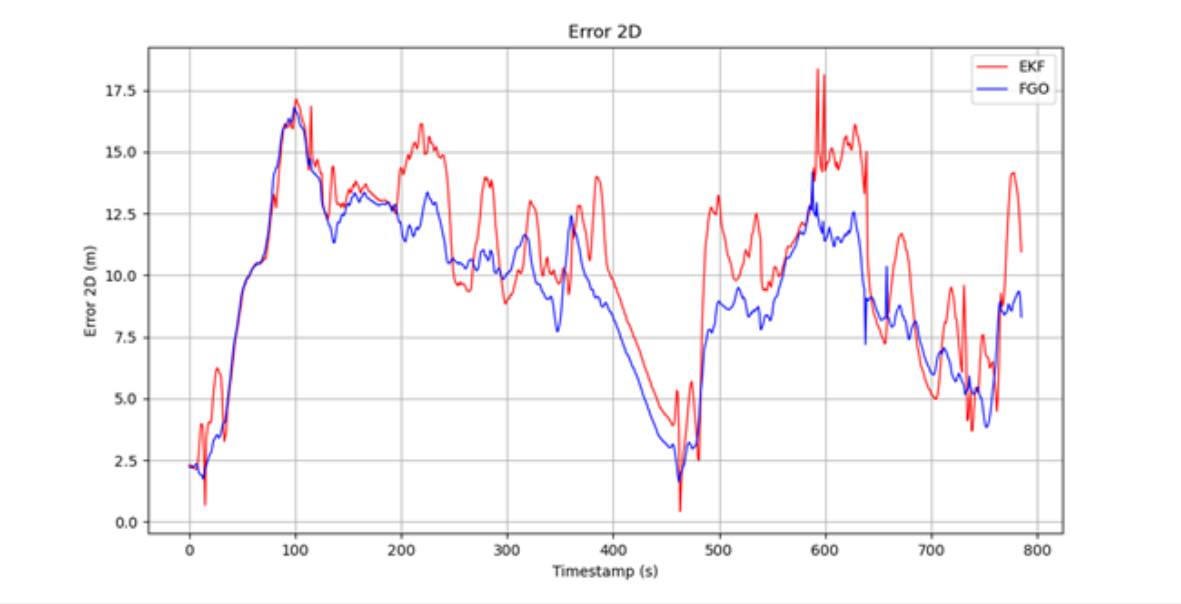
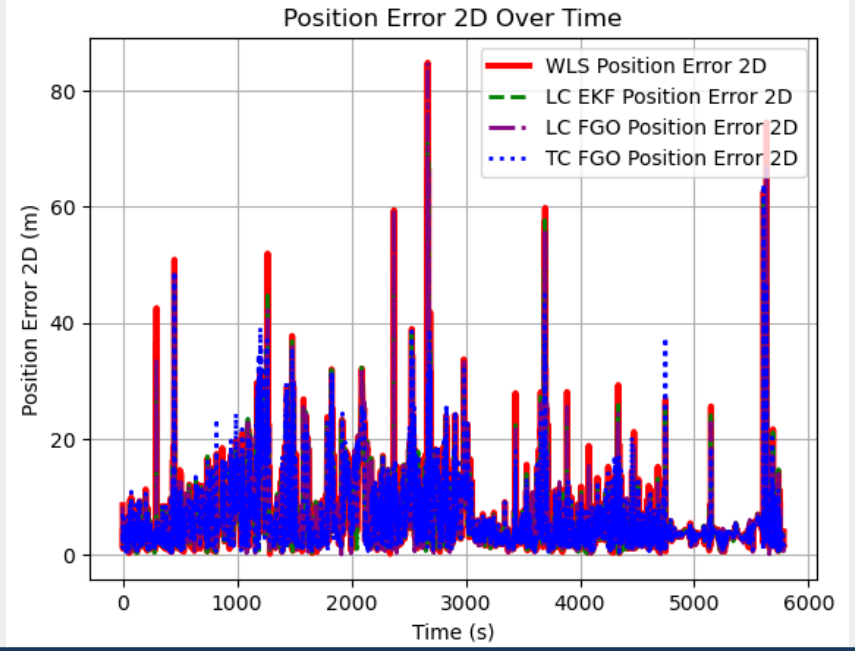


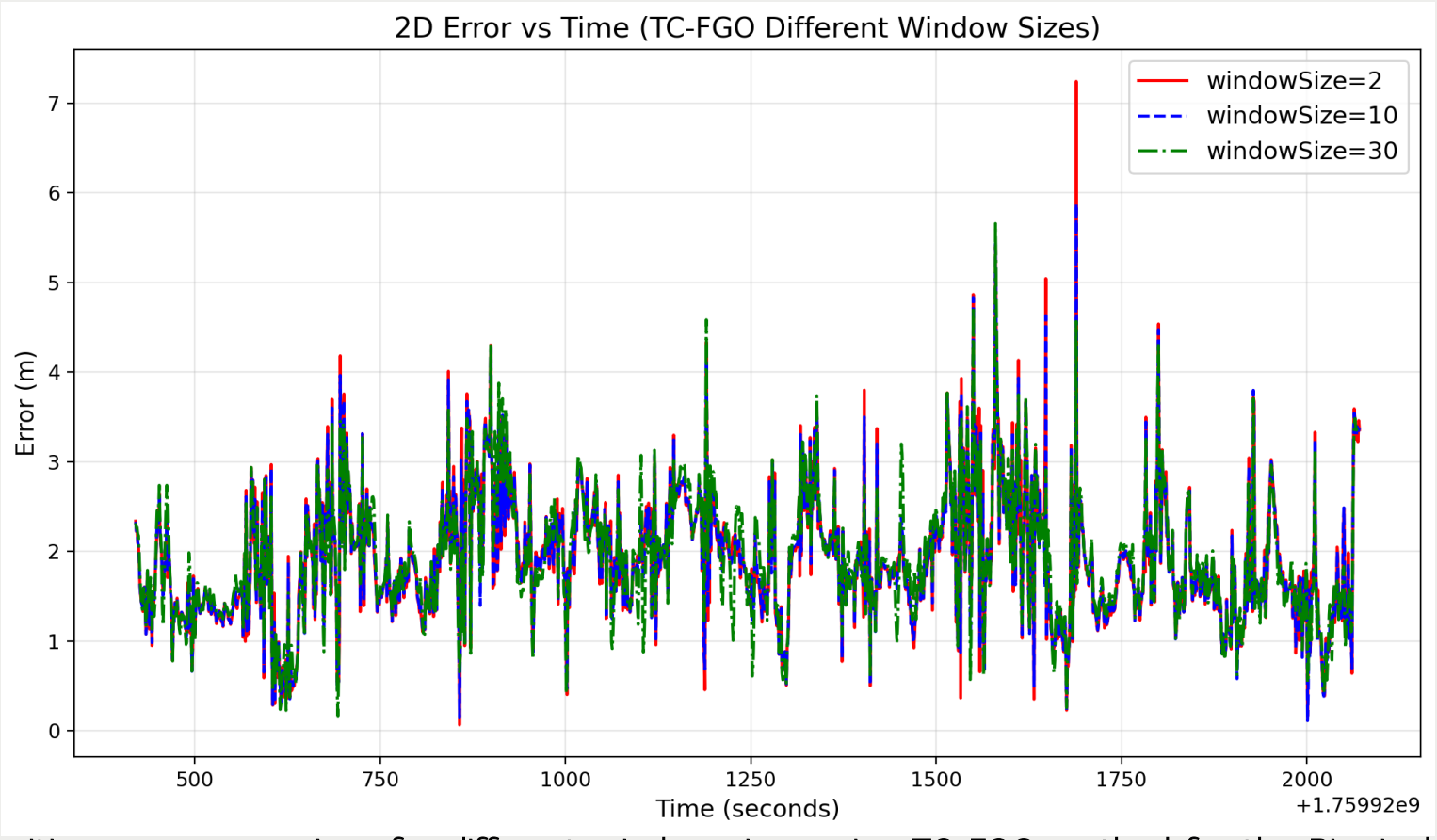
Table 5: KPI results of LC-FGO, TC-FGO, WLS, and LC-EKF, against reference GNSS, including RMSE values across 2D and 3D error components for London_03Oct25-R13 dataset.

KPIs in meters	WLS	LC-EKF	LC-FGO	TC-FGO
RMSE X	6.058059	5.921963	5.968623	5.714447
RMSE Y	6.790555	6.747896	6.754115	6.508593
RMSE Z	60.56854	60.47417	60.49212	59.3347
RMSE 2D	9.100094	8.977958	9.013464	8.661218
RMSE 3D	61.24835	61.13697	61.15995	59.96352



RMSE 2D of LC-FGO and TC-FGO for UrbanNav and JLR collected Data is < 10m

Window Size Sensitivity Test



2D position error comparison for different window sizes using TC-FGO method for the Birmingham_08Oct25-R10 dataset. The results show: windowSize = 2 → RMSE 2D = 2.022453 m; windowSize = 10 → RMSE 2D = 2.015695 m; windowSize = 30 → RMSE 2D = 2.041261 m.

A window size of 10 epochs provides a good balance between accuracy, robustness, and computational load

Software Performance Summary

Algorithm Comparison: WLS vs EKF vs FGO

Method	Operation	Advantages	Disadvantages
WLS	Minimizes weighted residuals using linear combinations	Simple, efficient for static problems, handles noise via weights	Limited adaptability, less effective in dynamic systems
EKF	Recursive state estimation with linearization	Real-time capable, modest computational needs, widely-used	Sensitive to poor initial conditions, limited non-linearity handling
FGO	Graph-based optimization using historical and current measurements	High accuracy, robust to outliers, flexible sensor integration	High computational cost, complex setup, less suited for real-time

Integration Strategy Comparison: LC vs TC

Type	Operation	Advantages	Disadvantages
Loosely Coupled (LC)	GNSS and INS operate independently; outputs fused at position level	Flexible, easier to implement, robust to full GNSS loss	Lower accuracy during outages, less synergy between sensors
Tightly Coupled (TC)	Raw GNSS data directly integrated with INS at measurement level	High accuracy, better performance in partial GNSS outages	Complex implementation, higher computational cost

FGO + Tightly Coupled GNSS/INS integration deliver best accuracy and robustness.

Conclusion

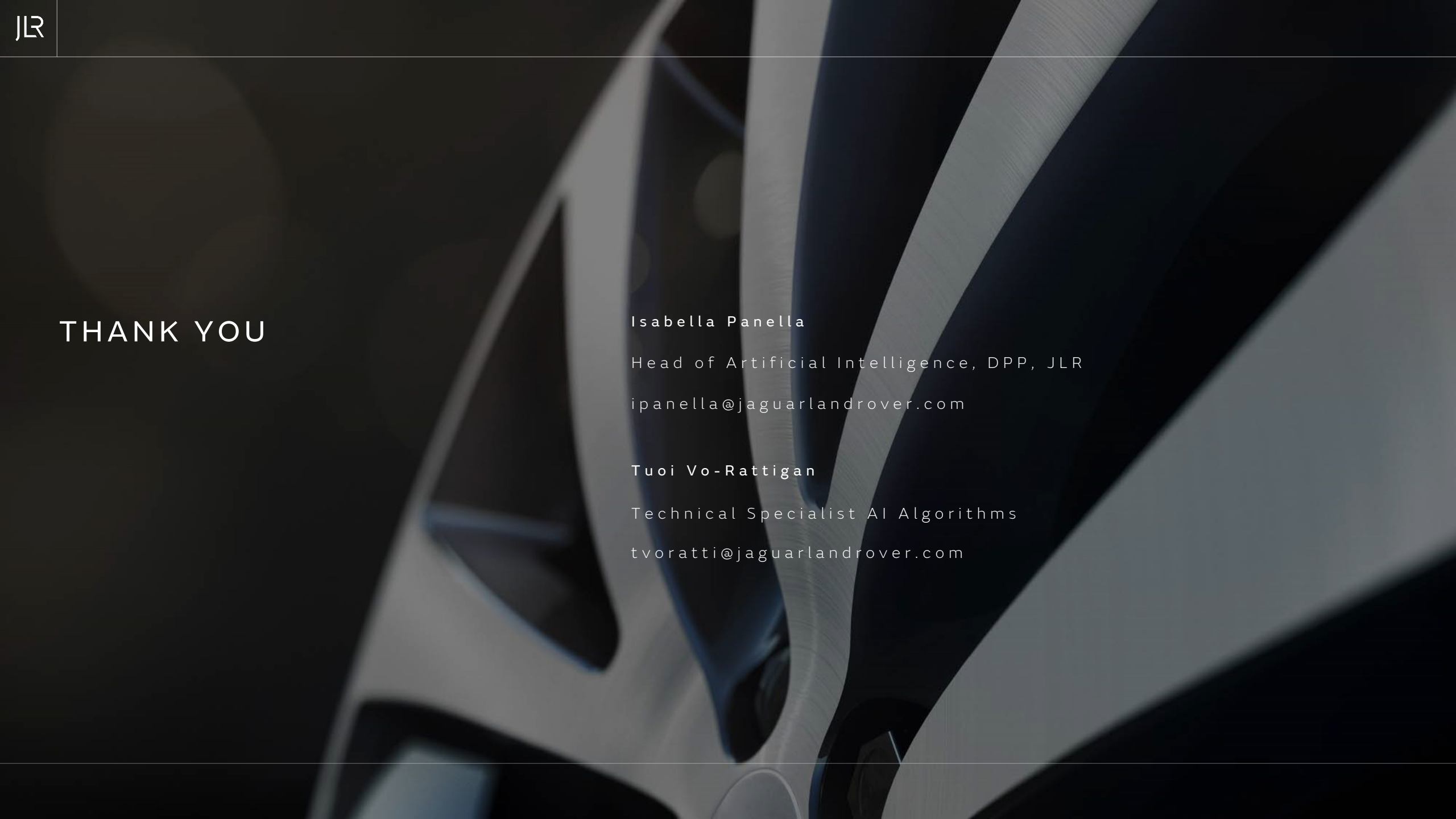
Conclusion

Key Outcome Area	Summary
Proof-of-Concept	Successful proof-of-concept of **FGO-Nav unit**
System Delivery	Complete modular architecture delivered (**Data Logger, SAL, FGO Engine, KPI, GUI**)
ESA Compliance	Full compliance with **ESA SoW requirements (REQ SOW 01–17)**
Data Collection	**24+ hours** of multi-environment, multi-sensor real-world data collected
Performance	Clear superiority of **FGO in GNSS-challenged urban conditions** vs. EKF and WLS
Real-Time Readiness	Real-time compatible, reproducible, and user-friendly system
Future Impact	Strong foundation for future **automotive, robotics, and localisation solutions**


Recommendation

Recommendation

Focus Area	Recommendation
Vertical Accuracy	Improve vertical accuracy using barometer, 3D maps, and dual-frequency corrections
Optimisation	Adaptive sliding-window management and improved noise modelling
GUI Improvements	Add real-time plotting
Dataset Expansion	Broaden dataset diversity (megacities, rural regions, adverse weather)
Sensor Fusion	Integrate Vehicle Odometry, LiDAR, Camera, and Radar for GNSS-outage resilience
Advanced Research	Explore hybrid **FGO-Kalman**, integrity-driven FGO, and cloud/map-assisted priors



THANK YOU

Isabella Panella

Head of Artificial Intelligence, DPP, JLR

ipanella@jaguarlandrover.com

Tuoi Vo-Rattigan

Technical Specialist AI Algorithms

tvoratti@jaguarlandrover.com