POLITECNICO
MILANO 1863

Engineering Challenge

Autonomous Vehicle Sensor Fusion

Team Secure Motion

May 27, 2026

Project Overview

Core Objectives

- Track opponent race car relative to ego vehicle
- Fuse LiDAR, radar, and camera inputs
- Deliver stable, real-time state estimates

The Dataset

- 120s of Yas Marina racing data
- 2,388 synchronized frames
- ROS2 SQLite3 (`.db3`) bag
- CDR serialized messages

Target Outputs: Range · Relative Heading · Absolute Velocity · Uncertainty

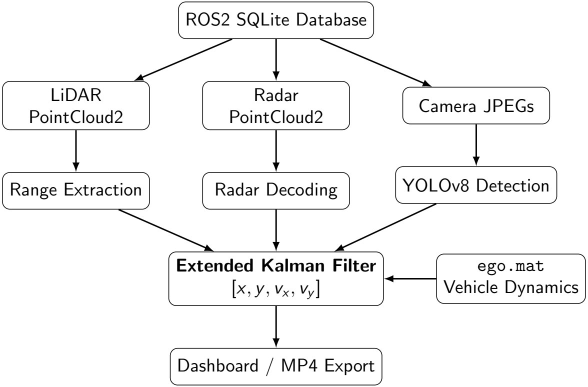
Sensor Suite Configuration

Sensor Type	Qty	Primary Role
Camera	7	Object classification & confirmation
LiDAR	3	Spatial geometry & precise range
Radar	4	Direct velocity & robust tracking

Active ROS2 Topics:

- `camera_[fl, fr, cl, cr, rl, rr, r]`
- `lidar_[front, right, left]`
- `radar_[front, back, right, left]`

Software Architecture



Core Pipeline Executables: `run_pipeline.py` · `ekf_track.py` · `dashboard.py`

Pipeline Execution Strategy

Phase 1: Offline Processing

- Ingest raw sensor logs $\rightarrow$ Cache extracted LiDAR frames
- Run baseline Linear KF and Extended Kalman Filter (EKF)
- Export structured results directly to CSV

Phase 2: Live Insights

- Real-time multi-sensor telemetry visualization
- Synchronized 4-panel monitoring dashboard
- Direct telemetry video render (MP4 export support)

Sensor Fusion Mapping

Sensor Source	Functional Contribution
Front LiDAR	Point cluster segmentation → Exact target geometry
Front Radar	Doppler radial velocity
Camera Array	YOLOv8 bounding boxes → Visual confirmation + Position tracking
`ego.mat`	Dynamic vehicle state → EKF kinematic correction
Side/Rear Sensors	360° situational awareness & clutter visualization

Key Algorithmic Steps

- **Temporal Alignment:** search nearest-frame matching anchored to camera timestamps.
- **LiDAR Range Tracking:** Decoded `PointCloud2` blobs, spatial bounding-box filters, and 20th percentile forward depth calculation.
- **Vision Inference:** Light YOLOv8n model deployed on CPU using a multi-stage confidence cascade.
- **Target Initialization:** Radar cross-section detections seed the local LiDAR point cluster search space.

Baseline: Linear Kalman Filter

State Space

$$x = \begin{bmatrix} r \\ \dot{r} \end{bmatrix}$$

Kinematics

$$r_{k+1} = r_k + \Delta t \cdot \dot{r}_k$$

Prediction

$$\hat{x}^- = A\hat{x}, \quad P^- = APA^T + Q$$

Correction

$$K = P^- H^T (HP^- H^T + R)^{-1}$$

$$\hat{x} = \hat{x}^- + K(z - H\hat{x}^-)$$

Core Engine: Extended Kalman Filter

Tracked State: $x = \begin{bmatrix} x_w & y_w & v_x & v_y \end{bmatrix}^T$

- **Asynchronous Updates:** Handles unique measurement models ($h(x)$) for LiDAR range, radar position, and radar radial velocity.
- **Ego-Pose Integration:** Incorporates true vehicle velocity, yaw rate, and absolute heading via trapezoidal dead reckoning.
- **Rich Outputs:** Generates 2D coordinates, true absolute target speed, heading vectors, and continuous covariance tracking.

Data Flow Pipeline

Processing Steps

1. Parse raw SQLite bag data
2. Synchronize multi-rate clocks
3. Track spatial LiDAR features
4. Log telemetry baselines to CSV
5. Decode radar Doppler returns
6. Compete ego vehicle kinematics
7. Execute EKF state tracking
8. Update dashboard UI

Output Artifacts (.csv)

- `all_frames_measurements`
- `all_frames_ekf`
- `all_frames_ekf_submission`

Tracking Performance

Evaluation Metric	EKF Engine
Range Profile	Accurate tracking ($\sim$X m)
Target Speed	True fused velocity
Relative Heading	Active tracking ($\pm$X$^\circ$)
Positional Variance	Stable ($\sim$0.3 m)
Velocity Variance	Stable ($\sim$0.2 m/s)

Real-Time Dashboard UI

- **Vision Pane:** Camera mosaic overlayed with active YOLO bounding boxes
- **Spatial Pane:** Overhead bird's-eye-view of LiDAR cluster sets
- **Radar Pane:** Multi-directional live radar scatter return map
- **HUD Overlay:** Critical telemetry strips and state confidence steps

Dashboard

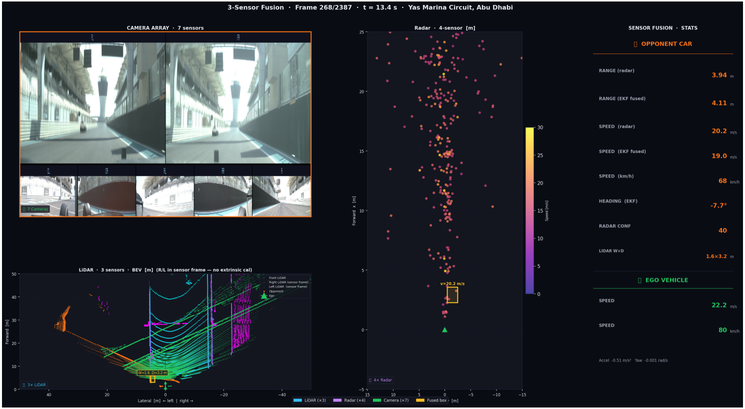


Figure: A snapshot of the dashboard with the LiDAR bird eye view, camera feed, radar scan, and the telemetry data.

LiDAR Scan

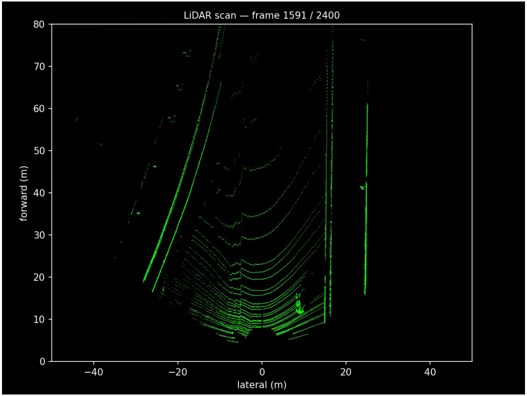


Figure: Bird Eye View from the front LiDAR sensor.

LiDAR overlay on Camera Feed

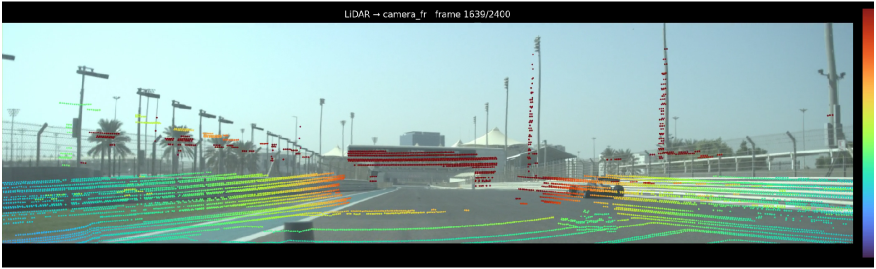


Figure: Overlay of the front LiDAR sensor view on the Center left camera feed.

Detection of Opponent

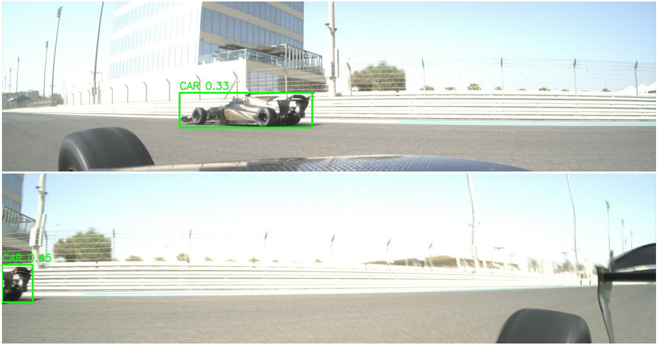


Figure: Detection of the Opponent from the Camera feeds.

Detection of Opponent - Abnormalities

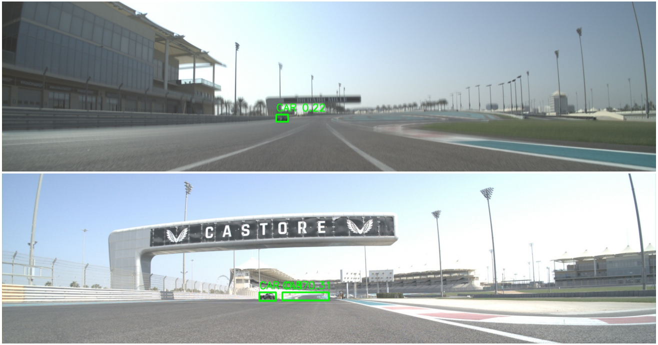


Figure: Abnormalities during the detection of the Opponent from the Camera feeds.

Tracing of opponents path

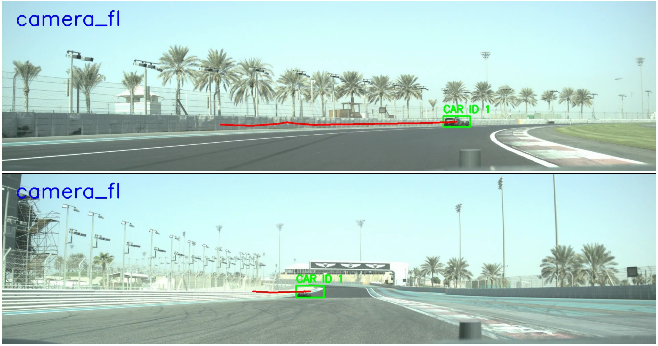


Figure: Tracking the path of the Opponent from the Camera feeds.

Edge Cases Resolved

Issue	Root Cause	Applied Resolution
LiDAR Alignment	Raw blob header offset error	Implemented fixed offset correction
Radar Tracking	Coordinate system mismatch	Inverted axes for sign consistency
YOLO Accuracy	Formula cars missing in training set	Deployed low-floor confidence cascade
Visual Crashes	Matplotlib colorbar redraws	Initialized static persistent colorbar

Future Iterations

Algorithmic Scope

- Multi-LiDAR extrinsic auto-calibration
- Pixel-level projection fusion
- Map/Track-aware motion priors
- Radar occupancy mapping

Platform Optimization

- Fine-tune custom YOLO weights for formula cars
- Real-time C++ inference pipeline rewrite
- RTK-GPS ground-truth verification

Ego Vehicle Trajectory & Heading

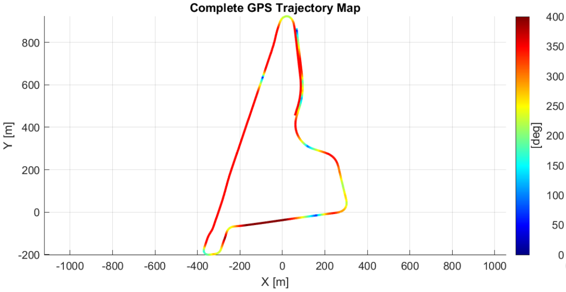


Figure 1: GPS Map with Speed Intensity

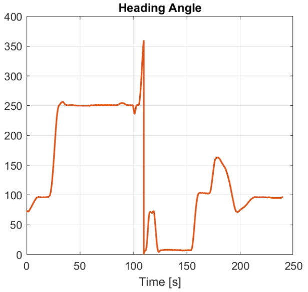


Figure 2: Unwrapped Heading Angle

- Sharp heading change at ∼110 seconds matches the tight hairpin turn at the bottom of the map.

IMU Acceleration & Velocity Profiles

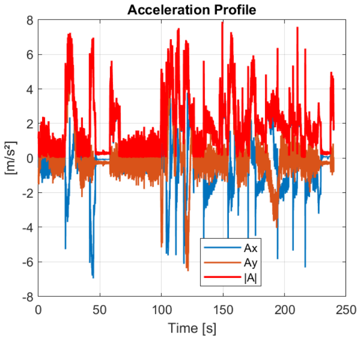


Figure: Total Acceleration Magnitude $|A|$

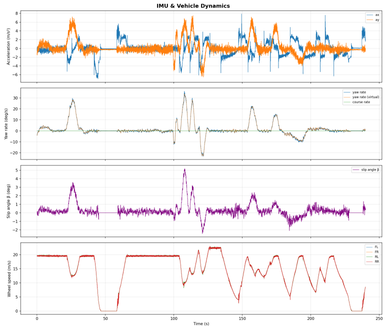


Figure: Multi-Axis Kinematic Signals

- Peak G-forces approach 8 m/s^2, correlating directly with wheel speed drops.

EKF State Uncertainty Bounds

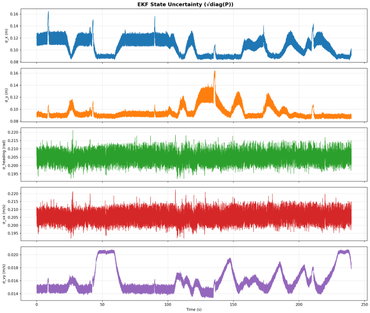


Figure: Covariance Bounds $\sigma = \sqrt{\text{diag}(P)}$ Over Time

- Error bounds remain strictly bounded, with minor lateral expansion during high-G turns.

Subsystem Benchmarks & Latency Analysis

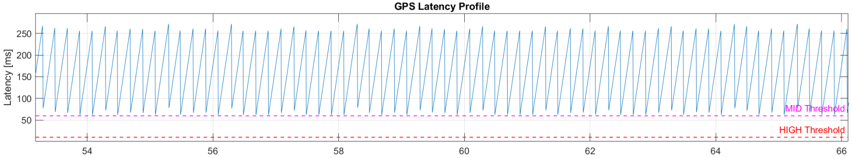


Figure 1: GPS Delay Profile

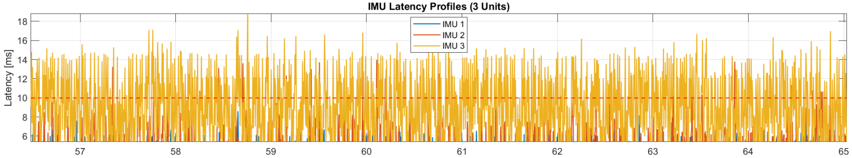


Figure 2: IMU Tri-Unit Latency

- Complete hardware uptime verified alongside stable sub-14 ms IMU latencies.

THANK YOU