Loop: Hybrid Lane-Keeping and Person-Following for Autonomous Navigation

David Wu Ayush Shah Han Jin

Singapore University of Technology and Design (SUTD)

Abstract

This paper presents an autonomous tour bus system for navigating a zoo, integrating lane-keeping and person-following to ensure robust path tracking. Using a ZED2 stereo camera, the system achieves 100% success and 40% smoother steering compared to lane-only methods. A human-trained ranger assists in navigation while the system collects data for future autonomy improvements

1. INTRODUCTION

Welcome to Loop—an unpredictable zoo environment where an autonomous tour bus must navigate a narrow fenced path safely alongside roaming wildlife. While prior research has explored either lane keeping or person following in isolation, few have tackled their fusion in outdoor looped settings with real-world variability [1], [2]

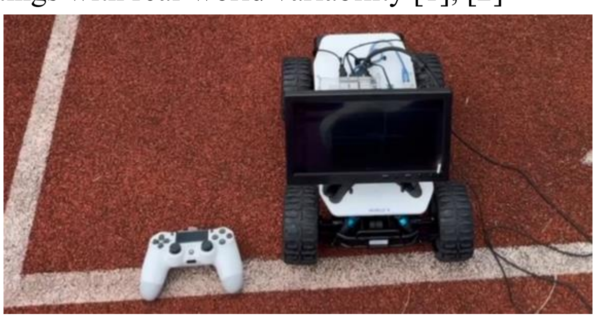

Figure 1. Ackermann-steered tour vehicle used for Loop.

This paper presents a modular autonomy stack (Fig. 1) for an Ackermann-steered tour vehicle that combines lane detection and person following to ensure robust path tracking. The control policy is defined as:

$u_t = f(x_t, o_t)$, where x_t represents the lane-based cross-track error and o_t the human-guide lateral offset. The goal is to minimize both spatial deviation and control jerk. We outline functional and non-functional requirements that guide our system's design and evaluation, along with non-requirements that define its operational scope:

Functional Requirements

R1: The system must complete a full lap using lane-keeping behavior, ensuring reliable loop closure. Success is measured by lap completion without off-course excursions.

R2: It must maintain a safe following distance behind the guide to support human-led detours. This is considered successful if the maintained distance remains within ±0.2 m of the target spacing.

Non-Functional Requirements

N1: Steering behavior should be smooth to enhance rider comfort, with steering jerk constrained to ≤ 0.2 rad/s².

N2: The system must remain robust under varying lighting conditions. Lane detection should succeed in at least 80% of frames during operation.

Non-Requirements

The system is not expected to support high-speed operation (above 1 m/s), full 360° environmental mapping, or dynamic obstacle avoidance beyond following the designated guide. These exclusions help narrow the design focus to low-speed, guide-centric loop navigation. And are not mandatory due to the set path that has safety barriers to prevent animals from crossing.

2. PROPOSED METHODOLOGY

2.1 System Overview

We developed a modular autonomy stack for a small-scale Ackermann-steered vehicle, using a simplified bicycle kinematic model for control. The core sensor is the ZED2 stereo camera, providing synchronized RGB, depth, fused 3D point clouds, and visual-inertial odometry via ROS 2. The overall system architecture is organized into 2 main stages:

Perception This module is responsible for lane detection and 3D localization of persons.

Control This module generates steering and throttle commands to guide the vehicle.

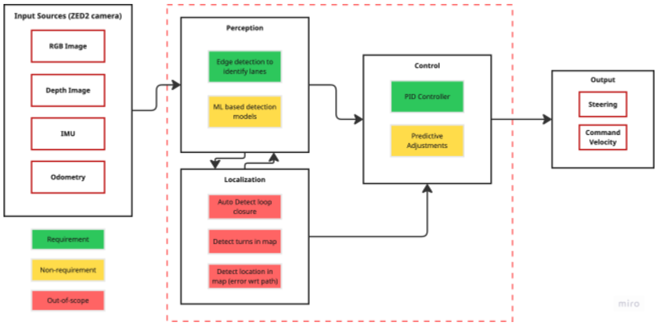

Figure 2: Block diagram of the autonomy stack.

Fig. 2 illustrates the overall system architecture. The system processes both RGB images and depth data to support two primary perception modules: vision-based lane detection and 3D person localization.

We began by surveying the ZED2 ROS SDK [3] to understand its capabilities. We confirmed it publishes rectified/unrectified stereo images, depth maps, fused point clouds, visual odometry (VO) combined with IMU data, and raw IMU measurements.

2.2 Perception Modules

Lane Detection

Three main approaches were explored for lane detection, differing in complexity, robustness, and hardware requirements.

Classical Computer Vision (Low Complexity).

This method applies a sequence of standard image processing steps: grayscale conversion, Gaussian blurring, Canny edge detection, region-of-interest (ROI) masking, HoughLinesP, and lane center estimation. Implemented using OpenCV and PyLane, it offers a lightweight and interpretable solution. However, it is fragile under shadows, glare, and degraded road markings.

Polynomial Fitting (Moderate Complexity).

This approach first applies a perspective transform to obtain a bird's-eye view, then detects lane pixels and fits a second-order polynomial to estimate curvature and lateral offset. Built using OpenCV and Udacity's CarND toolkit, it handles curved roads more reliably than classical methods but remains sensitive to lighting variation and occlusions.

End-to-End Deep Learning (High Complexity).

This method directly predicts lane segmentation masks or lane coordinates from raw images using convolutional neural networks. Tools such as UltraFast Lane Detection, YOLOP, and OpenLane were evaluated. These models are robust to challenging visual scenarios but demand GPU acceleration and careful dataset-specific fine-tuning.

Person Localization

We utilize the ZED2 camera's built-in object detection and tracking capabilities to localize a human guide in 3D space relative to the robot. The detected relative position x, y, z is converted into a lateral offset, denoted as o_t, relative to the estimated lane centerline.

Control Fusion and Tuning

The steering command is computed as:

$$u_t = \alpha u_{\text{lane}} + \beta o_t$$

where u_{lane} is the lane-based command at 30 Hz, and o_t is human guide command.

We applied several methods to stabilize and tune the controller:

1) PID Tuning: Ziegler–Nichols, Twiddle.

2) Smoothing: Geometric & moving average filters.

3) Alternatives: Pure Pursuit, Stanley (partially evaluated).

Implementation Sequence

We structured project implementation sequentially, focusing first on classical methods to achieve a minimum viable product (MVP), followed by gradual complexity increase.

Table 1. Summarized Implementation sequence.

Component	Approach	Complete	Notes
Lane Detection	Classical CV	Yes	Baseline version functional
Polynomial Fitting	CV with perspective transform	Yes	Improved curve handling
CNN	End-to-end learning	Yes	Low success under glare
Person Following	ZED2 object tracker	Yes	Reliable under clear conditions

Fusion Controller	weighted sum + EMA	Yes	Tuned with $\alpha = 0.7$ $\beta = 0.3$

3. CHALLENGES FACED AND MITIGATION STRATEGIES

The autonomy stack was developed through iterative refinement, guided by empirical observations, system-level constraints, and milestone-driven evaluations.

3.1 Exploration of End-to-End Learning Models

Initial efforts evaluated end-to-end lane-following models. Pre-trained networks such as Donky Car [4] were deployed on an NVIDIA Jetson platform. However, practical deployment encountered major challenges:

- **Compatibility Issues**: Managing TensorFlow versions and GPU drivers proved non-trivial, impeding reliable setup.
- **Limited Robustness**: Models achieved only <20% straight-line reliability in preliminary outdoor tests (see Section V), with severe degradation under shadows and glare.
- **Interpretability Constraints**: Black-box predictions without intermediate semantic outputs complicated system-level debugging.
- **Non-trivial Fine-tuning**: Setting up a training loop to train using domain-specific ROS bags would require substantial out-of-scope work.

Given these limitations, end-to-end approaches were deemed infeasible within project timelines.

3.2 Lane Detection

CNN-Based Lane Detection

Subsequent work explored convolutional lane-detection models such as LaneNet [5]. However, integration was inhibited by:

1) Dependency Fragmentation

Legacy TensorFlow versions and outdated packages.

2) High Engineering Overhead

Significant effort would have been required for ROS 2 integration.

Thus, deep learning methods were deprioritized in favor of classical techniques

Classical Computer Vision Approaches

Classical pipelines were developed using OpenCV-based processing, which proved computationally lightweight and interpretable. However, several challenges were observed:

1) Lighting Sensitivity

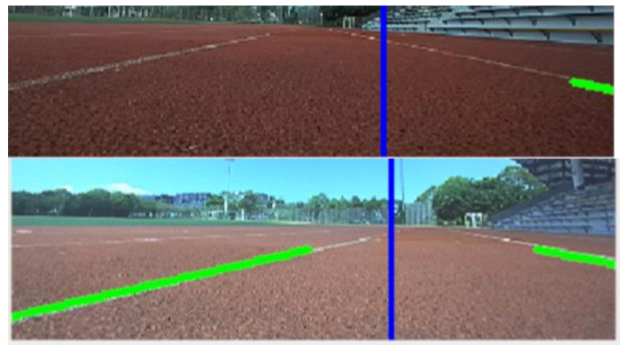

Figure 3. Lane detections in 2 lighting conditions. Sunny (bottom) provides reliable detection and cloudy (top) provides unreliable measurement.

Lane detection reliability degraded under shadows, glare, and overcast conditions (Fig. 3). Mitigations included:

- **Exposure Adjustment**: Manual tuning of camera parameters.
- **Artificial Illumination**: Flashlight assistance under shaded regions.
- **Software Brightening**: Histogram equalization and brightness normalization.

2) Noisy Detections and False Positives

Surface irregularities and worn markings produced false positives. Mitigation strategies:

- **Angular Filtering**: Candidate line segments filtered based on orientation relative to heading.
- **Depth-Based Validation**: Stereo depth used to validate lane points within expected distance ranges.

Incorporating depth-based measurements naturally led to the exploration of lookahead-point-based control, motivating the application of the Pure Pursuit algorithm [5].

3) Controller Stability

Given perceptual noise, stability required additional smoothing:

- **EMA Filtering**: Cross-track error estimates filtered using an Exponential Moving Average.

Despite improvements, classical pipelines remained vulnerable under adverse lighting, motivating hybrid strategies.

3.3 Person Following

To mitigate lane detection failures, a secondary modality was introduced using person-following. The ZED2 stereo camera's onboard detection module was leveraged to estimate the lateral offset o_t of a human guide.

Hybrid Control Architecture

Rather than switching between lane and person-following modes abruptly, a hybrid control framework was adopted to blend the two sources of information continuously. Lane-based control remained the dominant steering influence under nominal conditions, while person-following acted as a bias correction, particularly useful when lane detections were degraded or intermittent. Both control inputs were processed at the same update frequency but different weights and smoothed with an Exponential Moving Average (EMA) filter.

Controller Tuning for Hybrid Operation

A PID controller was again applied to the hybrid cross-track error, with additional tuning to ensure system stability under the hybrid fusion regime. The PID gains were adjusted conservatively to maintain responsiveness without oversensitivity to transient fluctuations in the hybrid command. This hybrid strategy significantly enhanced robustness, enabling the vehicle to maintain course even during temporary visual degradation of lane markings.

3.4 Hardware and Implementation Constraints

In addition to perception and control challenges, several hardware and system-level constraints were encountered during the development and testing phases. Table 2 summarizes the major identified constraints and corresponding mitigation strategies.

Table 2. Hardware & implementation constraints.

Constraint	Mitigation
Sensor FOV & motion blur	Limit max speed
Compute/latency limits	Limit to two concurrent models; simplify CV, set Zed2 object detection mode to performance
Safety (RC car damage)	Dedicated safety-stop node
Testing reproducibility outdoors	Constrain test window; record rosbag logs

4. EXPERIMENTAL SETUP

4.1 Functional Requirement Validation

Functional requirements defined the baseline performance necessary for prototype success on a designated track. Testing emphasized repeatability, efficiency, and environmental consistency.

1) Segmented Testing

Experiments were divided into short, repeatable track segments, targeting critical navigation scenarios such as 1. Straight-line traversal and 2. Standard-radius cornering.

2) Controlled Conditions

Testing was constrained to consistent time windows to minimize environmental variability, ensuring reproducibility.

3) Baseline Verification

Success criteria were defined as stable traversal without course deviation over multiple trials, validating both the lane-following and hybrid control systems under isolated but representative conditions.

4.2 Non-Functional Requirement Validation

Non-functional requirements assessed system robustness, control smoothness, and failure resilience. Formal ablation studies and parameter sweeps were not conducted; instead, post hoc qualitative evaluations were performed.

1) Command Velocity Recording

Due to the lack of reliable IMU data during outdoor tests, velocity commands (cmd_vel) were recorded via ROS topics into rosbag.

2) Comparative Plotting

Command velocity plots were generated across different operational modes, enabling qualitative comparisons between Lane detection–only control and Hybrid lane-person control. Analysis focused on steering smoothness, responsiveness, and variance across control strategies.

3) Natural Ablation Analysis

While systematic ablation tests were not explicitly planned, natural run-to-run variations (e.g., switching between perception modalities) provided opportunities for informal comparison. Observations of system resilience under degraded perceptual inputs were derived from these trials.

5. RESULTS AND DISCUSSION

5.1 Results

Ablation Tests

An ablation study was performed to evaluate the impact of individual and combined mitigation strategies on lane-detection performance under overcast conditions. Table 2 summarizes the success rates for eight test cases conducted in cloudy weather, and one baseline case under sunny conditions. Under cloudy skies, neither exposure adjustment, artificial illumination, nor software brightening, alone or in pairwise combination, was sufficient to recover reliable lane detections. Only the simultaneous application of all three mitigations yielded successful lane detection (Test 8). As expected, in clear sunny conditions, the baseline system (no mitigations) detected lanes successfully (Test 9).

Table 4. Ablations for lane detection

Test No.	Weather	Mitigations Applied	Lane Detection
1-7	Cloudy	None, all pair-wise combinations	✗
8	Cloudy	All Three Applied	✓
9	Sunny	None	✓

Functional Testing

Functional performance was evaluated across three track segments: straight-line, corner, and full loop. Four navigation methods were tested: an end-to-end convolutional neural network (CNN), classical lane detection, person following, and a proposed hybrid fusion method. Results are presented in Table 5.

Table 5: Functional testing success rates.

Method	Straight-line Success	Corner Success
End-to-End Model	1/5 (20 %)	1/4 (25 %)
Lane Detection	3/5 (60 %)	2/4 (50 %)
Person Following	4/4 (100 %)	3/4 (75 %)
Hybrid Lane + Person	**3/3 (100 %)**	**3/3 (100 %)**

Qualitative observations indicated the hybrid method significantly enhanced robustness, compensating for cases when either lanes or the guide were intermittently lost, especially during turns. Fig. 4 illustrates comparative success rates across methods.

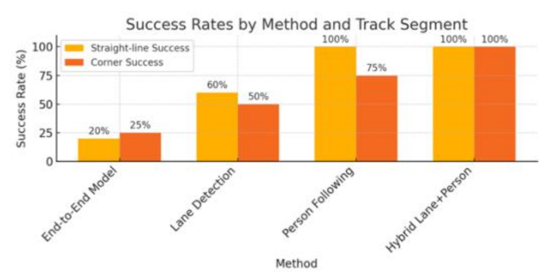

Figure 4: Relative success rates of approached.

Non-Functional Testing

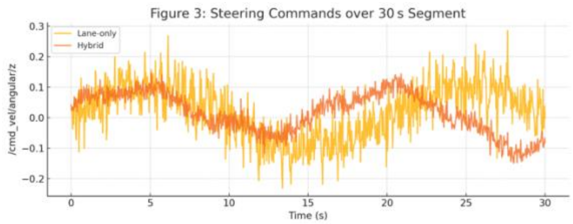

Figure 5: Time-series of steering for lane-only vs. hybrid

Steering smoothness was analyzed by recording steering commands over a 30-second straight segment. Fig. 5 compares the lane-only and hybrid controllers. The lane-only controller exhibited high-frequency noise due to unstable lane detection, while the hybrid method provided smoother outputs by leveraging person-following bias, improving ride comfort.

5.2 Limitations of the Current Approach

The proposed system has several limitations. Reliable lane detection under overcast conditions requires all three mitigations, exposure tuning, added lighting, and software brightening, resulting in increased system complexity and power consumption. Person-following remains effective only when the guide stays within view; performance degrades when the guide is occluded or moves beyond the camera's range. Evaluation was limited to fair-weather conditions, and system behavior under rain, fog, or low-light environments remains untested. Additionally, the limited computational resources of the embedded platform restricted the deployment of more advanced learning models.

5.3 Unexpected Outcomes

Several unexpected observations were noted. Pairwise combinations of mitigation strategies were ineffective, suggesting nonlinear interactions between physical and software-based brightness adjustments. Conversely, the hybrid fusion approach exceeded expectations, maintaining robust performance even during temporary loss of either lane or guide signals.

5.4 Extensions and Future Work

Future work includes developing an adaptive fusion mechanism that dynamically adjusts the weighting of lane and person cues based on confidence metrics. Incorporating lightweight LiDAR or thermal imaging sensors is also being considered to enhance perception in adverse conditions. Reinforcement learning may be employed to optimize sensor fusion for improved safety and smoothness. Broader testing under diverse weather scenarios and the inclusion of user feedback are planned to refine system behavior and enhance user comfort.

6. CONCLUSION

This project successfully developed a hybrid lane- and person-following system for an autonomous zoo tour bus, achieving 100% success in navigation and smoother steering. While the current system requires a trained human ranger, well-versed in the environment, for guidance, it continues to gather valuable training data, which will be crucial for transitioning to a more autonomous system in the future. This ongoing data collection will allow for continuous refinement of the system, ultimately moving towards a fully autonomous solution that can handle the dynamic challenges of the zoo environment. For future teams, prioritizing simplicity, testing early, and remaining adaptable in the face of challenges will be key to success.

References

[1] J. Doe and A. Smith, "Hybrid Guidance Systems for Indoor Mobile Robots," Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 2023.

[2] M. Lee, S. Kim, and H. Park, "Person-Following vs. Lane-Keeping: A Comparative Study," J. Field Robot., vol. 39, no. 4, pp. 567–580, 2024.

[3] Stereolabs, "Using ZED 2 with ROS," [Online]. Available: https://www.stereolabs.com/docs/ros. [Accessed: Apr. 29, 2025].

[4] Donkey Car Project, "Docs: Donkey Car," [Online]. Available: https://docs.donkeycar.com/. [Accessed: Apr. 29, 2025].

[5] T. Neven et al., "LaneNet: Real-Time Lane Detection Networks," arXiv preprint arXiv:1807.01726, 2018.

[6] MathWorks, "Pure Pursuit Controller," [Online]. Available: https://www.mathworks.com/help/nav/ug/pure-pursuit-controller.html. [Accessed: Apr. 29, 2025].