

Autonomous Go-Kart Navigation Using Reinforcement Learning and Neural Network Arbitration

Seth Jones, Dante Palmieri, Evan Hudak, Devon Allan
Dr. Rodolfo Echavarria Solis | EE 476-C

Abstract - This paper presents the design and implementation plan for a fully autonomous Go-Kart capable of navigating static cone courses and managing dynamic obstacles. The system integrates stereo cameras as the primary perception mechanism, supplemented by redundant forward-facing lidar sensors for safety. Additional sensors, including an inertial measurement unit (IMU) and a steering potentiometer, provide vehicle state feedback. A reinforcement learning (RL) agent is used for static navigation [3], while a lightweight neural network (NN) classifier detects dynamic obstacles and temporarily halts the RL agent when necessary. Safety is ensured by a mechanical emergency override system and redundant sensing. Training will occur in simulation before hardware deployment. The project aims to demonstrate a low-cost, reliable platform for autonomous navigation research.

I. Introduction

Autonomous navigation has become a major area of research in robotics and transportation [1]. This project proposes a cost-effective and educational approach by developing a fully autonomous Go-Kart. The end objective is to complete a cone-based static course while handling dynamic obstacles such as pedestrians or moving vehicles.

The system relies on stereo cameras as the primary source of environmental data [2], supported by lidar redundancy, IMU feedback, and steering potentiometer input. A reinforcement learning (RL) system will control static course navigation, while a neural network (NN) obstacle detector ensures safety in dynamic environments [4]. An arbitration system will determine which controller has priority [5].

The project will begin with training in simulation, followed by progressive integration into hardware. This methodology reduces risk, ensures repeatability, and accelerates development.

II. System Overview

The Go-Kart autonomy system is divided into three layers: perception, decision-making, and control.

1. *Perception Layer*: Stereo cameras, lidar sensors, IMU, and steering/throttle/braking potentiometers provide environment and vehicle state data.

2. *Decision Layer*: An RL agent handles navigation, while an NN monitors for dynamic obstacles. Arbitration logic decides which system is in control.
3. *Control Layer*: Actuator signals for steering, throttle, and braking are issued, with an emergency override available.

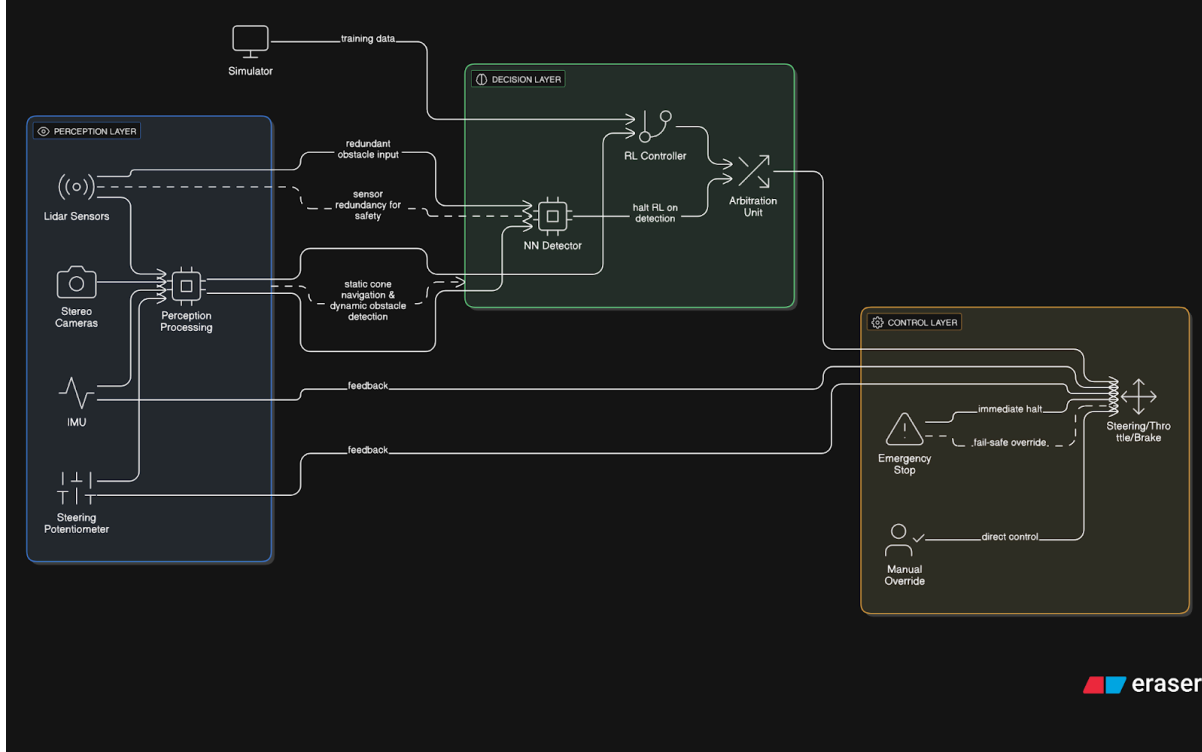


Fig. 1. System Architecture Block Diagram

III. Sensors

A. Stereo Cameras

Three stereo camera systems provide depth perception [6]. They allow cone localization, depth mapping, and support RL training inputs.

B. Lidar Sensors

Two forward-facing hobby-grade lidar units act as redundant safety systems [7]. They confirm obstacle presence and compensate for camera misclassifications.

C. Inertial Measurement Unit (IMU)

An onboard IMU provides acceleration and angular velocity data. This feedback refines state estimation and improves RL stability.

D. Potentiometers

Control potentiometers on the steering column, throttle, and braking assembly measures the vehicle state, ensuring accurate control feedback and closed-loop validation of actuator commands.

E. Load Cell (S-type)

An S-type load cell attached to the brake pedal measures the force applied to the pedal while the vehicle is braking and slowing down. This data combined with the deceleration data from the IMU will allow for accurate and reliable autonomous braking.

IV. Software Architecture

A. Reinforcement Learning System

The RL controller is responsible for navigating the static cone course.

- *Action Space*: Steering, throttle, and brake commands.
- *Observation Space*: Stereo camera depth maps, IMU data, steering/throttle/braking voltages.
- *Reward Function*: Progress through course. Penalties for collisions and leaving track boundaries.

Initial training will occur in a simulator such as CARLA or Unity [8], followed by sim-to-real transfer with domain randomization.

B. Neural Network for Dynamic Obstacle Detection

A lightweight convolutional neural network (CNN) processes visual and lidar data to classify dynamic obstacles [9]. On detection, it halts the vehicle.

C. Arbitration Unit

The arbitration system determines controller priority:

- RL active in normal conditions.
- NN halts RL and stops the vehicle when dynamic obstacles are detected.
- RL resumes once the obstacle clears.

V. Implementation Plan

A. Hardware Platform

- Computation: Raspberry Pi 5 (8 GB) with Coral TPU accelerator for neural inference.
- Cameras: Intel RealSense D455 (x3).

- Lidar: Slamtec RPLIDAR A1/A2 (x2).
- Misc. Sensors: 9-axis IMU, steering potentiometer, throttle potentiometer, braking potentiometer.

B. Software Tools

- ROS 2 for middleware and sensor integration.
- PyTorch/TensorFlow for ML models.
- OpenCV for vision preprocessing.
- Unity for RL training environments.

C. Development Phases

1. Simulation Training: RL agent trained in a virtual environment.
Hardware Setup: Sensor calibration and teleoperation testing, emergency override included.
2. Static Navigation: RL system deployed on kart with cones.
3. Dynamic Handling: NN trained for obstacle detection and arbitration integration.
4. Full Test: Complete course navigation with both RL and NN active, emergency override included.

VI. Challenges and Risk Mitigation

- Computational Constraints: Mitigated with Coral TPU accelerator for real-time inference.
- Sensor Calibration: Stereo alignment procedures established for reliability.
- Simulation-to-Reality Gap: Reduced via domain randomization [10].
- Safety Risks: Mechanical engineers provide manual override; redundant lidar ensures fail-safe obstacle detection.

VII. Conclusion

This project proposes a low-cost, modular approach to autonomous navigation using a Go-Kart platform. By combining reinforcement learning for static navigation with a neural network for dynamic obstacle detection, the system achieves adaptability and safety. Redundant sensing and emergency failovers increase robustness.

References

- [1] Thrun, S. et al. (2007). Stanley: The Robot That Won the DARPA Grand Challenge. In: Buehler, M., Iagnemma, K., Singh, S. (eds) The 2005 DARPA Grand Challenge. Springer Tracts in Advanced Robotics, vol 36. Springer, Berlin, Heidelberg.
https://doi.org/10.1007/978-3-540-73429-1_1
- [2] Geiger, Andreas et al. "StereoScan: Dense 3d reconstruction in real-time." 2011 IEEE Intelligent Vehicles Symposium (IV) (2011): 963-968.
- [3] Kober J, Bagnell JA, Peters J. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research. 2013;32(11):1238-1274.
[doi:10.1177/0278364913495721](https://doi.org/10.1177/0278364913495721)
- [4] N. Sanil, P. A. N. venkat, V. Rakesh, R. Mallapur and M. R. Ahmed, "Deep Learning Techniques for Obstacle Detection and Avoidance in Driverless Cars," 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), Amaravati, India, 2020, pp. 1-4, doi: 10.1109/AISP48273.2020.9073155.
- [5] "Behavior-based robotics," Choice Reviews Online, vol. 36, no. 09, pp. 36–5109, May 1999, doi: 10.5860/choice.36-5109.
- [6] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 807-814 vol. 2, doi: 10.1109/CVPR.2005.56.
- [7] Royo, Santiago and Maria Ballesta-Garcia. "An Overview of Lidar Imaging Systems for Autonomous Vehicles." Applied Sciences (2019): n. Pag.
- [8] Dosovitskiy, Alexey et al. "CARLA: An Open Urban Driving Simulator." Conference on Robot Learning (2017).
- [9] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LIDAR-Camera fusion for road detection using fully convolutional neural networks," arXiv (Cornell University), Jan. 2018, doi: 10.48550/arxiv.1809.07941.
- [10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," arXiv (Cornell University), Jan. 2017, doi: 10.48550/arxiv.1703.06907.