Vision-Based Real-Time Autonomous Racing for a Model Car

Technische Hochschule Mittelhessen (THM)
Department IEM – Control, Computer and Communications Engineering

Project: IPIE Project (Proposed)
Term: Winter Semester 2025

Student(s):
Shantanu Shende, Swayam Jakhalekar, Eya Chouk, Antoine Feuilette

Supervisor:
Prof. Dr.-Ing. Hartmut Weber

Processing Platform:
Laptop (Ubuntu Linux, C), Raspberry Pi 4, BLE-controlled model car

1. Project Overview

This project presents the design and implementation of a **fully autonomous, vision-based racing system** for a model-scale vehicle using a **single overhead camera** as the sole perception sensor.

Unlike earlier tracking or visualization-focused projects, this work emphasizes **real-time closed-loop autonomous control**, enabling **collision-free navigation at maximum feasible speed** on a dynamically changing racing track.

The system performs:

- Full perception from vision
- Dynamic track modeling
- Real-time state estimation
- Autonomous steering and speed control
- Low-latency BLE-based actuation

The project targets **deterministic execution**, **predictable timing**, and **fault-free operation**, aligning with real-world ADAS and autonomous driving principles.

2. Detailed Project Objectives

Primary Objectives

- Design a **vision-only autonomous racing pipeline**
- Detect track boundaries and compute a centerline in real time
- Estimate vehicle pose (x, y, θ) and velocity
- Implement **fully autonomous steering and speed control**
- Maintain **collision-free operation at high speed**
- Ensure **real-time execution with bounded latency**

Secondary Objectives

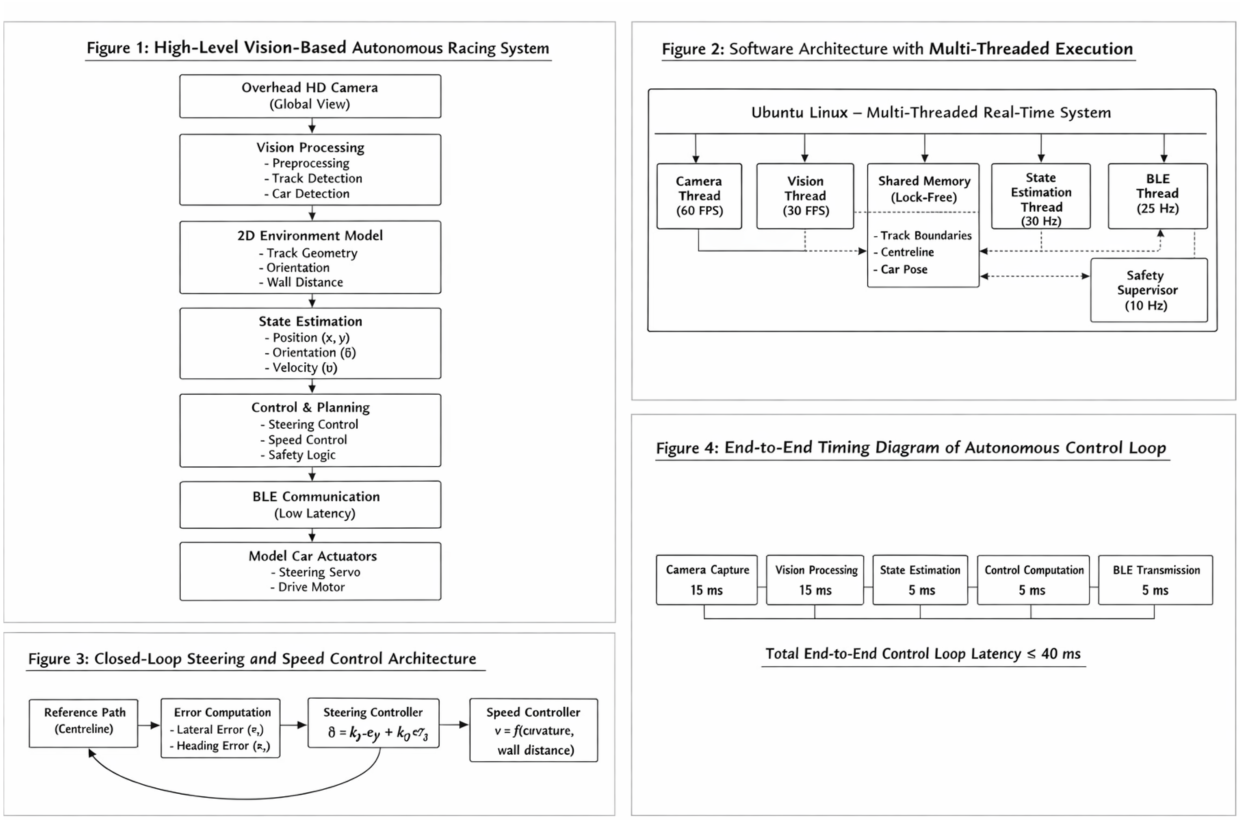
- Demonstrate robust operation under dynamic track layouts
- Introduce safety mechanisms (timeouts, emergency stop)
- Design the system to be extensible for future ADAS features

3. Hardware Components

Component	**Description**
Overhead Camera	Sony HD camera mounted on fixed tripod
Processing Unit	Laptop running Ubuntu Linux (C/C++)
Model Car	Driftstormkind BLE-controlled car
Communication	Bluetooth Low Energy (BLE)
Track	Modular racing track with red–white barriers

4. System Architecture

4.1 End-to-End Pipeline



This pipeline executes continuously in a **closed-loop real-time system**.

4.2 Baseline Operational Flow

1. Capture camera frame
2. Detect track boundaries
3. Compute centreline
4. Detect car position & orientation
5. Estimate speed and wall distance
6. Compute steering and speed commands
7. Transmit commands via BLE
8. Monitor safety constraints

5. Software Architecture (Multi-Threaded)

To minimize latency and jitter, the software is organized as **parallel real-time threads**.

Thread	Function	Frequency
T1	Camera Capture	60 FPS
T2	Vision + Track Modeling	30 FPS
T3	State Estimation	30 Hz
T4	Control & Planning	25 Hz
T5	BLE Communication	25 Hz
T6	Safety Supervisor	10 Hz

Threads communicate using **lock-free shared memory** and timestamped data.

6. Timing Diagram & Data Flow

Capture	Vision	State	Control	BLE
15ms	15ms	5ms	5ms	5ms

End-to-End Latency Target:
≤ 40 ms

This latency ensures stable high-speed autonomous operation.

7. Control Strategy (Formal Equations – Required)

7.1 Steering Control

Let:

- e_y = lateral error from centreline
- e_θ = heading error

Steering command:

$\delta = k\_y \cdot e\_y + k\_\theta \cdot e\_\theta$

Clamped:

$\delta\_min \leq \delta \leq \delta\_max$

7.2 Speed Control

Speed depends on curvature and wall distance:

$v\_cmd = \min(v\_max , k\_d \cdot d\_wall)$

Where:

- $d\_wall = \text{distance to nearest wall}$

8. (Optional) Kalman Filter

A discrete Kalman Filter improves pose and velocity estimation.

State Vector:

$x = [x , y , v\_x , v\_y]^T$

Prediction:

$x\_k = A\ x\_\{k\text{-}1\} + w\_k$

Measurement Update:

$z\_k = H\ x\_k + v\_k$

This reduces vision noise and improves control stability at high speed.

9. Safety Requirements (SaR)

Sr	Requirement
01	Vehicle shall stop if BLE timeout > 100 ms
02	Vehicle shall reduce speed near walls
03	Emergency stop on perception failure
04	Steering commands shall be rate-limited

10. Team Members & Capabilities (Updated)

Shantanu – Embedded Systems & Autonomous Control Lead

- Real-time systems, embedded C/C++
- ADAS control pipelines
- Safety-critical design
- System integration & optimization

Swayam – Control & Hardware Integration

- Control theory & tuning
- Vehicle dynamics
- Hardware interfacing

Eya – Vision & Data Processing

- Computer vision pipelines
- Image segmentation & tracking
- Performance optimization

Antoine – Communication & Safety

- BLE protocol
- Low-latency communication
- Safety & fault handling

12. Work Packages (Updated)

WP	Description
WP1	Requirements & Architecture
WP2	Vision & Track Detection
WP3	State Estimation
WP4	Control & Safety Logic
WP5	BLE Communication
WP6	Performance Optimization
WP7	Integration & Testing
WP8	Final Report & Demo

13. Conclusion

This project demonstrates a **fully autonomous, real-time vision-based racing system** using minimal hardware and strict timing constraints.
The system reflects **industry-relevant ADAS design principles**, making it suitable as a foundation for further autonomous driving research.