Automated Driving Using External Perception

Spring Test Plan
February 13, 2023

Team E - Outersense

Team Members:

Ronit Hire
Atharv Pulapaka
Dhanesh Pamnani
Jash Shah
Shreyas Jha

Carnegie Mellon University

Contents

1 Introduction

This document outlines various test plans to be conducted for the OuterSense system during the spring semester. This is to validate that the system meets the performance and non-functional requirements. The tests have been structured in a progressive manner, starting with simpler tests and gradually increasing in complexity as the subsystems evolve. This document also provides a timeline for completing each functionality. The results of the tests will be presented during progress reviews. By the spring validation experiment, a proof of concept (automated driving using external perception sensors) of the system shall be shown using a scaled-down model with 1 RC car and 2 infrastructure sensing units placed on linear track layout. A schedule for reaching the key milestones is included in the document.

2 Logistics

The tests (including the Spring Validation Demonstration), will be conducted in Newell-Simon Hall B-level. The demonstration will be a live presentation and all team members will be in attendance. For most of the tests, the following equipment will be required:

- **Modular track:** Test bed on which the vehicle will be driven. Must be modular to account for shift in location.
- **RC Cars:** Scaled down vehicle with ackermann steering system.
- **Intel Realsense cameras:** To estimate pose and detect deviation in yaw of the RC vehicle and act as feedback for high level control.
- **Infrastructure units:** Required to mount perception subsystem at an adjustable height.
- **WiFi router:** Required to establish a WLAN for communication between infrastructure units, the RC car, and the central decision making system.
- **NVIDIA Jetson:** Module required for edge compute on the infrastructure sensing units.
- **Monitor:** necessary for interfacing with single-board computer(s)

3 Schedule

Dates	Event	Capability Milestone(s)	Test(s)	Requirement(s)
Feb 15	PR 1	• Modular track surface is laid out • Test build of mechanical mounts for infrastructure sensors is complete • RC car can be controlled via microcontroller • Yaw of a colored identifier can be determined • Review of lane assist control examples in Simulink is finished	1, 2, 3	NFR1, FR1
Mar 1	PR 2	• Process for calibration of infrastructure sensors is established and functional • RC can be teleoperated via ROS messages • Track setup with all infrastructure units is complete	4, 5, 6	NFR1, FR1
Mar 22	PR 3	• Pose of static car can be estimated from from infrastructure camera • Velocity and steering profile generation from control block is working • RC car can track timestamped steering and velocity profiles	7, 8, 9	NFR1, FR1, FR3, FR5
Apr 6	PR 4	• RC car can follow motion cues sent from the central decision system • Perception system is deployed on edge compute and can communicate with decision system • Handover from one infrastructure unit to another is working	10 , 11	NFR1, FR1, FR3, FR4, FR5, PR3
Apr 19	SVD	• All subsystems are integrated, functional and can drive 1 RC car in a straight line	12	NFR1, PR2, PR3, PR4

4 Tests

4.1 Test1: Controller exploration and latency budget

Test 1 : Controller exploration and latency budget	
Objective	
Generate assisted steering for lane keeping in Simulink for different tracks	
Elements	Control System
Equipment	Computer, MATLAB
Location	NSH B-level (B512)
Personnel	Atharv Pulapaka
Procedure	
1. Launch MATLAB 2. Open Simulink 3. Open Simulink model for testing 4. Create track on driving scenario designer to test on different track designs 5. Run the Simulation and analyze the plots on data inspector and observe the vehicle behavior on birds eye view 6. Add delay and offsets in lateral deviation and yaw angle measurements and observe control response	
Verification Criteria	
Smooth response from controller, vehicle stays within the lane and has no jerks in steering inputs, verified via simulation results displayed on the screen.	

4.2 Test2: Control of RC car using a microcontroller

Test 2 : Control of a RC car using a microcontroller	
Objective	
Drive the RC car as per user defined steering and velocity controls with a microcontroller	
Elements	RC car customization (mechatronic system)
Equipment	1/10th scale RC Car Traxxas 4TEC2.0 Traxxas Velineon 3500 BLDC Motor 3S LiPo battery 30A programmable ESC Arduino Mega 2560 User interface device
Location	NSH B-level (outside MRSD cage)
Personnel	Shreyas Jha
Procedure	
1. Place the RC car in an open space 2. Connect the LiPo battery to the ESC 3. Establish a connection between the UI device and microcontroller 4. Provide steering and velocity commands on the UI to control the RC car	
Verification Criteria	
The RC car responds as per the user input, evident via visual inspection.	

4.3 Test3: Object detection and yaw estimation

Test 3 : Object detection and yaw estimation from a BEV (Birds Eye View)	
Objective	
Detect colored object and estimate it's yaw angle using Intel RealSense with a BEV	
Elements	Perception subsystem
Equipment	Intel Realsense D435i
Location	NSH B-level (B512)
Personnel	Jash Shah, Ronit Hire
Procedure	
1. Mount the camera at a known position with know orientation 2. Perform three-layer filtering (shape, size, color) 3. Create a baseline to detect angles about this segment. 4. Tune the filter based on lighting conditions and other external factors using track bars 5. Calculate relative angle between the base line and segment made by 2 corners detected on the shape.	
Verification Criteria	
Corroborate output of the algorithm against known ground truth angle value.	

4.4 Test4: Camera pose estimation from fiducial markers

Test 4 : External-camera pose estimation from fiducial markers	
Objective	
Ensure that the calibration pipeline is able to properly estimate extrinsics for camera	
Elements	Perception subsystem
Equipment	Intel Realsense D435i, Printed Aruco markers
Location	NSH B-level
Personnel	Jash Shah, Ronit Hire
Procedure	
1. Mount the camera at a known position with know orientation 2. Place the printed Aruco markers at known locations 3. Start the camera calibration program 4. Get intrinsics and extrinsics from the calibration program 5. Corroborate the extrinsics with ground truth values	
Verification Criteria	
Estimated camera extrinsics match the known rotation and translation.	

4.5 Test5: Stability test of infrastructure unit

Test 5 : Stability test of infrastructure sensing unit	
Objective	
Test the stability of the infrastructure unit with a mounted payload	
Elements	Infrastructure sensing unit
Equipment	Infrastructure unit, payload (embedded and perception hardware) counter weights, measurement tape, level gauge indicators, tension strings
Location	NSH B-level
Personnel	Dhanesh Pamnani
Procedure	
1. Mount the payload on the infrastructure unit 2. Place the cantilever at the required heights (2.2m - 3m) 3. Place counterweights on the mounting plate 4. Apply a push on the trunk of the infrastructure unit in any direction 5. Wait for wobble to stop 6. Repeat for other heights	
Verification Criteria	
The unit does not topple in any direction on application of mild push.	

4.6 Test6: Remote operation of the RC car

Test 6 : Remote operation of the RC car	
Objective	
Drive the RC car on the track using ROS enabled teleoperation	
Elements	Communication
Equipment	1/10th scale RC Car Traxxas 4TEC2.0 Traxxas Velineon 3500 BLDC Motor 3S LiPo battery 30A programmable ESC Raspberry Pi 4B WiFi Router Laptop with WiFi and ROS
Location	NSH B-level
Personnel	Shreyas Jha
Procedure	
1. Power on the pre-configured WiFi router 2. Place the RC car in an open space 3. Connect the LiPo battery to the ESC 4. Connect power to the Raspberry Pi 5. Connect the laptop to the WiFi network 6. Launch the ROS node on the laptop and publish steering and velocity commands via a topic 7. Visualize raw intrinsic sensor data from the RC car on the laptop	
Verification Criteria	
1. The RC car responds as per the user input. 2. The laptop continuously receives sensor data from the RC car, verified via the logs of the concerned ROS topic.	

4.7 Test7: On-track pose estimation

Test 7 : On-track pose estimation	
Objective	
Pose estimate of static RC car from infrastructure mounted cameras	
Elements	Infrastructure sensing unit and RC Car
Equipment	Infrastructure unit, realsense cameras, jetson board, counter weights, measurement tape, level gauge indicators, tension strings, RC Car
Location	NSH B-level
Personnel	Jash Shah, Ronit Hire
Procedure	
1. Mount the camera at a known position with know orientation 2. Fix the ArUCo markers at dedicated position on the track 3. Place the RC car in FOV of one of the infrastructure unit 4. Run the pose estimation script 5. Compare the estimated pose with predefined ground truth for the unit test	
Verification Criteria	
Pose estimate from the tracking algorithm matches the known ground truth.	

4.8 Test8: Simulation test of controller node

Test 8 : Simulation test central controller node	
Objective	
Generate finite horizon control signals from control blocks. When there is a loss in communication, the RC vehicle should have a steering and velocity profile which will ensure the vehicle stops.	
Elements	Control System
Equipment	Computer, MATLAB, RC car controller
Location	NSH B-level
Personnel	Atharv Pulapaka
Procedure	
1. Connect the controller to central system 2. Run the control block code that is generating new motion cues every time step 3. Monitor data received on the controller RC car controller	
Verification Criteria	
The velocity and steering profiles are smoothened to ensure the RC car comes to a halt after a fixed interval of new commands not being received. Verified by visualizing computed profiles on a monitor - the motion cues received in real-time by the on-board controller on the RC car.	

4.9 Test9: Communication systems test

Test 9: Communication systems test	
Objective	
Measure latency and datalink strength between central computing system, RC car, edge computing system	
Elements	Electronics and communication
Equipment	RC car, Intel Realsense cameras, Jetson Board, computer running decision making system, track setup, WiFi Router
Location	NSH B-level
Personnel	Atharv Pulapaka, Shreyas Jha
Procedure	
1. Set up the track, place the RC car on the track 2. Switch on the central system, the RC car, edge computing and perception sensors and confirm link is established between all units 3. To measure data link performance between edge compute and central system, analyze the number of data packets received and lost over a time period and ensure if desired number of packets containing properly defined data are received every time step 4. To measure data link performance between RC vehicle and central system, analyze the number of data packets received and lost over a time period from the RC car and ensure if properly defined and time stamped data are received every time step. To ensure data is sent from the central system to RC car check acknowledgement status from the RC car confirming commands are received every time step. 5. On confirming both data link systems are healthy, run the entire integrated system with central system, RCcar and edge compute and confirm if desired results are obtained	
Verification Criteria	
Desired results are obtained from the perception unit and RC vehicle moves correctly based on commands from the central system, all within allowed latency.	

4.10 Test10: Subsystem test for RC car

Test 10: Subsystem test for RC car	
Objective	
RC car tracks predefined and timed steering and control profiles, returns timestamped & processed sensor data	
Elements	RC Car Subsystem (mechatronics, communications, controls)
Equipment	1/10th scale RC Car Traxxas 4TEC2.0 Traxxas Velineon 3500 BLDC Motor 3S LiPo battery 30A programmable ESC Raspberry Pi 4B WiFi Router Laptop with WiFi and ROS
Location	NSH B-level
Personnel	Shreyas Jha
Procedure	
1. Power on the pre-configured WiFi router 2. Place the RC car in an open space 3. Connect the LiPo battery to the ESC 4. Connect power to the Raspberry Pi 5. Connect the laptop to the WiFi network 6. Launch the ROS node on the laptop and publish steering and velocity profiles via a topic 7. Visualize timestamped intrinsic sensor data from the RC car on the laptop	
Verification Criteria	
1. The RC car responds as per the set control profiles. 2. The laptop receives timestamped sensor data from the RC car and creates a dead-reckon state estimate (visualized on a monitor).	

4.11 Test11: Check handover from one sensor unit to another

Test 11 : Check handover from one sensor unit to another	
Objective	
Verify that detections always have the correct sensor ID associated with them	
Elements	Perception subsystem
Equipment	3 Intel Realsense cameras, Jetson Board, 1 RC car, 2 cuboids of blue color, computer running decision making system, WiFi router
Location	NSH B-level
Personnel	Ronit Hire
Procedure	
1. Mount all the cameras on infrastructure units 2. Power on the edge compute system and wait for the calibration script to execute 3. Wait for the perception system to be initialized 4. Place the RC car in FOV of one camera and the cuboids in FOVs of other cameras 5. Check the id associated with the detected objects on the Central Decision making system 6. Slide the objects from one FOV to the next and keep track of the IDs in the overlap region	
Verification Criteria	
All detected objects have the correct camera id associated with them and handover is detected by change in ID when objects move from the field of view of one camera to another.	

4.12 Test12: Spring Validation Demonstration

Test 12 : Spring Validation Demonstration	
Objective	
To demonstrate key progress made over the course of this semester and integration of the perception, control, and mechatronic systems.	
Elements	Integrated system
Equipment	Test track, 3 x Infrastructure unit OuterSense cyber physical systems Monitor
Location	NSH B-level
Personnel	Entire Team
Procedure	
1. Place the Infrastructure units besides the track at the appropriate position and height 2. Check power and readiness of the RC car by running it remotely 3. Power the infrastructure units 4. Calibrate the cameras 5. Set up the central Decision Making System 6. Run system checks for communication, perception, and controls 7. If no errors start with the demonstration, follow debugging procedure if required 8. Place the RC cars at the start line 9. Reset the clock to measure average speed of the vehicle in straight line 10. Repeat steps 11 and 12 11. Document all errors built up during the test and all other data obtained	
Verification Criteria	
1. The RC car is driven in a straight line within 12.5 % of lane boundaries. 2. The average speed of the RC car is greater than 12.5 cm/second.	

5 Appendix

5.1 Mandatory Functional and Non-functional requirements

S.N.	Mandatory Functional Requirements (FR)	FR ID
1	Drive vehicles using external perception	FR1
2	Avoid collisions with any controlled vehicle	FR2
3	Stop vehicles if there is an emergency	FR3
4	Track vehicles	FR4
5	Localize vehicles	FR5

S.N.	Mandatory Performance Requirements (PR)	PR ID
1	Ensure 0 collisions amongst controlled vehicles	PR1
2	Drive controlled vehicles within 12.5% of lane boundaries	PR2
3	Stop all controlled vehicles within 2 seconds of E-stop or communication loss	PR3
4	Pilot controlled vehicles at an average speed 12.5 cm/sec	PR4

S.N.	Mandatory Non - Functional Requirements (NFR)	NFR ID
1	Demonstrate proof of concept on a scaled down model	NFR1
2	Have minimum number of infrastructure sensor units	NFR2

S.N.	Desired Performance Requirement (DPR)	DPR ID
1	System shall control 4 vehicles	DPR1
2	System shall drive vehicles at maximum 20 cm/sec speed	DPR2
3	System shall have 0 collisions with pedestrians and rogue on road	DPR3
4	System will identify selected vehicles 100% of the time	DPR4