UTAC Challenge - Lotus 7 automation

Technical presentation

Arne Jacobs, Arthur Coron, Nicolas Damageux
Basile Mollard, Tihbault Edouard, Ambre Ricouard

ENSTA Bretagne

May, 2024

Team composition

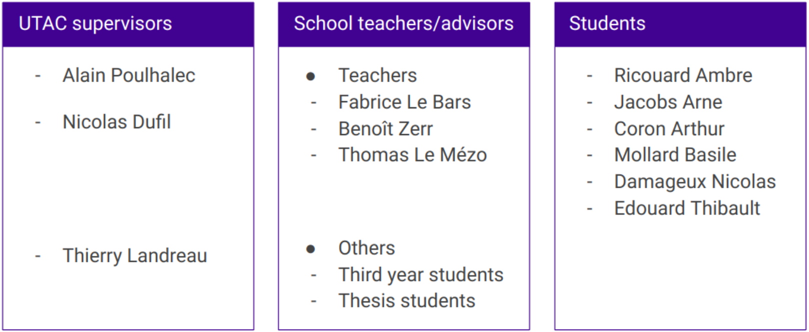


Figure 1: All parties involved

Organization

Table 1: Tasks distribution

Student Names	Tasks Accomplished
Ambre Ricouard	GNSS/IMU calibration and data recovery
Arne Jacobs	Lidar calibration, data segmentation and obstacle detection
Arthur Coron	Direction control and computer communication
Basile Mollard	GNSS/IMU calibration and data recovery
Nicolas Damageux	ROS2 architecture - CoppeliaSim Simulations
Thibault Edouard	Line detection

Project Overview

From Vehicle Architecture to Robotics

- Former teams composed of vehicle architecture students
- Supervisors with expertise in mechanical parts

A Multi-Disciplinary and Complex Project

- Involves multiple fields: automation, simulation, mechanics, electronics, embedded systems, control, etc.
- Transition from theory (simulations) to practice (track tests)

Our Dedicated Car

- The Lotus 7 Series 2
- Accessible to students, teachers, supervisors, interns, and thesis researchers

Project Objectives and Results

Objectives

- Navigate the highway circuit of the UTAC challenge.
- Meet the requirements to participate in the challenge.
- ENSTA Bretagne's participation in the 2024 challenge aims to combine skills from multiple disciplines to present a physical autonomous vehicle.

Results

- Finalized car assembly, including actuators and sensors.
- Real practice sessions with the car began in mid-April.
- Focus was primarily on simulations.

Implementation choices

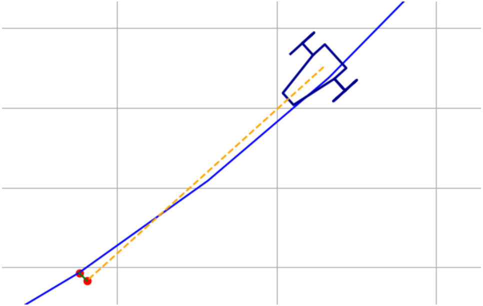


Figure 2: Long-distance control

Lateral control :

- e_Δ : distance : red point/the nearest point on the trajectory ;
- e_ψ : angular deviation : tangent/vehicle heading ;

Mathematical modeling of our car

The kinematic model of the bicycle is used , and is given by :

$$\begin{cases} \dot{x} = v.\cos(\theta).\cos(\delta) \\ \dot{y} = v.\sin(\theta).\cos(\delta) \\ \dot{\theta} = v.sin(\delta) \\ \dot{v} = u_1 \\ \dot{\delta} = u_2 \end{cases}$$

, with :

- (x, y) the cartesian coordinates of the vehicle ;
- v its speed ,
- θ the vehicle's heading ;
- δ the wheels' angle ;

Simulation : TEQMO

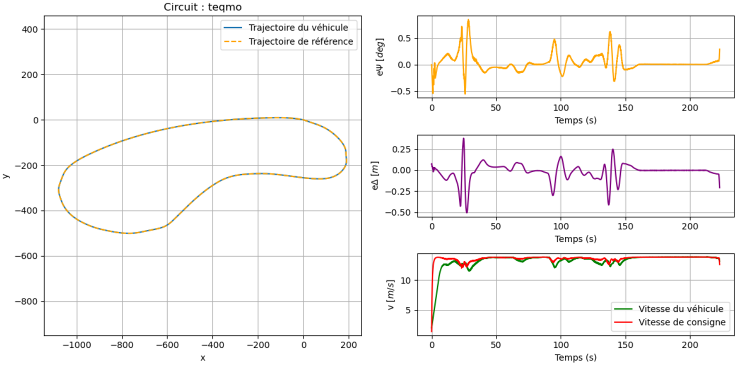


Figure 3: Control : TEQMO

Polynomial approximation

Inspiration from : https://github.com/ndrplz/self-driving-car/tree/master

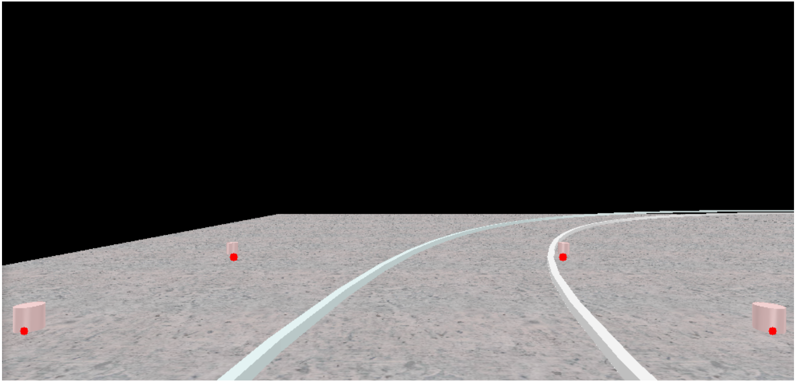


Figure 4: CoppeliaSim : vision calibration

Polynomial approximation

Inspiration from : https://github.com/ndrplz/self-driving-car/tree/master

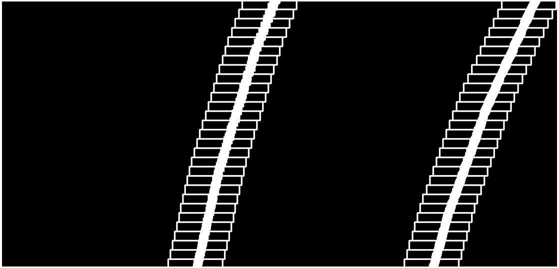


Figure 5: CoppeliaSim : line detection

IRL application : https://youtu.be/EyIWkzR3NIk

1. Presenting the context

2. Technical description
   - Inertial contral
   - Line following
   - Obstacle detection
   - Finished product
   - The car itself

3. Future prospects

4. Conclusion

5. Appendices

Using PC2 messages

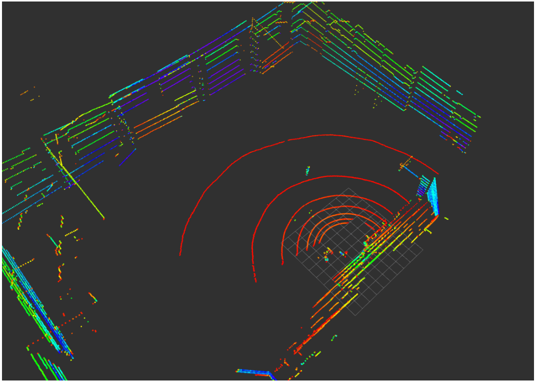
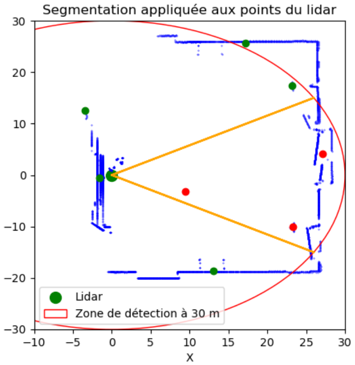


Figure 6: Environment vizualisation

Segmentation and detection



First detection algorithm

- DBSCAN segmentation algorithm used ;
- Does not rely on potentiometer ;
- Pessimistic ;
- 45deg opening, 30m radius ,

Simulations

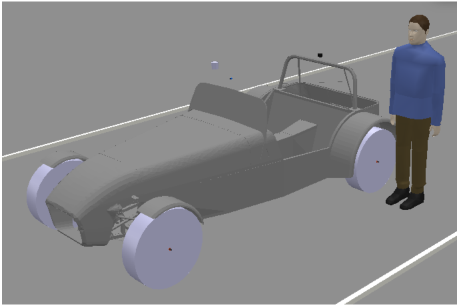


Figure 7: Our Lotus 7 in CoppeliaSim

CoppeliaSim : advanced robotic simulation platform that allows for modeling and testing robotic systems in a virtual environment (AIRBUS, ESA, etc.).

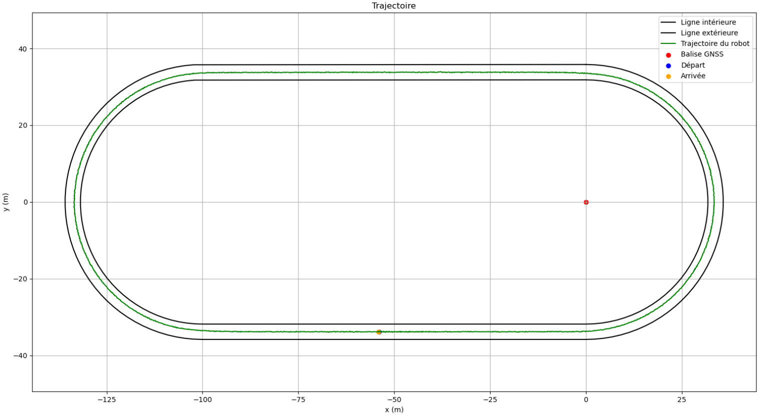


Figure 8: Finale simulation

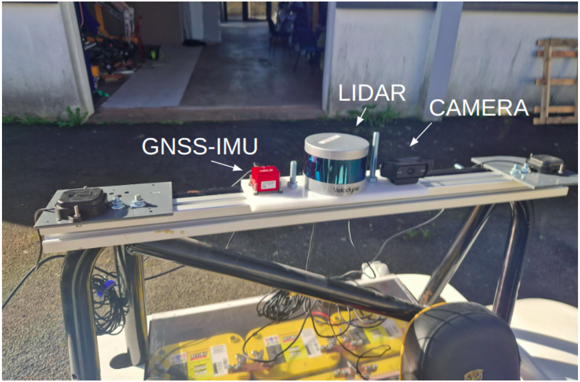


Figure 9: Our 3 main sensors

1. Presenting the context
2. Technical description
3. Future prospects
4. Conclusion
5. Appendices

Future Work and Continuation

Project Timeline

- A few months of work to have a robot using all actuators.
- Less than a year to present a functional car.
- Future improvements are expected.

Continuation Plan

- Project continued in 2025 by other robotics students.
- Access to all current tools and progress will be provided.
- Follow-up will include:
  1. Use of all actuators, especially the motor.
  2. Optimal line monitoring, considering tunnels and line cut-offs.
  3. Possible interval analysis/SLAM implementation via lidar.

Conclusion

Summary of Achievements

- Successful integration of GNSS/IMU, Lidar, and other sensors.
- Development and testing of control algorithms in both simulations and real-world tests.

Key Learnings

- Interdisciplinary collaboration is crucial for complex projects.
- Transitioning from simulations to real-world applications presents strong challenges.
- Continuous testing and iteration are essential for achieving reliable performance.

Thank you for your attention!

$$u_2 = k_\psi \cdot e_\psi + k_{i\psi} \cdot \int e_\psi + k_{d\psi} \cdot \dot{e_\psi} + \arctan\left(\frac{k_\Delta \cdot e_\Delta + k_{i\Delta} \cdot \int e_\Delta + k_{d\Delta} \cdot \dot{e_\Delta}}{k_s + k_d \cdot v}\right)$$

Figure 10: PID u_2

$$u1 = k_v \cdot e_v + k_{iv} \cdot \int e_v + k_{dv} \cdot \dot{e_v}$$

Figure 11: PID u_1

Simulation 1 : Austin

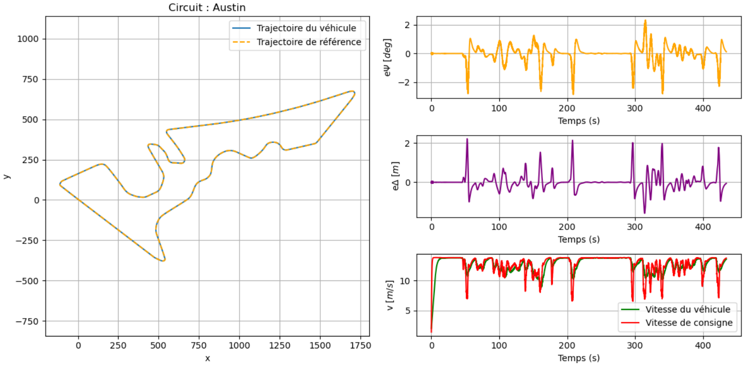


Figure 12: Control : Austin track

https://youtu.be/-dgl1t4Jhhs

Simulation 1 : Monza

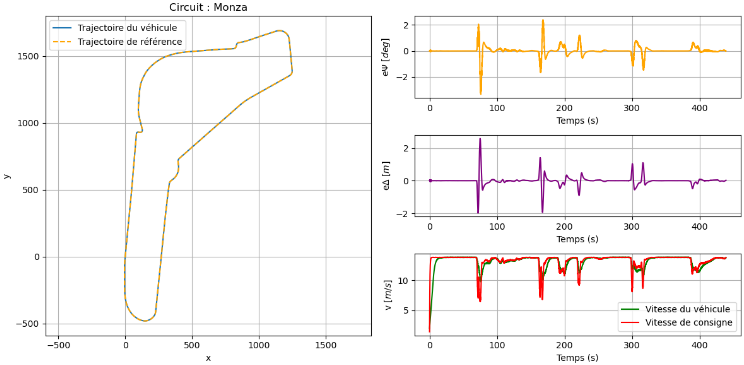


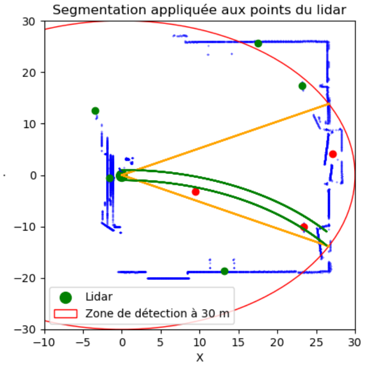
Figure 13: Control : Monza track

https://youtu.be/cig3Yc7v3FQ

Example : CoppeliaSim

```
src = np.float32([[width-30, 435],           # br
                  [30, 435],                 # bl
                  [300, 340],                # tl
                  [width-300, 340]])         # tr
dst = np.float32([[width, height],           # br
                  [0, height],               # bl
                  [0, 0],                    # tl
                  [width, 0]])               # tr
```

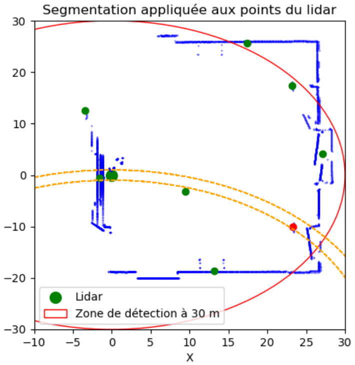
Second detection algorithm



Second detection algorithm

- Does rely on potentiometer ;
- Pessimistic ;
- Variable opening, 30m radius ,

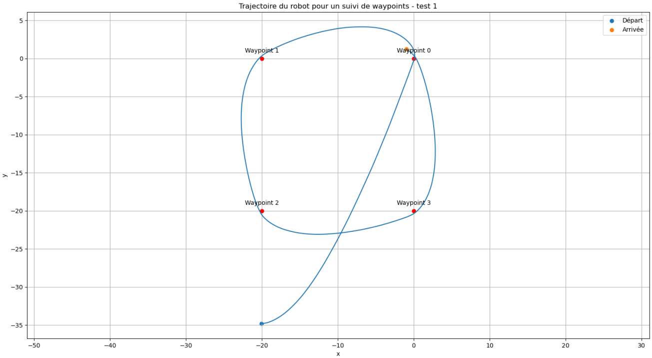
Third detection algorithm



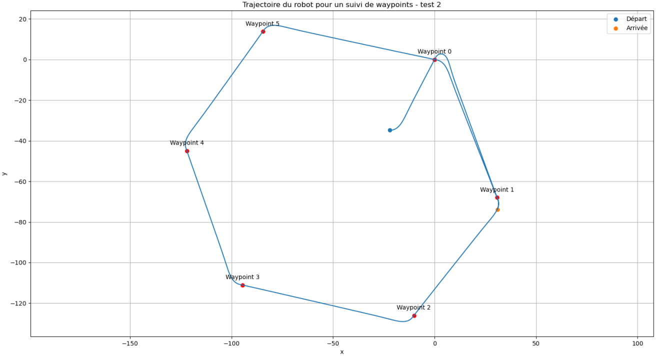
Third detection algorithm

- Does rely on potentiometer ;
- Optimistic ;
- 30m radius ,

4 waypoints


Trajectoire du robot pour un suivi de waypoints - test 1
Départ
Arrivée
Waypoint 0
Waypoint 1
Waypoint 2
Waypoint 3
x
y
5
0
−5
−10
−15
−20
−25
−30
−35
−50
−40
−30
−20
−10
0
10
20
30

6 waypoints


Trajectoire du robot pour un suivi de waypoints - test 2
Départ
Arrivée
Waypoint 0
Waypoint 1
Waypoint 2
Waypoint 3
Waypoint 4
Waypoint 5
x
y
−150
−100
−50
0
50
100
20
0
−20
−40
−60
−80
−100
−120