

Autonomous driving: an intro

Real-Time Embedded System - The F1tenth
autonomous racing



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

High Performance
Real Time **Lab**



Course outline

- › Intro course + basics of AD
- › ROS2: Installation and profiling
 - Ex: ROS2 to HiL, open a bag
- › Navigation: FTG, FTW, Pure pursuit
 - EX: navigation HiL
- › Perception: scan matching, PF, LIO?
 - Ex: perception (PF with PThreads)
- › Build the car

I do not cover all aspects of AD!!!

- › Systems and control theory => Prof. Falcone
- › Platforms and algorithms for autonomous systems => Prof. Sanudo & Prof. Falcone
- › High-Performance Computing => Prof. Marongiu (FIM)
- › Machine Learning => Cucchiara's

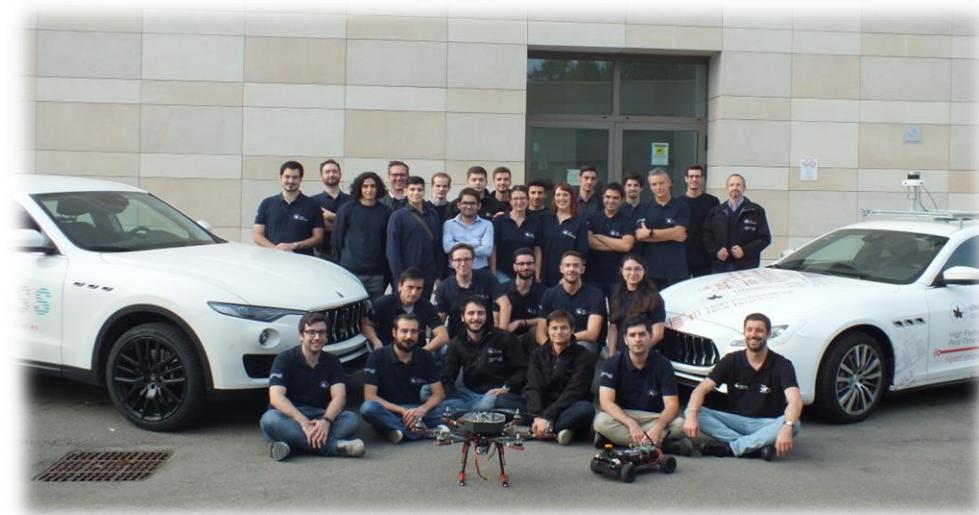


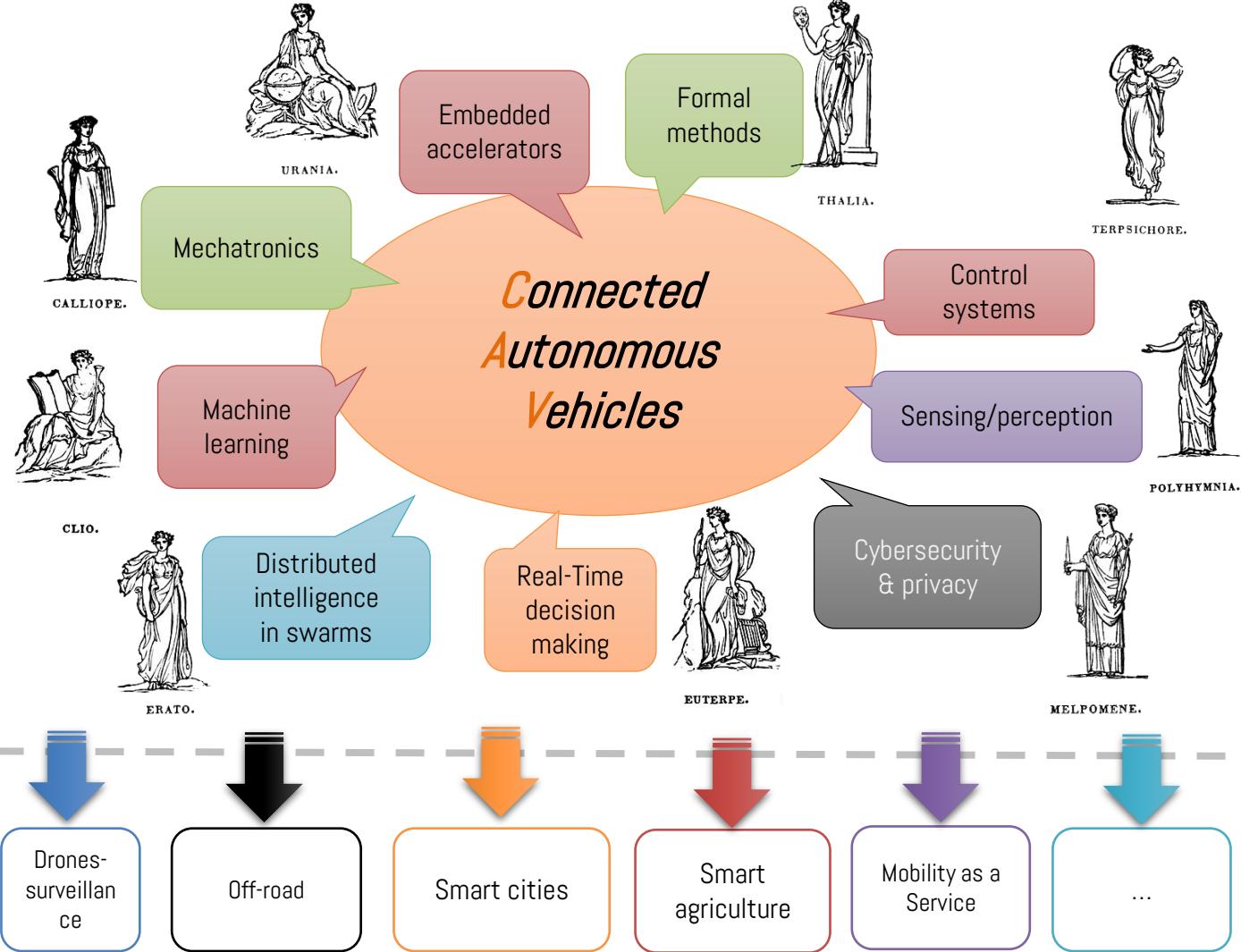
Credits

- › The official “full” F1/10 course
 - Uni Penn/Uni Virginia
- › It's a team work!
 - Hipert Lab & Hipert srl



UNIMORE High Performance
Real Time **Lab**
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA







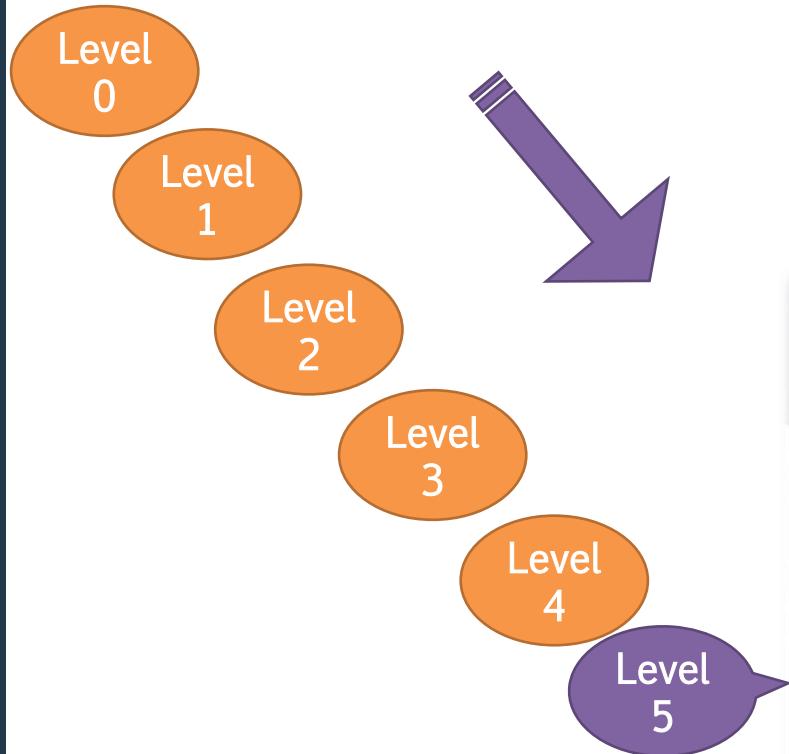
Self driving cars

Capable of **sensing** the environment
and **moving** safely
with **little or no human input**

- › It's hard to model/interpret the world and be able to act intelligently
- › Easy to detect the color of a traffic light, not that easy if the sun is directly behind the traffic light

Moravec's paradox: "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility"

The importance of standardisation





Level 0
Traditional cars

0



Level 1

Vehicle assists the driver

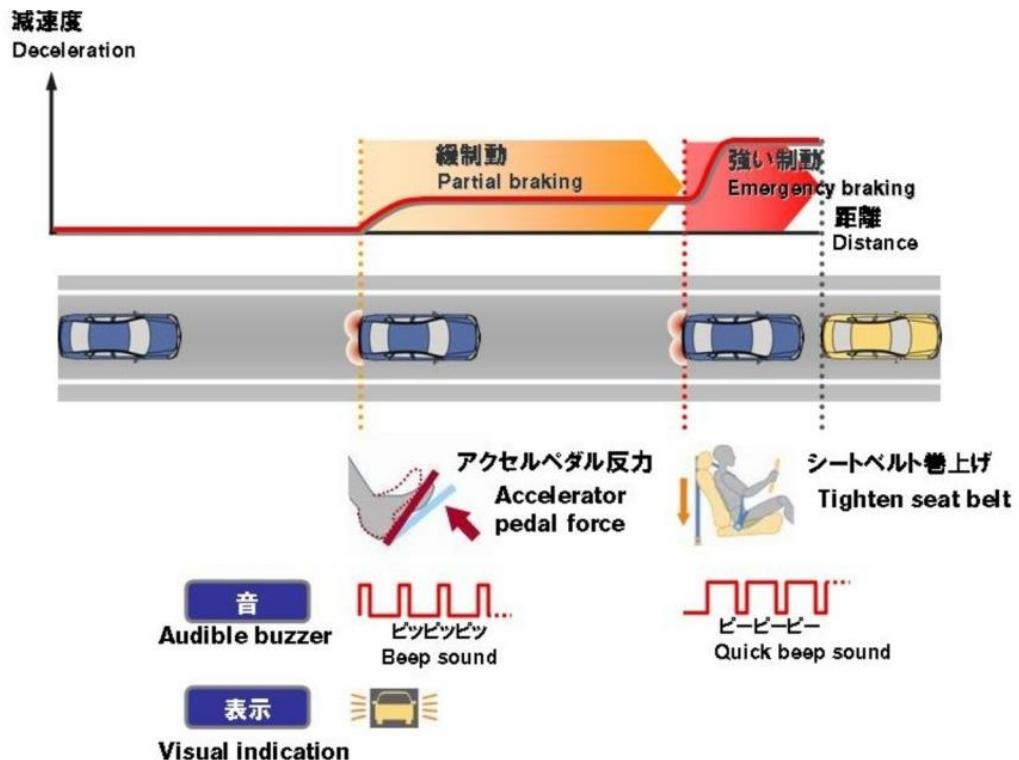




L1: assistant braking

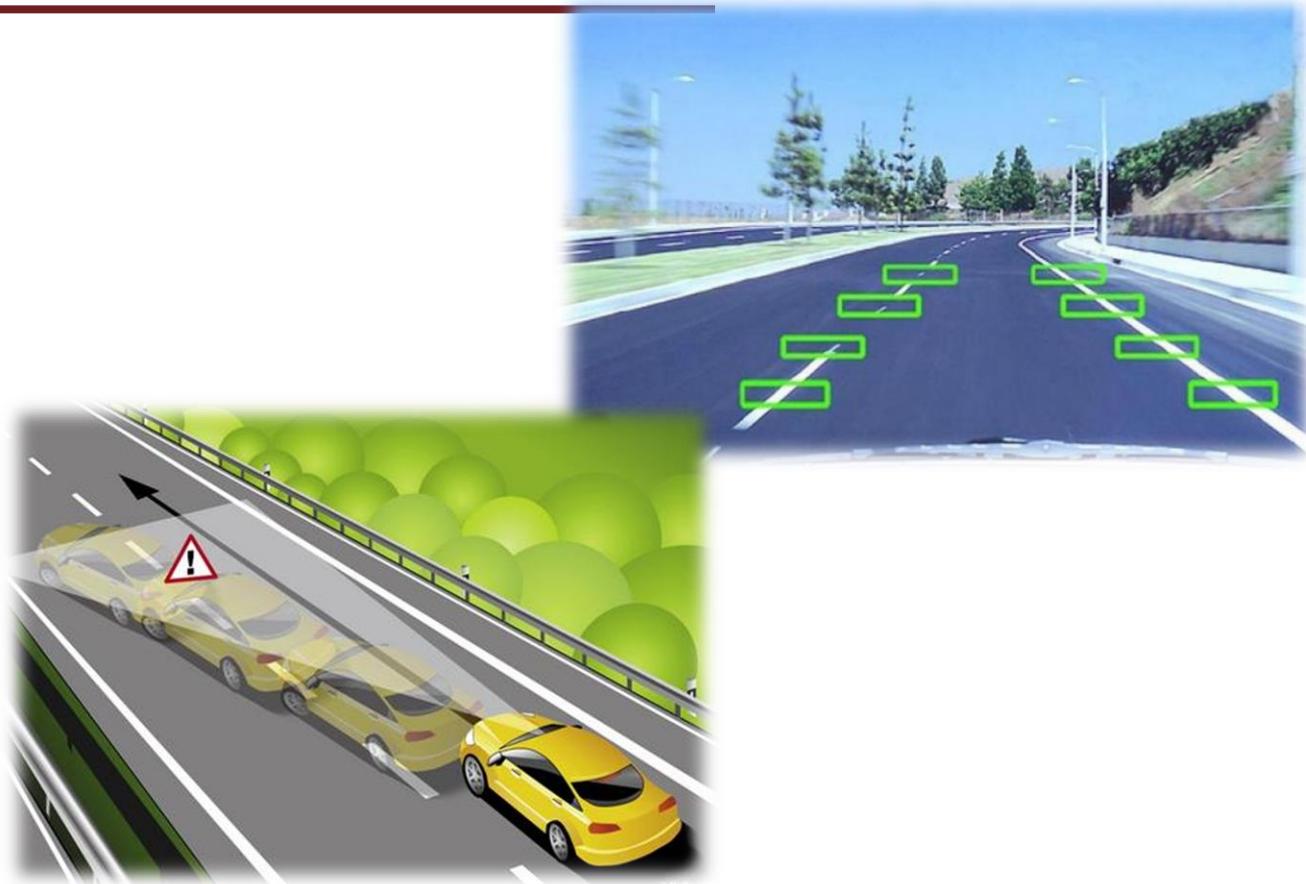
› Nissan's

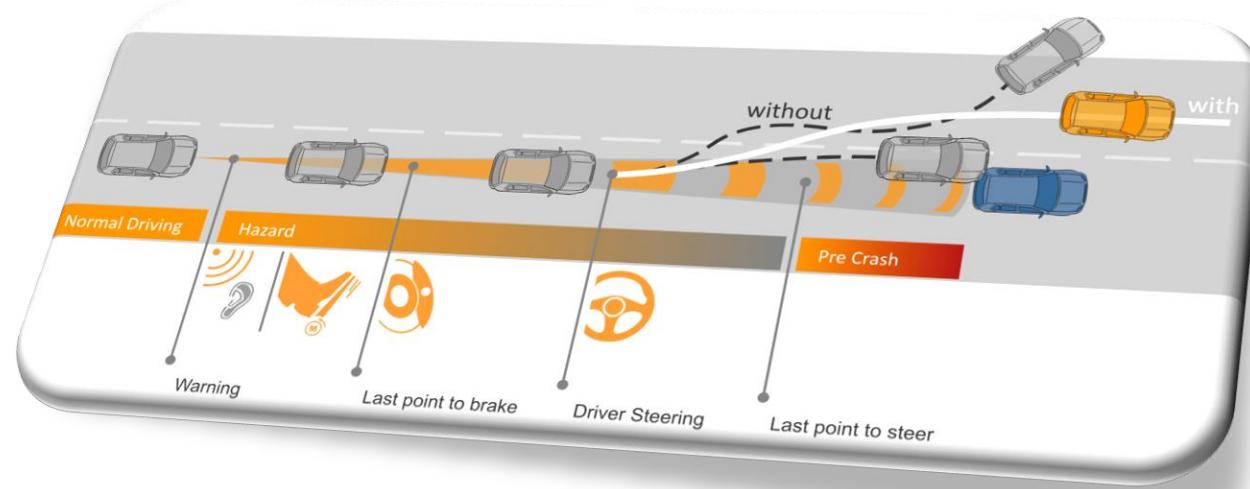
› Also, warnings
– E.g., buzzer



L1: lane centering

- › Detect lane
 - Does not overtake





Level 2

2+ features, but driver
still controls the car





Level 3

Car becomes a co-pilot



Level 4

Car becomes the pilot

Safety-critical driving functions are performed by the car in well-known Operational Design Domains

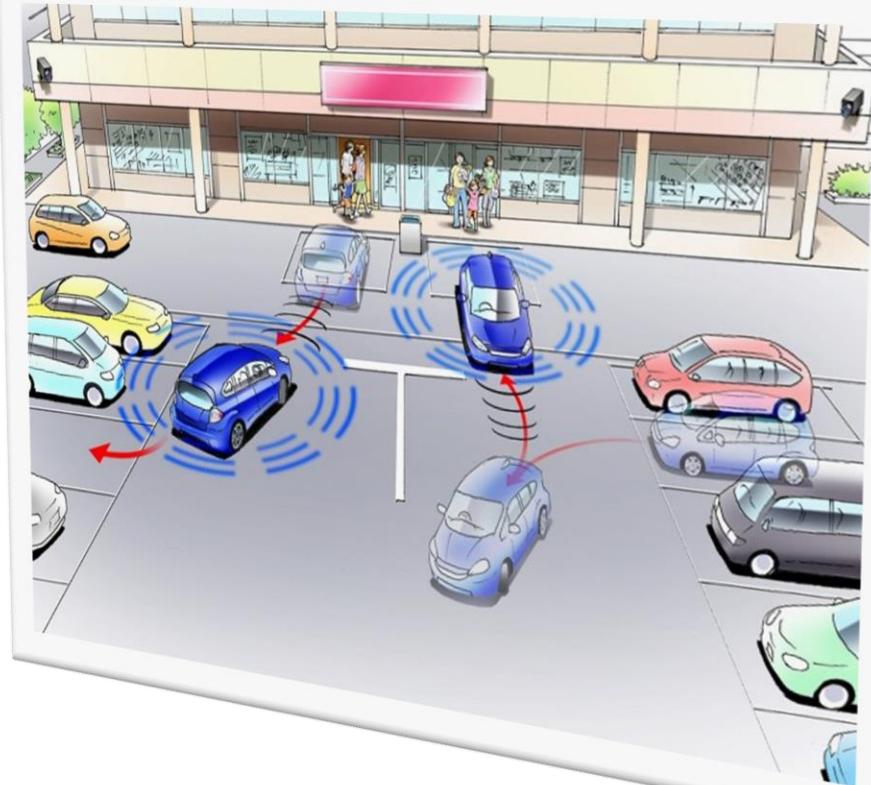
L3 (/4): valet parking

Autonomous search, park and return functionality

- › E.g., at shopping mall, cinemas, ...

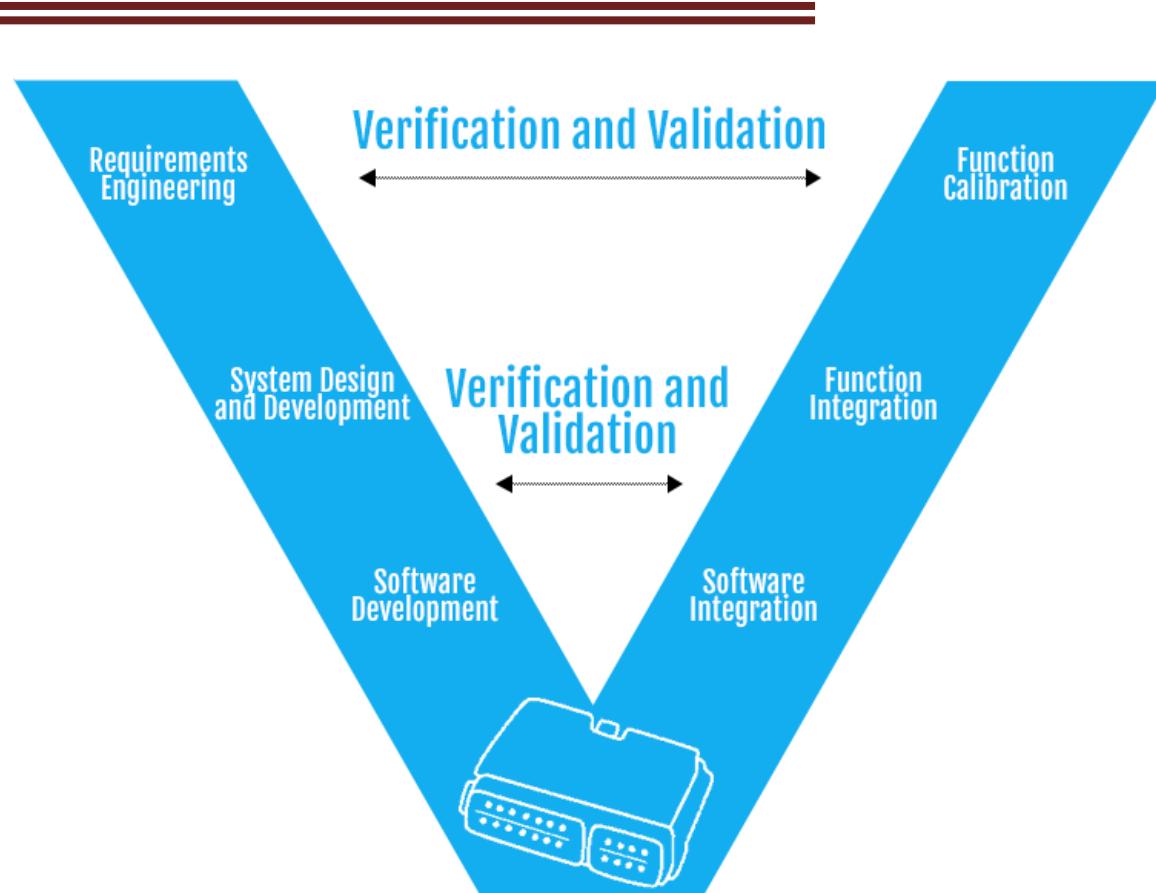
No humans, nor cars inside the parking area (ODD)

- › Leave your car at entrance

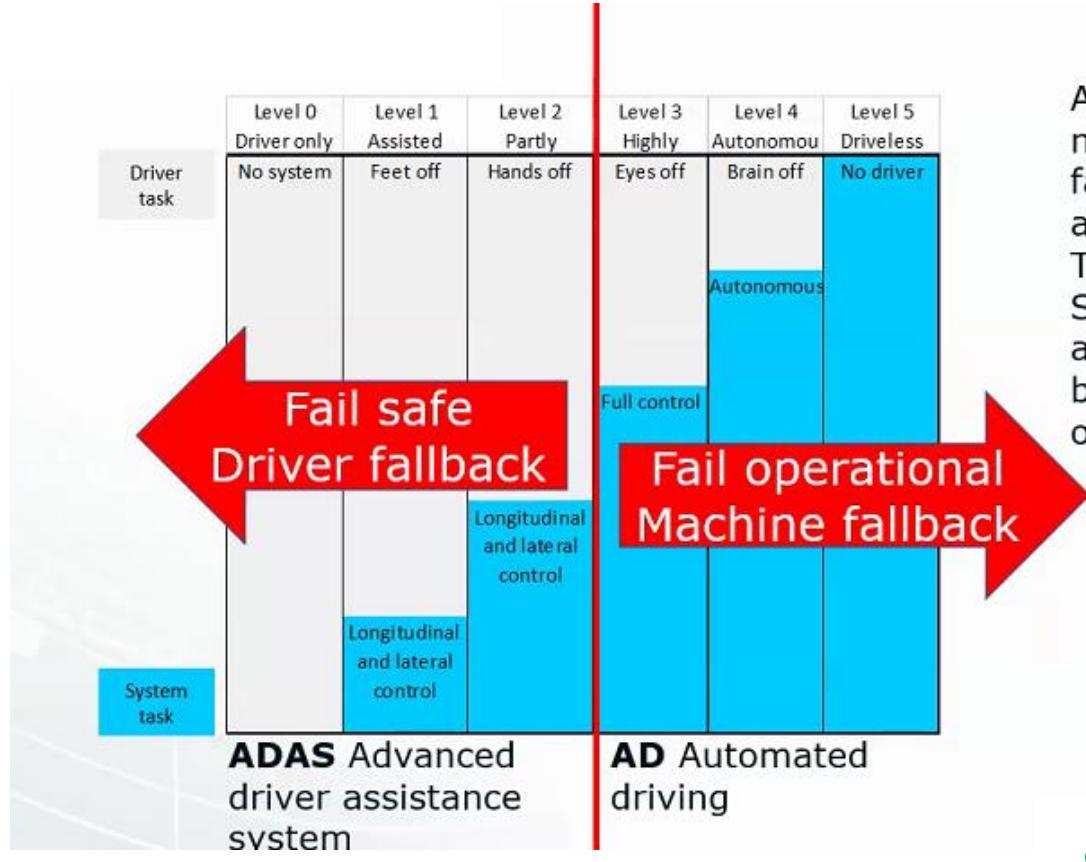




The V-Cycle automotive Software Development Process:



From fail-safe to fail-operational



After SAE level 3 it is necessary to have a fail-operational approach. The whole system: Sensors, ADAS ECU and actuators, has to be designed fail-operational.



Extraurban Obstacle Avoidance

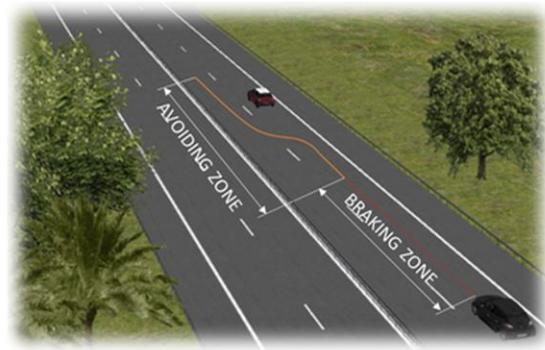
[Danisi, Marelli, 4SG, I&M, POLITO]

A fail operational Extraurban Obstacle Avoidance application to be demonstrated on a real vehicle (UNIMORE Maserati) and on a simulated environment (DANISI dynamic simulator)

FO Aspect: Fail-operational onboard system based on two redundant computers

The goal

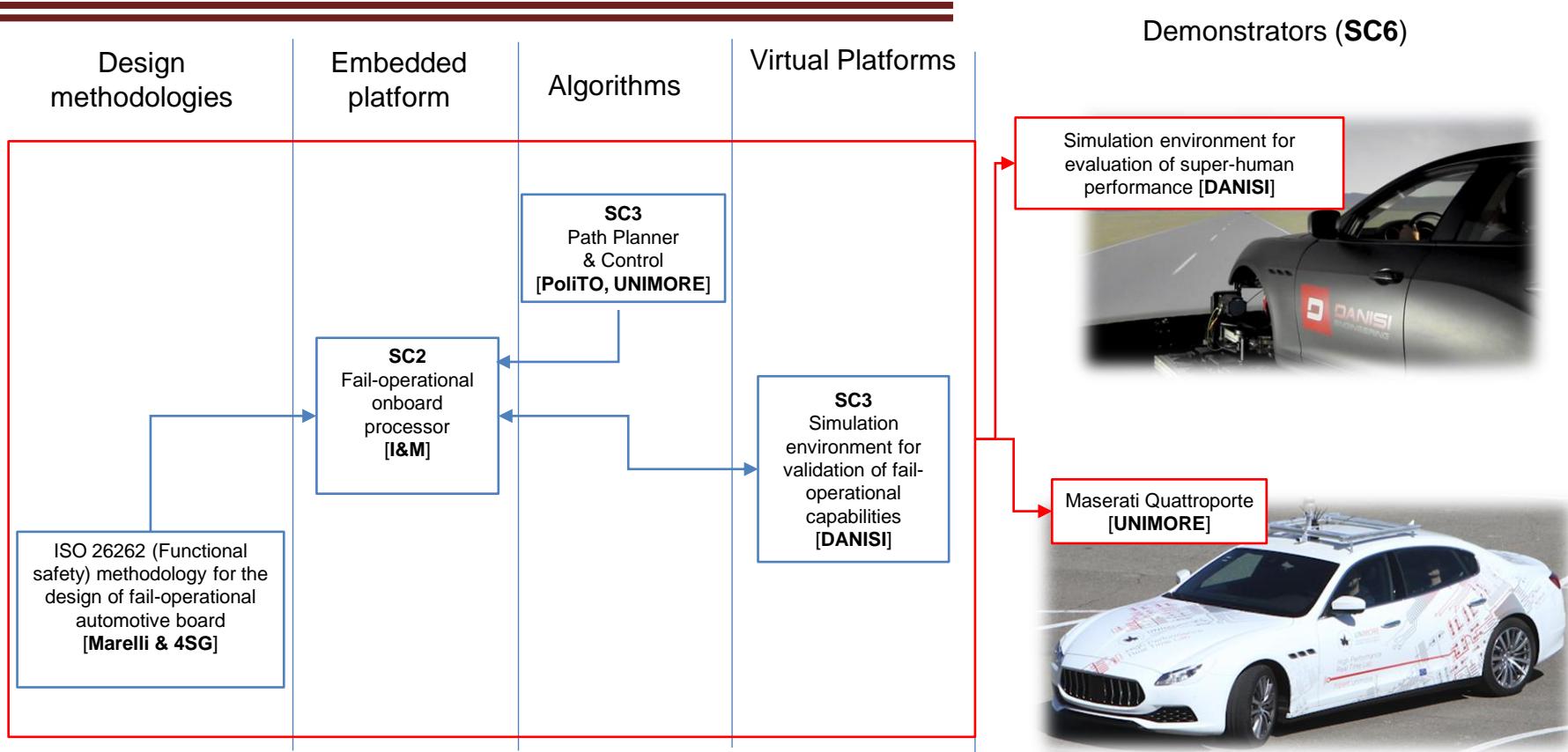
- › Being able to switch between the two computers during a L4/L5 manoeuvre



The scenario

- › Start the manoeuvre
- › Inject/simulate a fault in the board 'A'
- › The board 'B' seamlessly takes over, and completes the manoeuvre





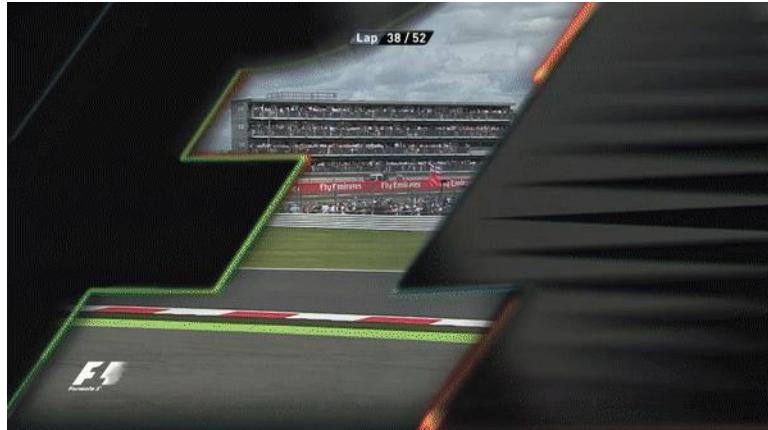


Balancing Performance and Safety

Current AV technology still struggles in non-cooperative scenarios like merging due to competing objectives:

- ✓ **Maximize performance:** negotiate the merge without delay or hesitation
- ✓ **Maintain safety:** avoid catastrophic failures and crashes

Racing (autonomously) highlights this performance safety tradeoff.



Autonomous Driving: Why racing ?

1

Detecting the vehicle limits



2

Decision making at the vehicle limits



3

Handling at the vehicle limits



Unstructured Environment
Different Tracks and Conditions
Different vehicle setups



High prediction uncertainty
Strategy, Energy, Overtaking



High speeds, High accelerations
High planning horizon necessary
Small reaction times



Autonomous Racing @Hipert/UNIMORE



FITENTH



Roborace



Formula
Student



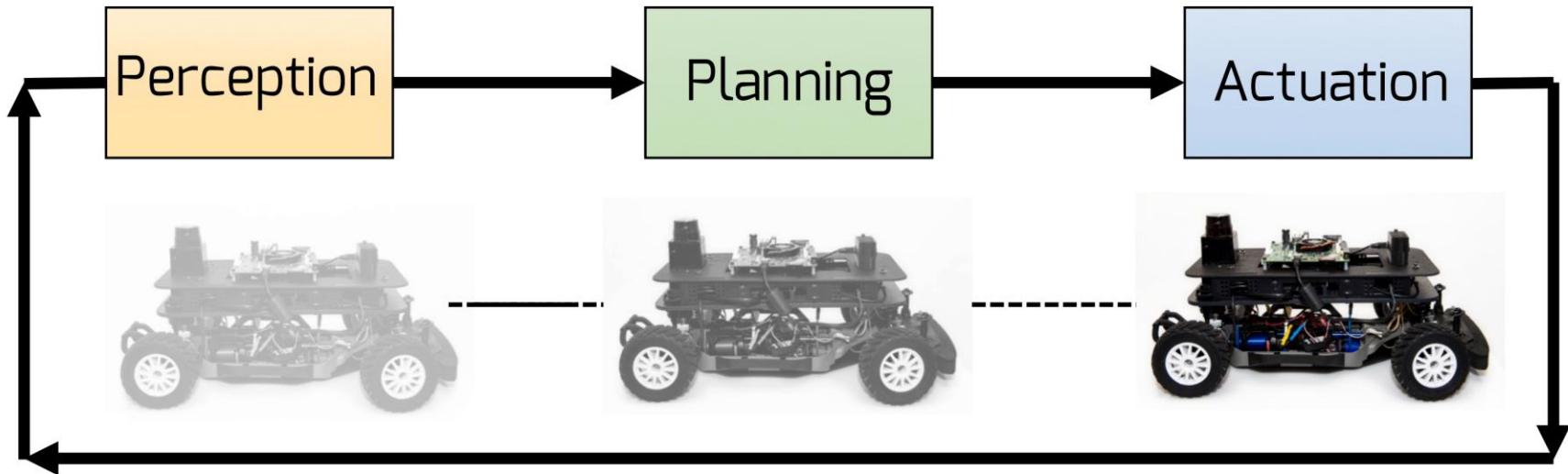
Indy
Autonomous
Challenge



A deeper look....

How do we do AD (at UNIMORE)?

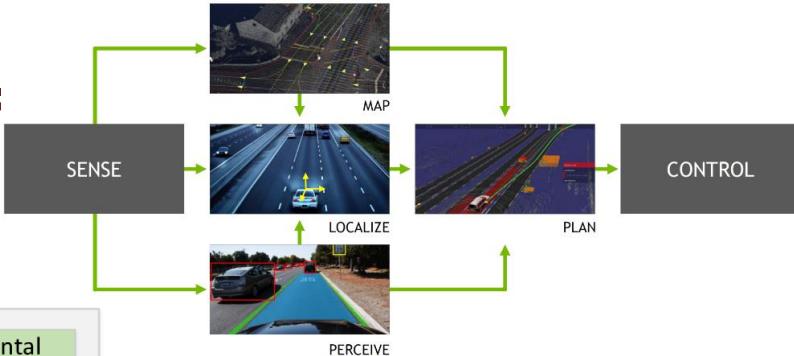
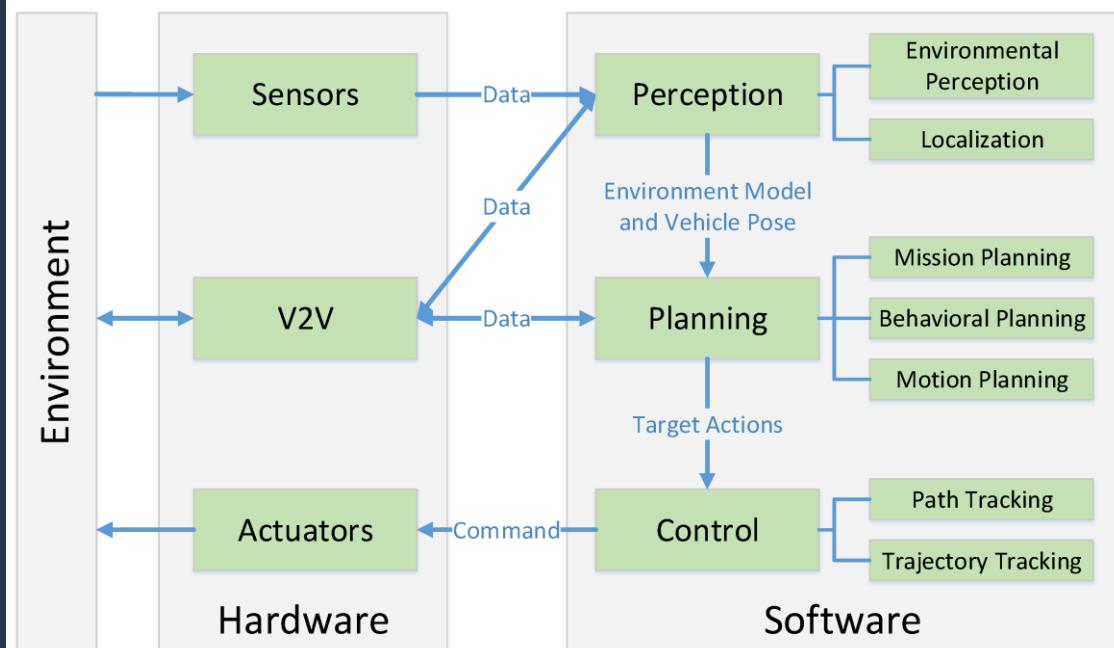




(At least) three stages – F1/10

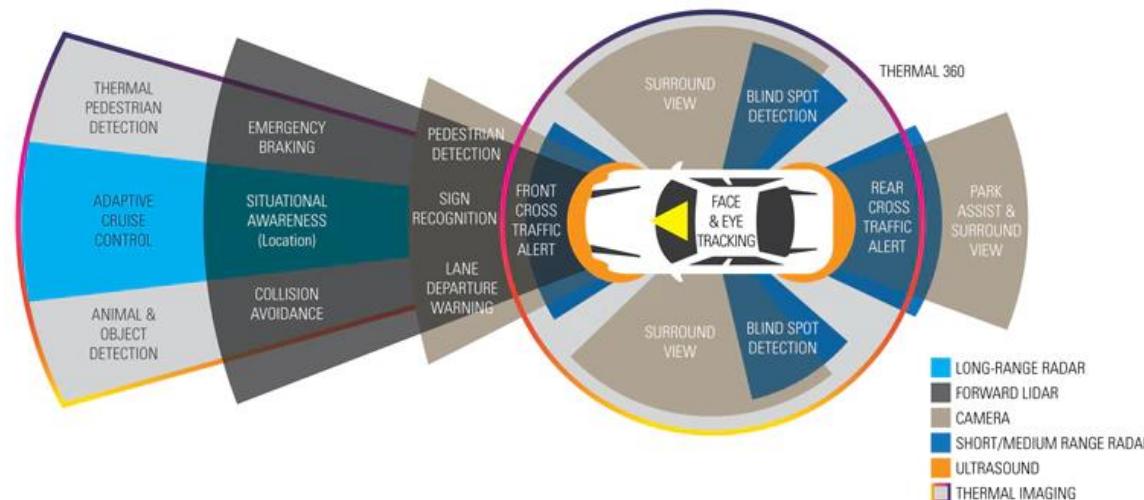


...the tough reality



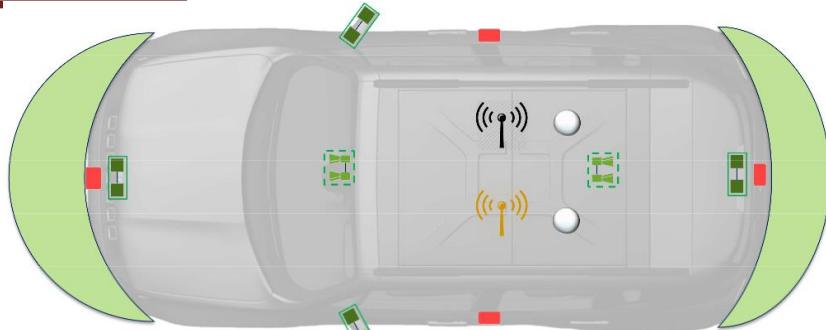
Sensors

- › **Exteroceptive:** capture the external environment
- › **Proprioceptive:** capture of the ego vehicle
- › The number and complexity of the sensors increased dramatically in the last few years



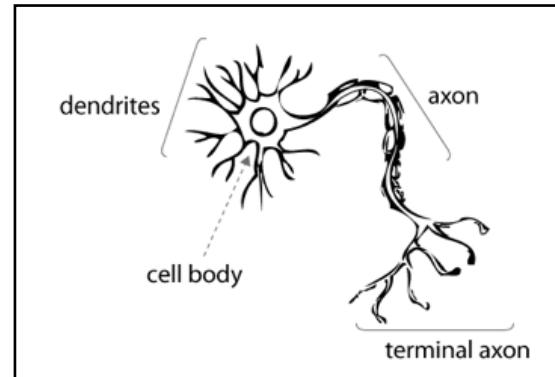
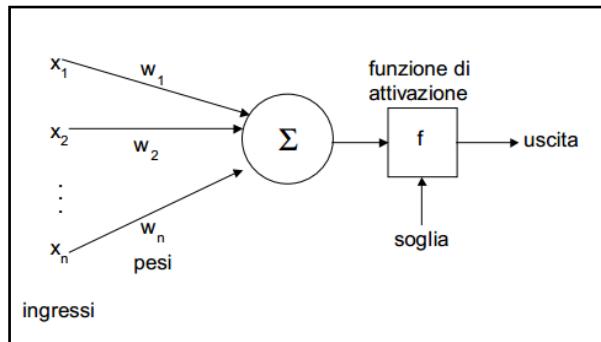
Maserati quattroporte

- › 5 sekonix cameras (GMSL)
- › Ouster OS1 64/128-120m (Ethernet)
- › GPS (CAN)
- › Radar ARS301
- › Multiple ECU (Electronic Control Unit)
 - Pegasus
 - Drive PX2
 - Xavier
 - TX2



Neural networks

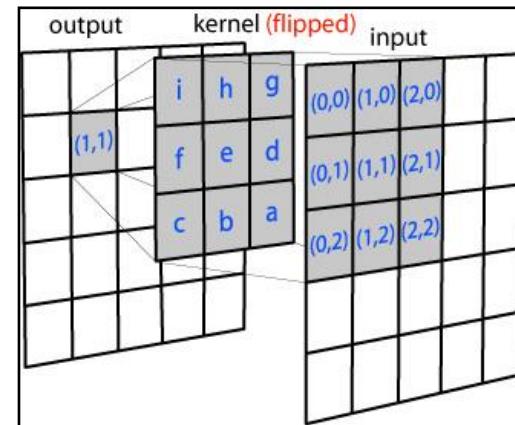
- › Bio-inspired
- › Based on neurons arranged in layers
 - And sub-layers
- › Convolutional neural network
 - Perform Convolutions



Convolution

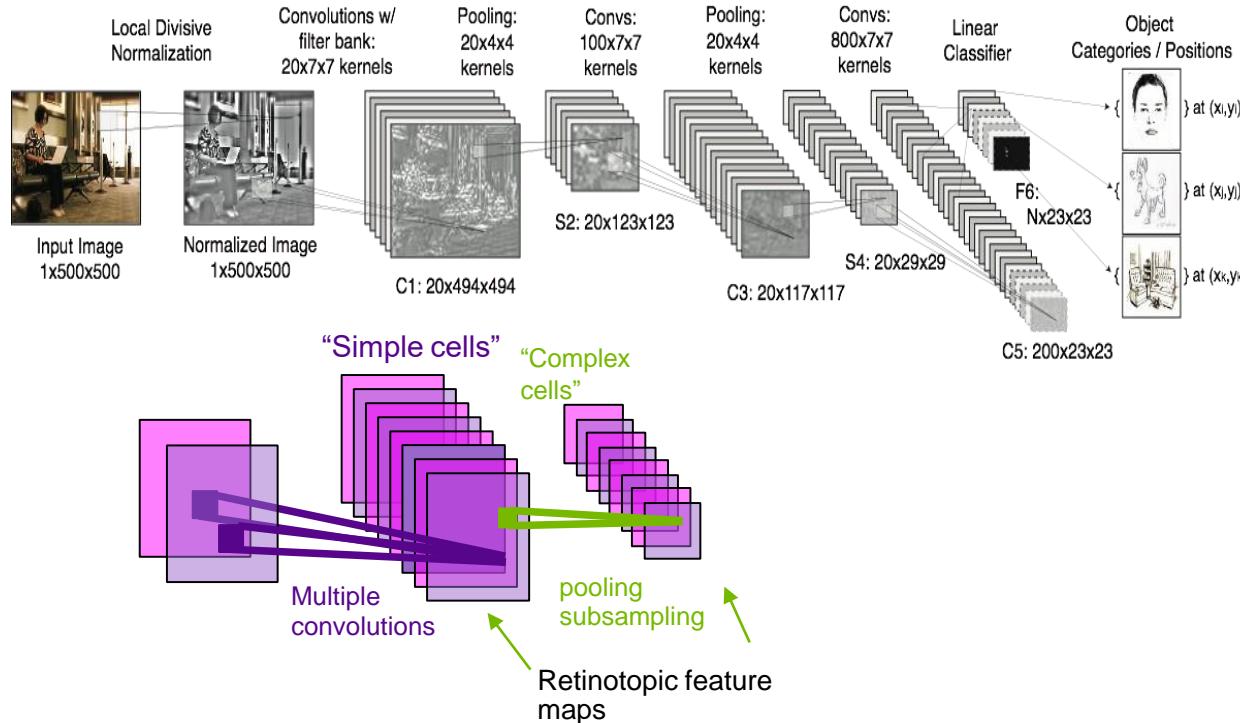
- › Computation-intensive
- › Suitable for implementation in hardware
- › In computer vision, blurring

$$\begin{aligned}(f * g)(t) &\stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau \\ &= \int_{-\infty}^{\infty} f(t - \tau)g(\tau) d\tau.\end{aligned}$$

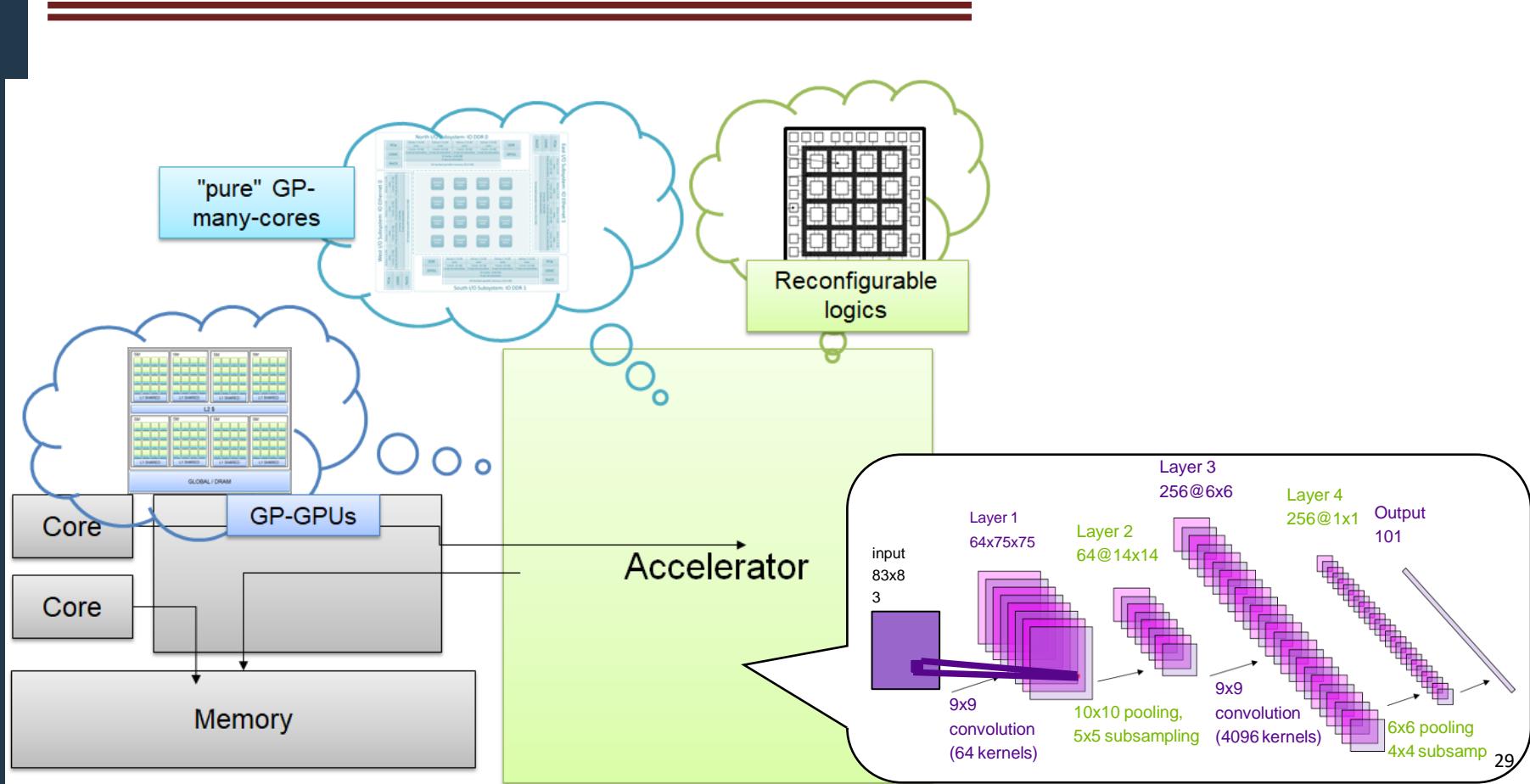


The convolutional net model

› (Multistage Hubel-Wiesel system)



The convolutional net model (cont'd)



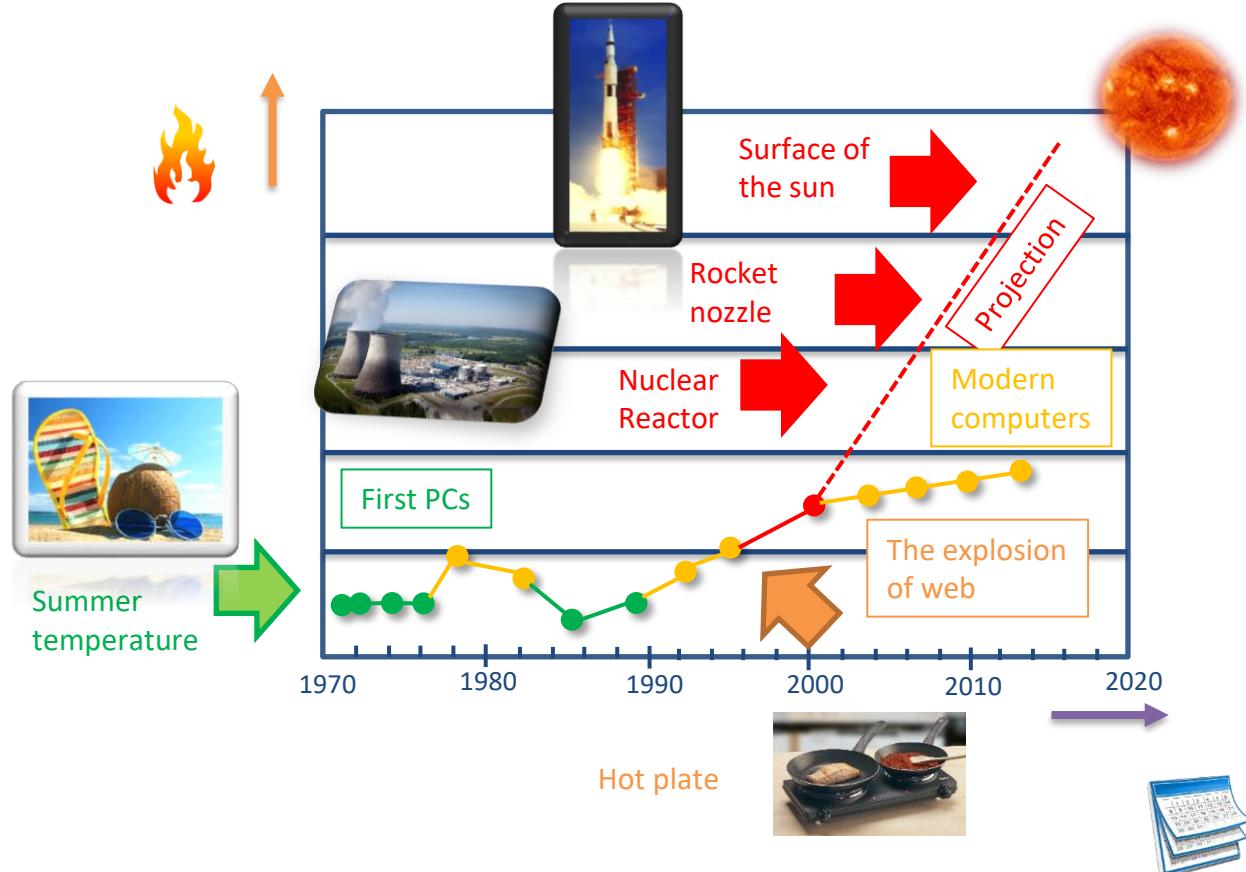


"What's the problem?"





I'm hot, baby!

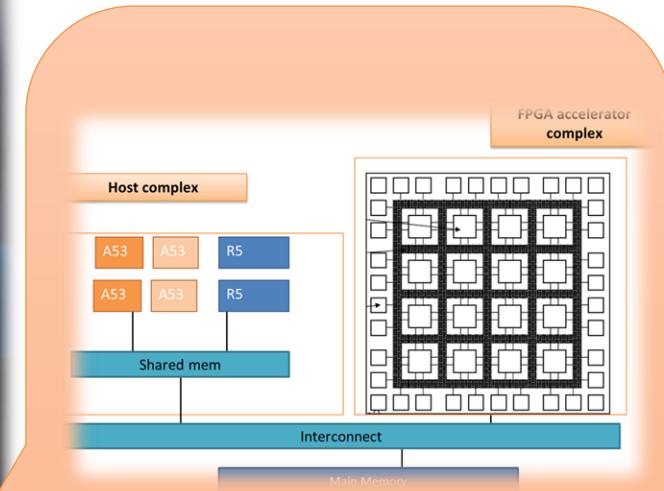
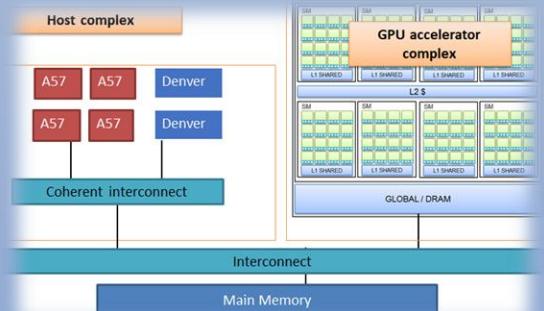




Embedded systems



GPGPU SKY HARD



W LA FPGA



Real-Time Embedded Systems

Aka: Why we're all here..

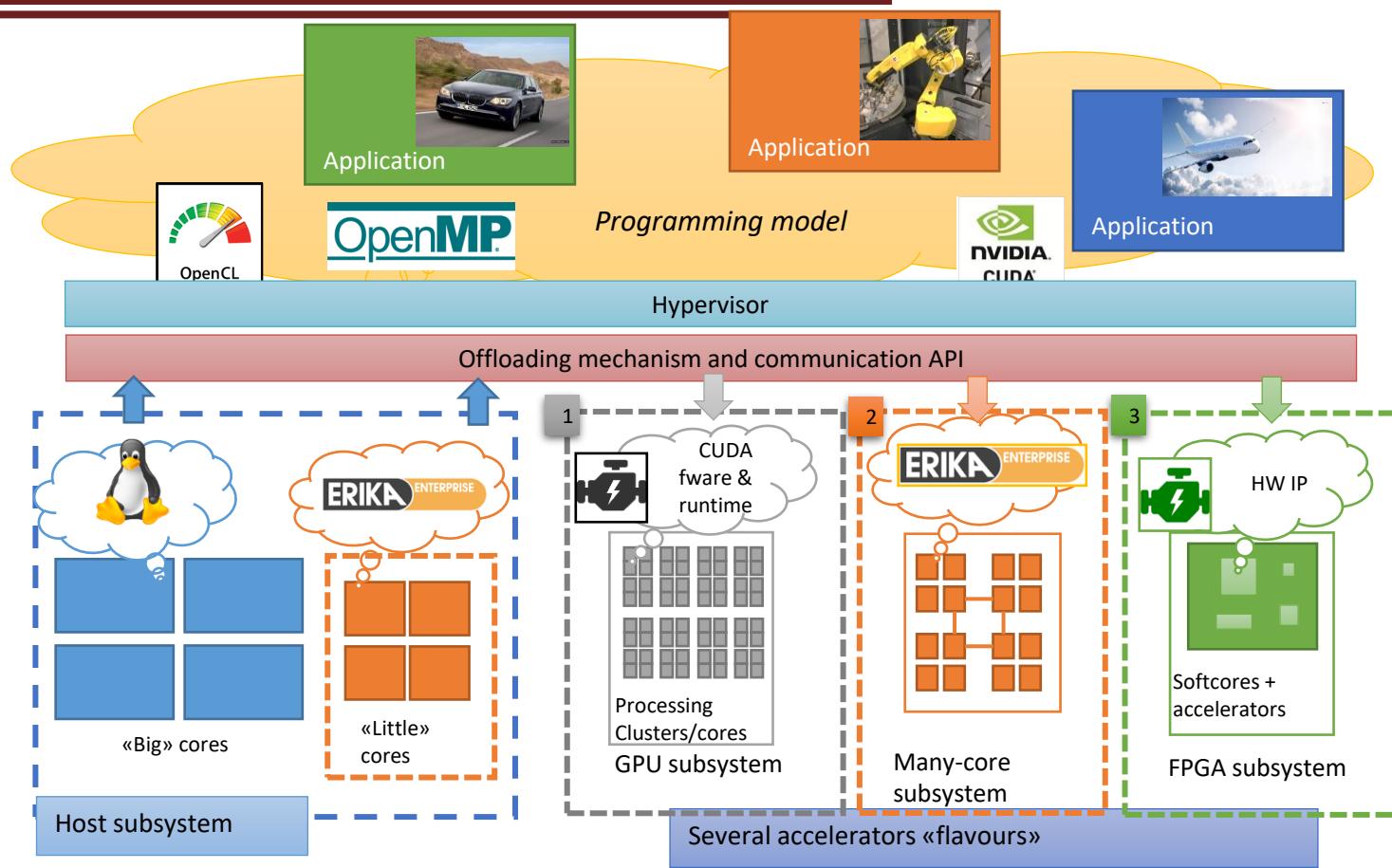
Cars are constrained!

- › Size, Weight and Power – SWaP
- › High workload (e.g, Machine Learning), require high-performance and power efficiency
- › Example: How does it take to turn on MR-23DL ("Diletta")?

Safety critical systems require predictable performance

- › At all the level of the software stack
- › Not only application...but also RTOSes, hypervisors...

An example: the Hercules stack



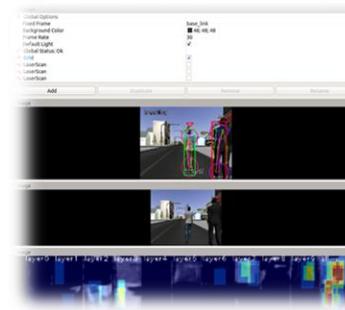


How do I become a professional AD engineer?

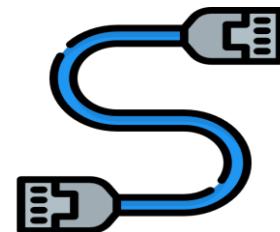
Can I learn it?

First grade - simulators

- GTA-V
- Assetto Corsa
- Carla
- LGVSL
- ...

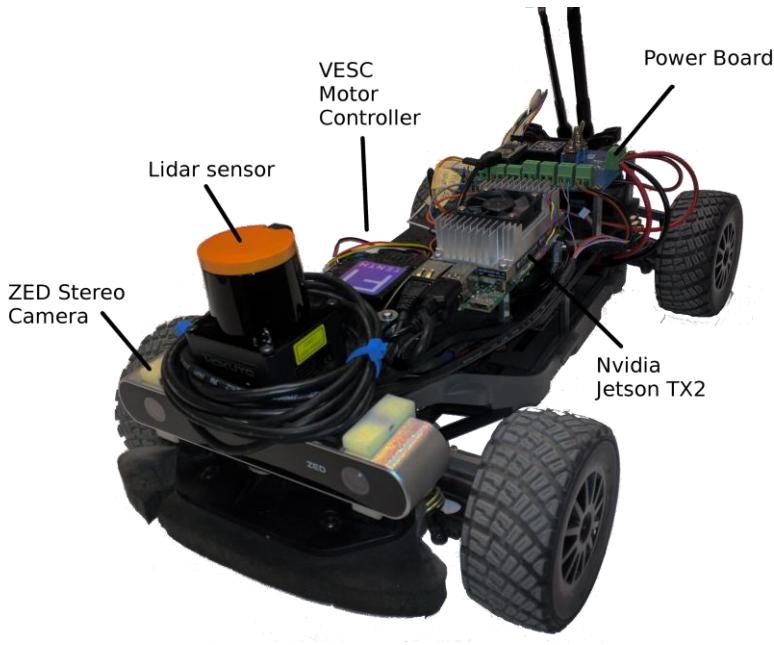


- Python
- C++
- Matlab
- ...



(Optional)
On-board computer
Hardware-in-the-Loop

Second grade - the F1/10 challenge



A scale car

- › Instructions are provided
- › Challenge is on software
- › 2-3 Challenges per year, span across the world

Less than 3k€ to build the car

› Join the race!



Third grade - the real car

For students

- › Professors are just advisors; teams are 100% self-managed (even fundings!)

Very good for CV and team/soft skills

- › Part of our BS/MS courses

Build a fully autonomous racing car

- › Not only engineers...a full team, with marketing, management...
- › Competition also on project only...no need to race or build the car



I can get no-satisfaction



M23-DL

TECHNICAL DATA

CHASSIS: Carbon fiber monocoque
WEIGHT: you don't ask to ladies!
ENGINE: 600 cc, 4 cylinders in line NA, longitudinal
POWER: 80 CV
WIRING: + 1 km of cables
AUTONOMOUS DRIVING: 5 actuations

A.I.
2 Embedded Computers
4 Control Units

DOUBLE DRIVING MODE
Manual & Autonomous

STEERING ACTUATOR

LIDAR

Sensor to detect the position of the cones

BRAKE ACTUATOR

CAMERA
Sensor to detect the colours of the cones

THROTTLE ACTUATOR

CLUTCH ACTUATOR

GEARSHIFT ACTUATOR

ENGINE

IMU
Sensors to detect the movements of the vehicle and its position in the track

PHONIC WHEELS
Sensors to detect the speed of the vehicle



The Team 2021-22



Formula ATA, Varano De' Melegari (PR), July 2022



We want you!

- › We Make Future fair, UNIMORE vs. UNIBO
 - <https://en.wemakefuture.it/>
 - 15-16-17 June 2023

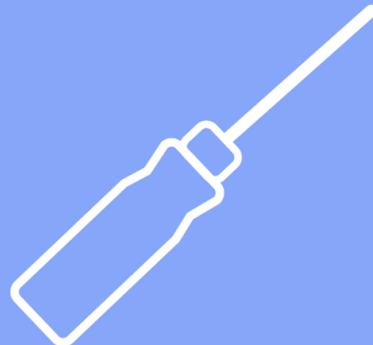


The image shows the header of the We Make Future website. It features a red circular logo with 'WTF!' in white. To the right, the text 'We Make Future' is displayed in large white letters, followed by 'International Trade Fair and Festival on Tech and Digital Innovation' and 'JUNE 15-16-17, 2023 / RIMINI EXPO CENTRE'. Below this, a button says 'Full Ticket at 149€+VAT offer ends on April 20 BUY NOW'.

We need 1-2 persons to run the race/show...

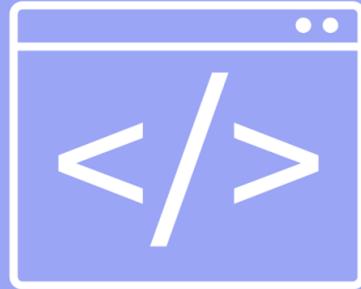


F1/10: 1/10 the size, ~~10x the fun!~~ (1/10 the problems)



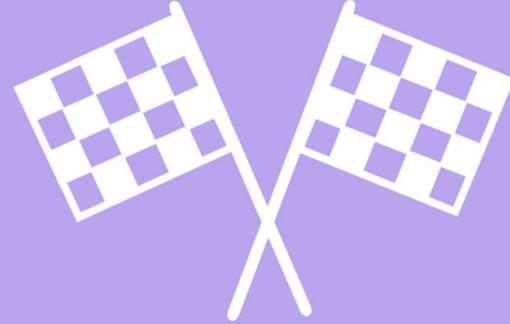
BUILD

Construct your vehicle
using our starter kit.



CODE

Learn to drive your vehicle
autonomously.

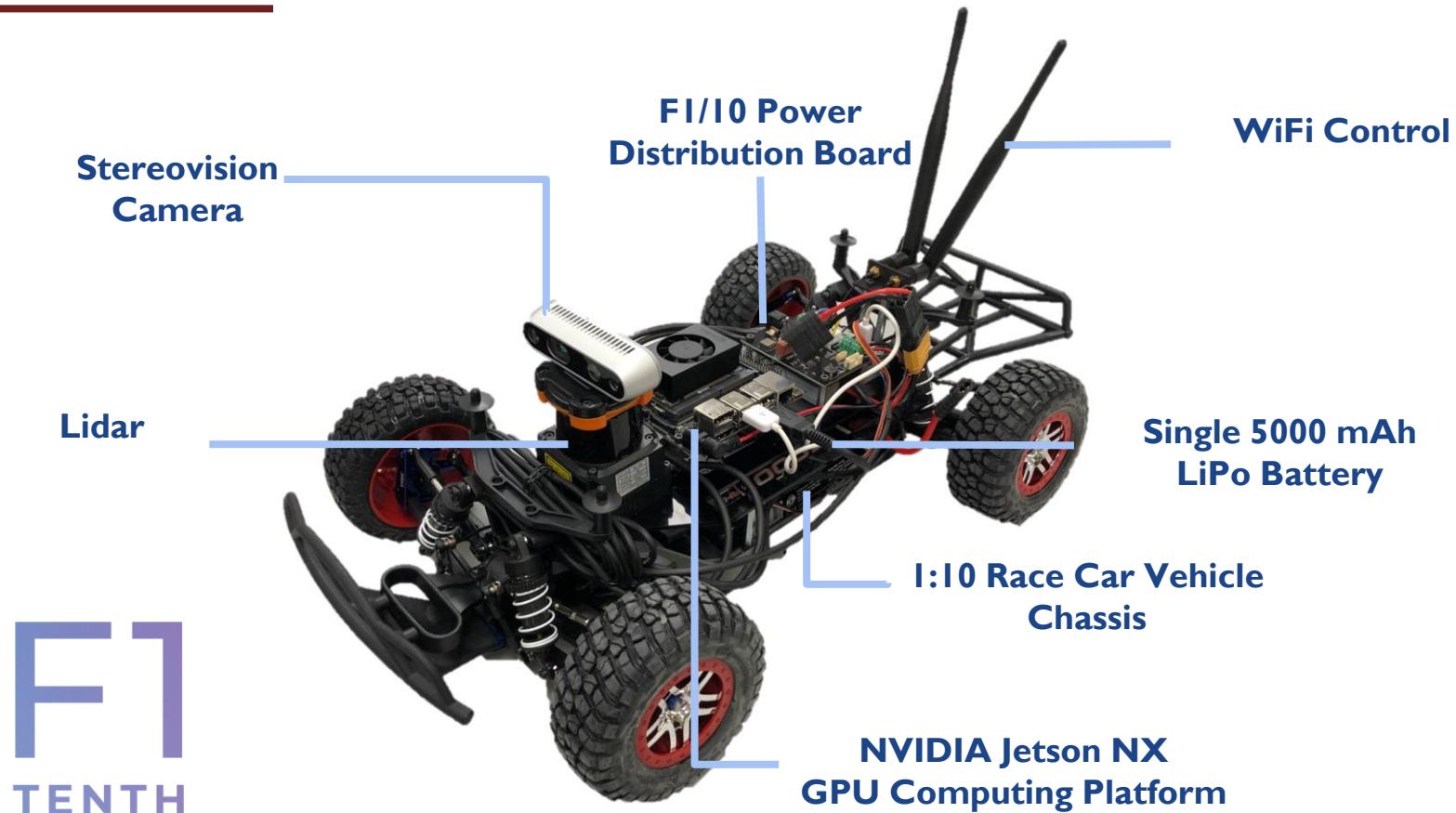


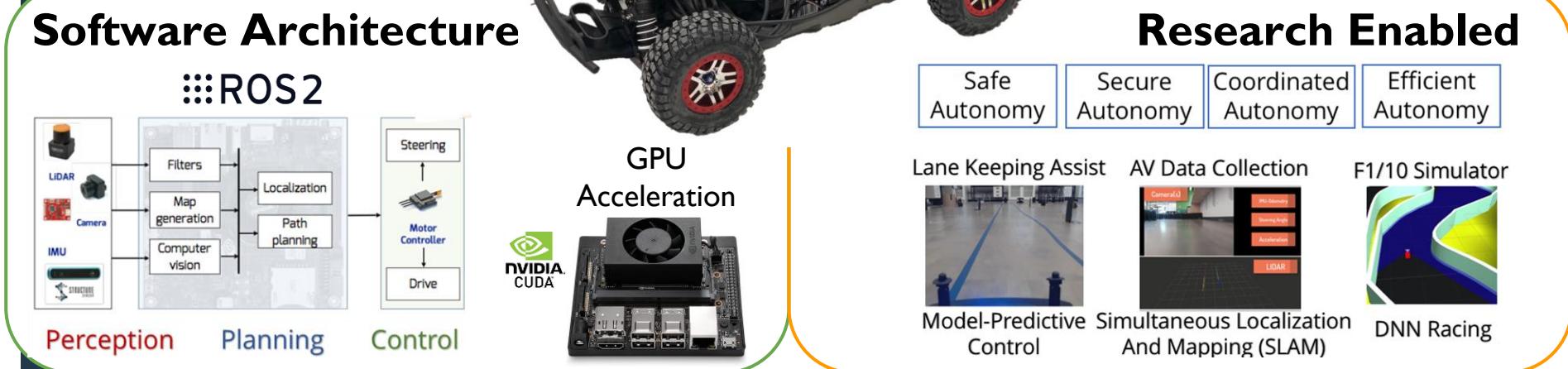
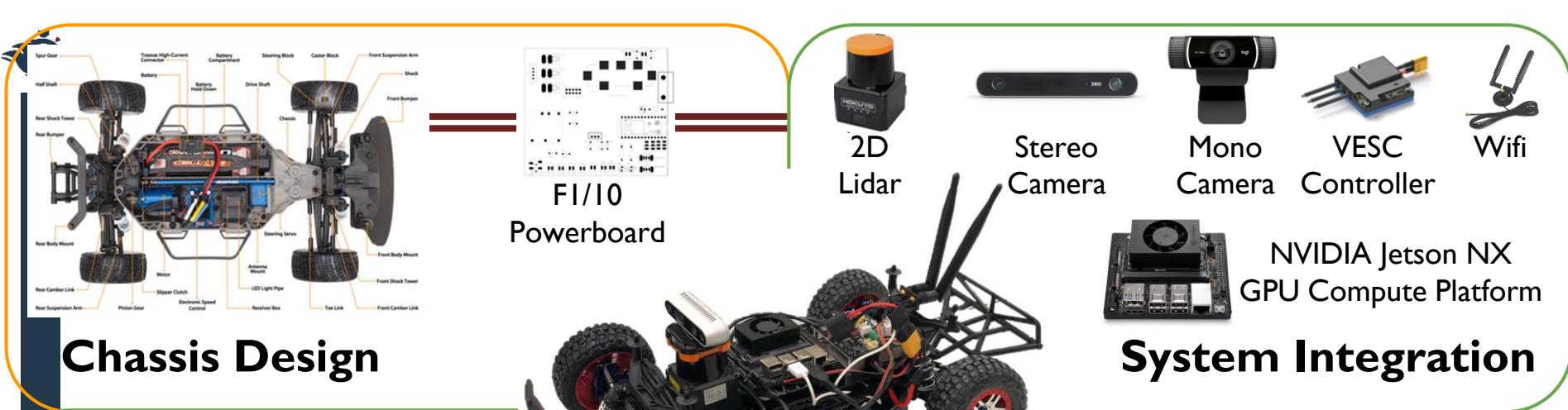
RACE

Register to Compete

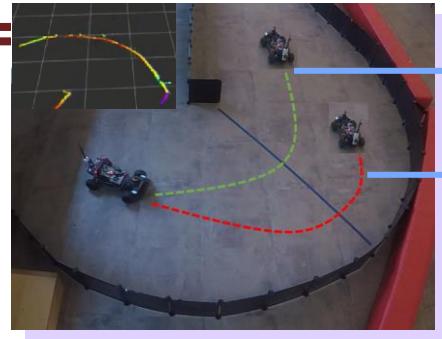


Low-Cost Open-Source Platform



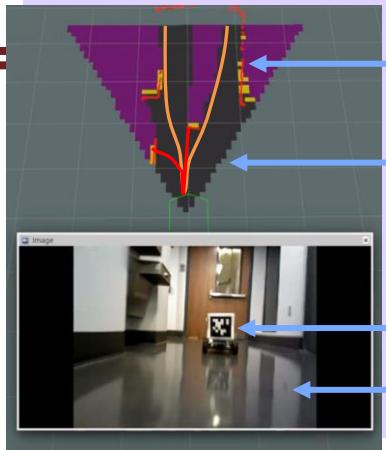


FOLLOW THE GAP METHOD

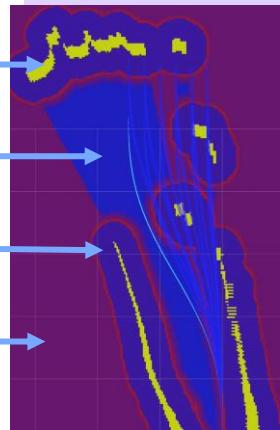


Follow the gap method
Simple obstacle avoidance

MODEL PREDICTIVE CONTROL



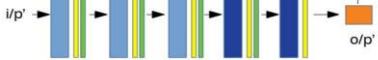
Generated paths
Max. cost (obstacle)
Drivable region
Drivable region
Medium cost
Unexplored space



END-TO-END DNN

F1
TENTH

FPV data annotation

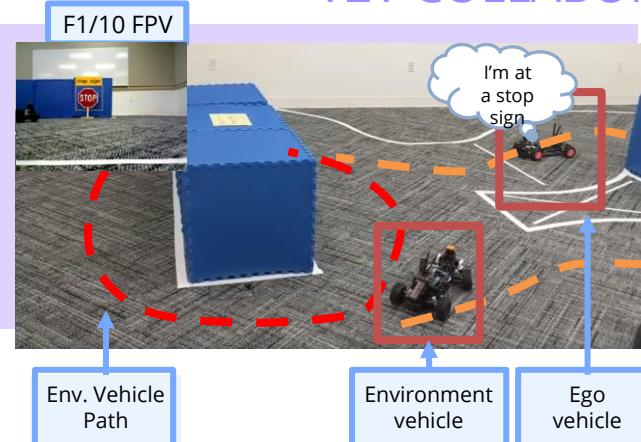


Conv 5x5,stride=2
Conv 3x3
BatchNorm
ReLU
RNN Cell

Predicted steering (blue)
Ground Truth (green)

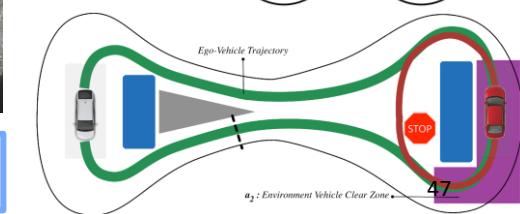


V2V COLLABORATION



Env. Vehicle Path
Environment vehicle
Ego vehicle

Collaborative behavior automaton

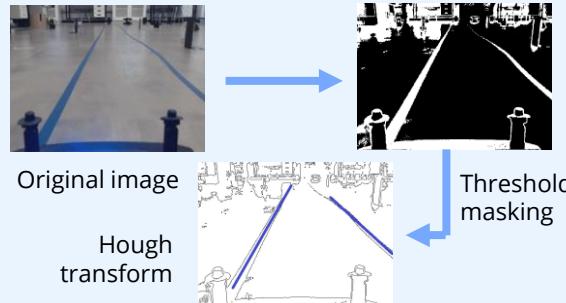


a_2 : Environment Vehicle Clear Zone

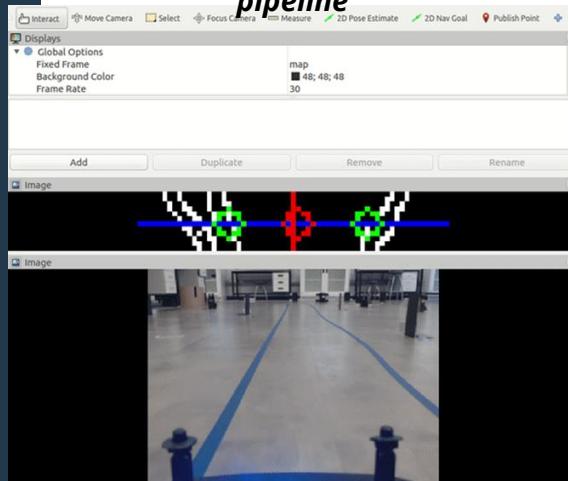
47



LANE KEEPING ASSIST

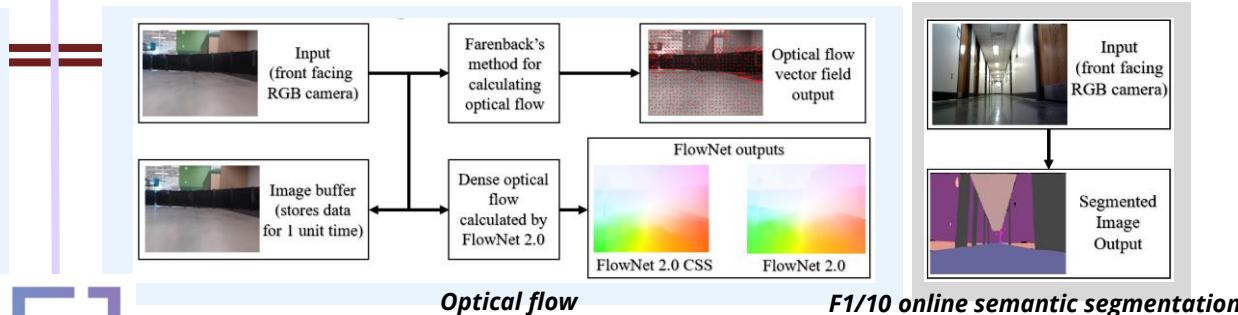


Lane detection pipeline



rviz visualization

COMPUTER VISION EXAMPLE



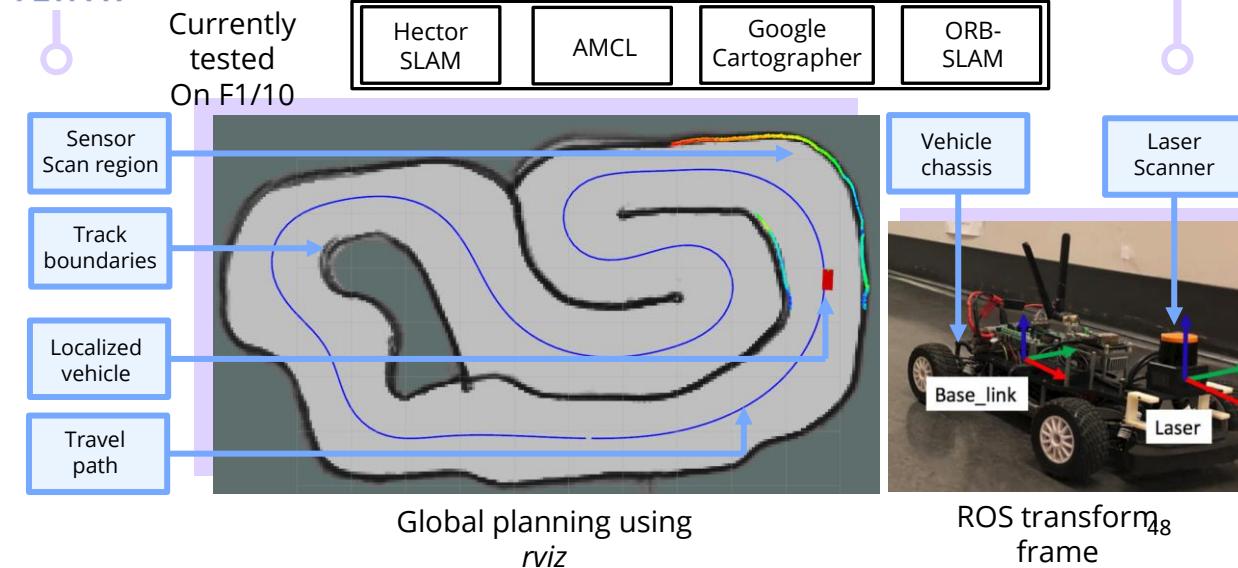
F1
TENTH

Currently tested
On F1/10

Optical flow

F1/10 online semantic segmentation

LOCALIZATION AND MAPPING

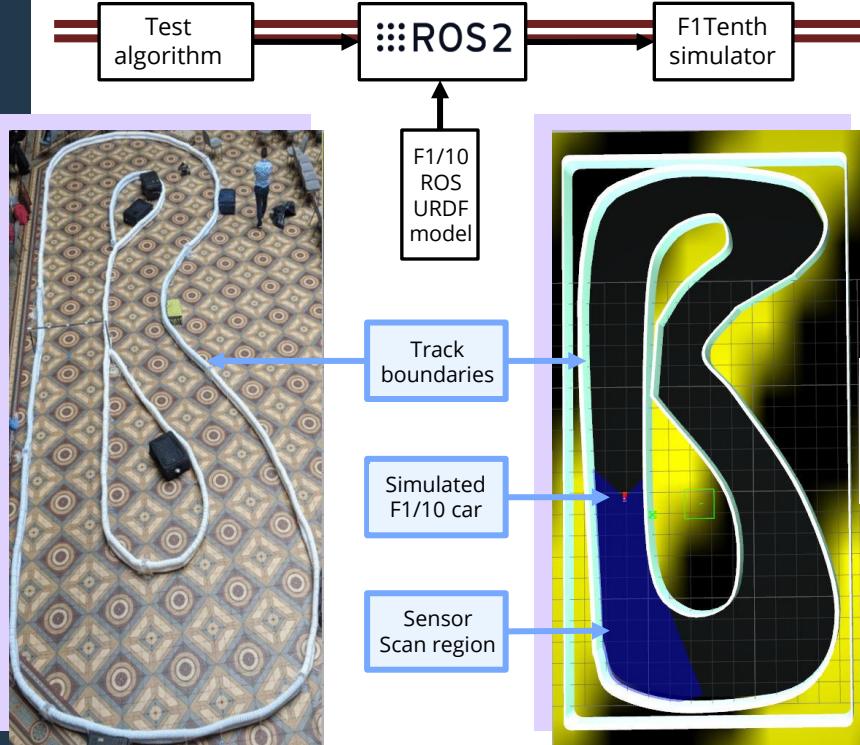


**Global planning using
rviz**

**ROS transform
frame**



F1TENTH SIMULATORS

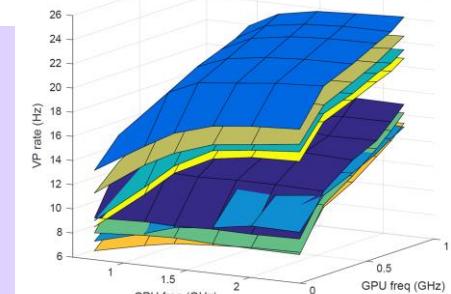
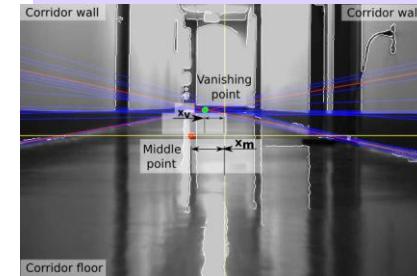


ROSbag logs
LiDAR and
Odometry data

Porto track built using
ROSbag data

REAL TIME SCHEDULING

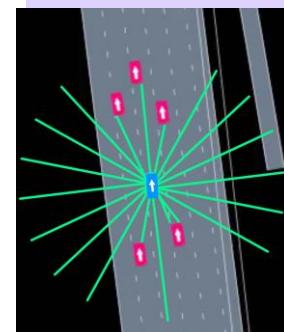
Vanishing point (VP) algorithm
implemented on F1/10



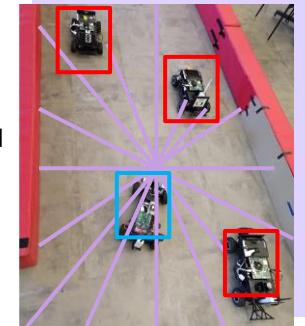
VP throughput based on
CPU & GPU frequencies

F1
TENTH

TESTING & VERIFICATION



GAIL model: Outputs include mean &
variance of steering & throttle



Data captured
from traffic on I-80

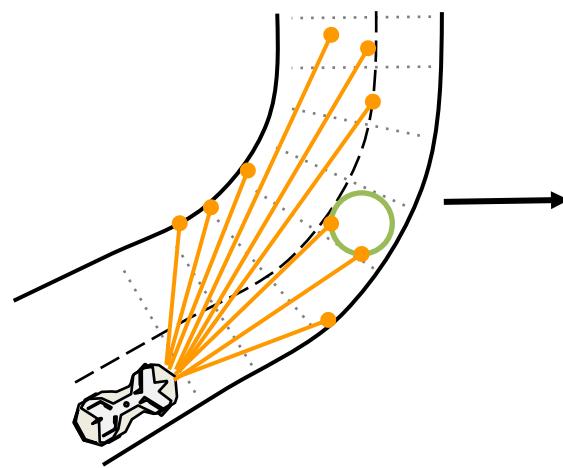
F1/10 helps recreate traffic
scenarios to verify model



Lesson(s) Plan

Part I: Learn to Drive

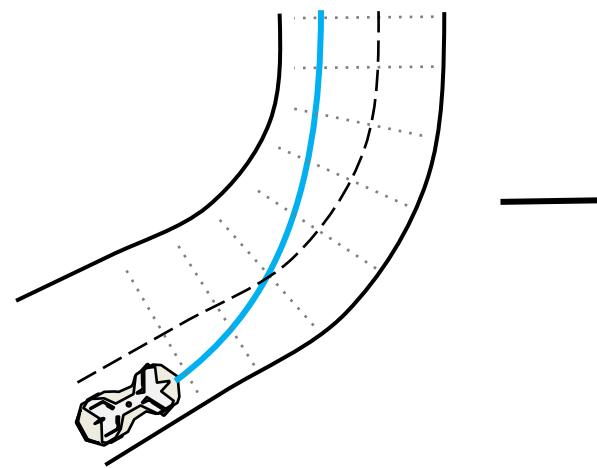
Autonomous Driving w/ FITENTH
Avoid the crash - Reactive Planning



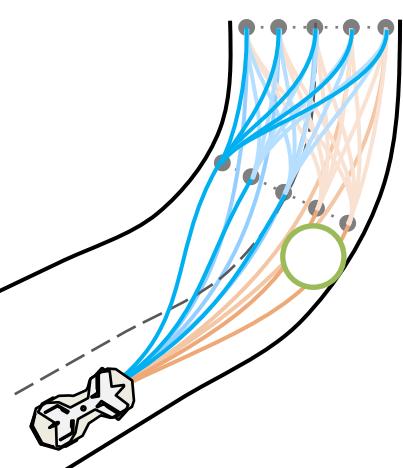
I. Follow The Gap

Part II: Learn to Race

Follow the Raceline - Pure Pursuit
Race & Overtake - Graph Planner



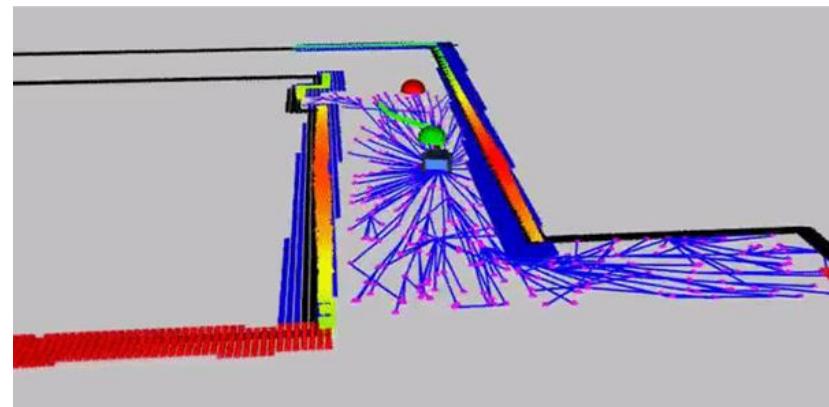
2. Follow The Raceline



3. Race

FITENTH Gym:

- Lightweight 2D simulator built in Python
- Asynchronous
- Faster than real-time execution (30x realtime)
- Realistic vehicle simulation and collision
- Runs multiple vehicle instances
- Publishes laser scan and odometry data
- Built for fast prototyping



List of assignments

...we'll go through them...



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

High Performance
Real Time Lab



Race 1: Reactive Methods

Race Format: Time attack, single car

Penalties: Crashing

Baseline: Complete 5 laps without crashing

Example Video: Follow the Gap in CPSWeek Grand Prix'18





Pose Representation and Transforms

Each sensor provides measurements in the frame of reference specific to that sensor



Pose Representation and Transforms

Challenge:

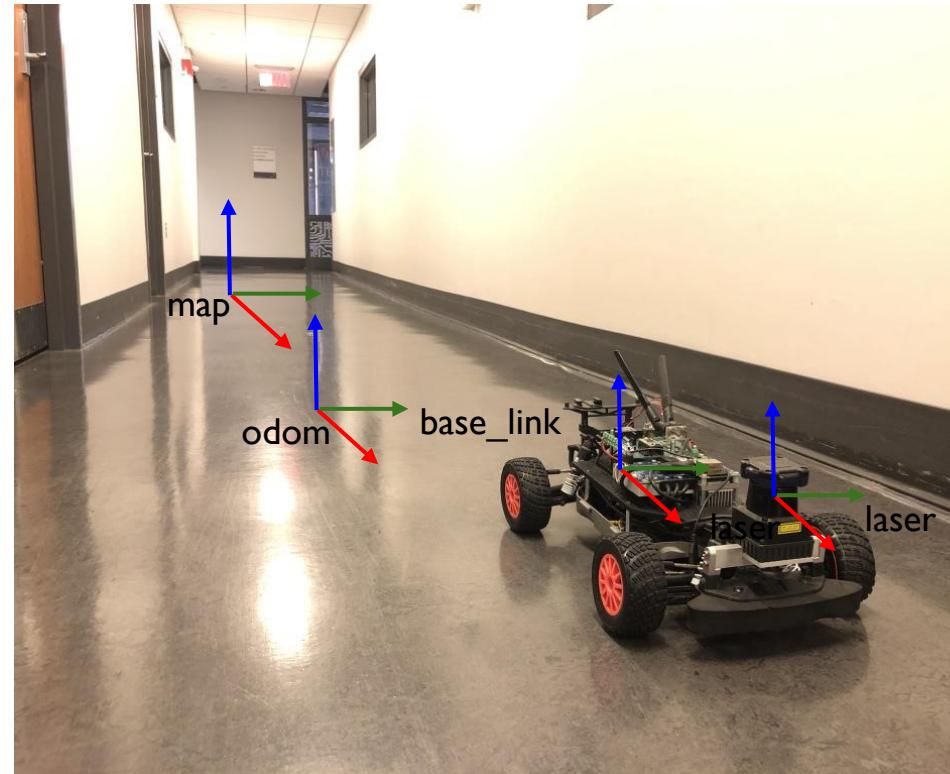
Sensors in different reference frames

Learning outcome:

Coordinate frames, Rigid body transforms

Assignment:

Pose transformations in ROS





Wall-following

Challenge:

How can we drive the car around the track

Learning Outcome:

Basics of PID, how to compute error, failure modes.

Assignment:

Wall following in simulation and on the vehicle.





Obstacle Avoidance: Follow the Gap

Challenge:

How can we avoid obstacles

Learning Outcome:

Basics of reactive navigation,
avoidance on both static and
dynamic obstacles

Assignment:

Follow the gap in simulation and on
the vehicle.





Obstacle Avoidance: Follow the Gap

Challenge:

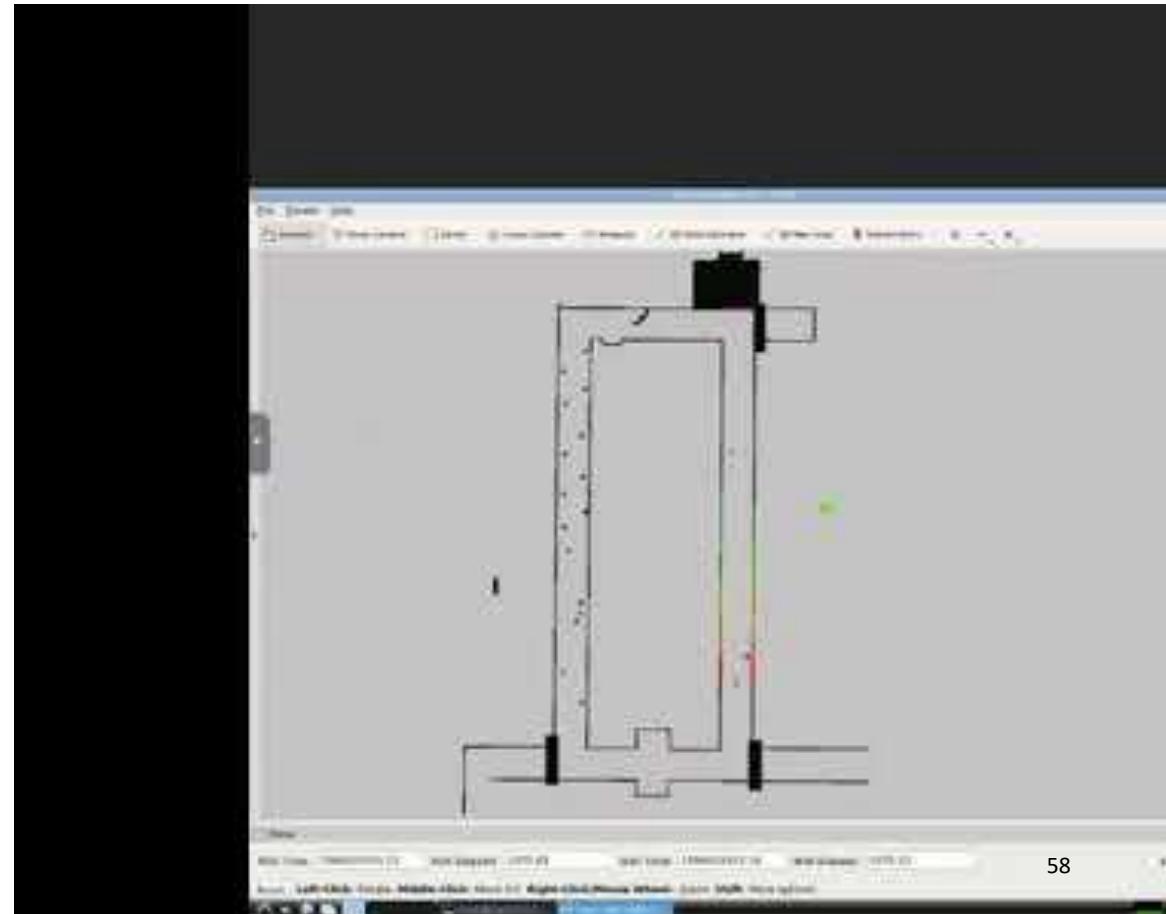
How can we avoid obstacles

Learning Outcome:

Basics of reactive navigation,
avoidance on both static and
dynamic obstacles

Assignment:

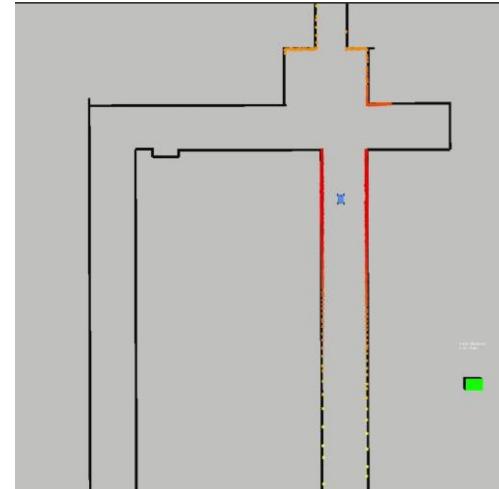
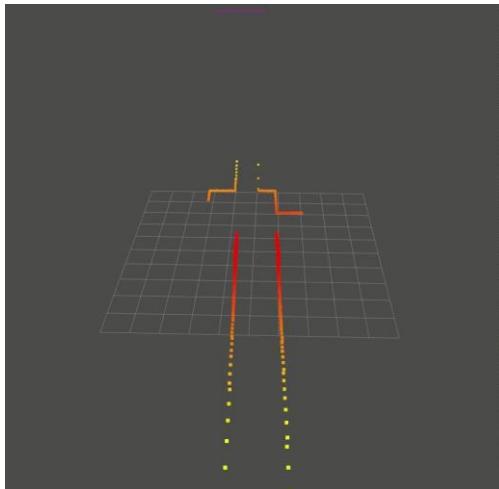
Follow the gap in simulation
and on the vehicle.



Multiple Reference Frames

The world makes more sense if we put the laser scans in the **global map frame** instead of the **laser frame**.

More information available when planning in a **global frame** instead of extracting information in the observation frame.





Race 2: Map-based Methods

Race Format:

Time attack, single car

Penalties:

Crashing

Baseline:

Complete 5 laps without crashing

Example Video:

Pure Pursuit CPSWeek 2019



Localization: Scan Matching

Challenge:

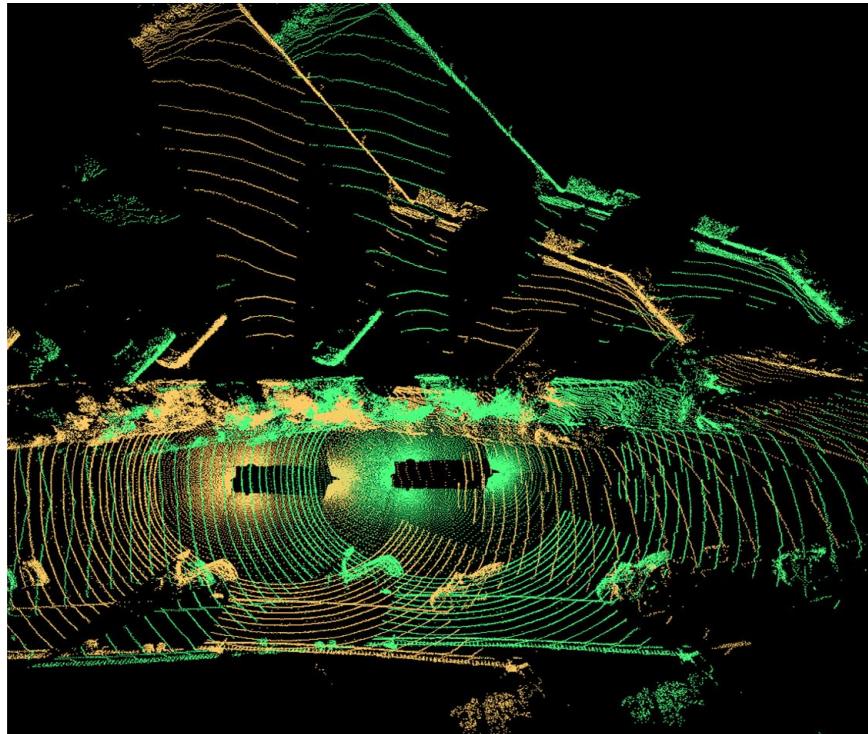
Where is the robot with respect to the previous frame

Learning Outcome:

Iterative closest point algorithm,
implementing a real research paper

Assignment:

Scan matching using iterative closest
point in the simulator

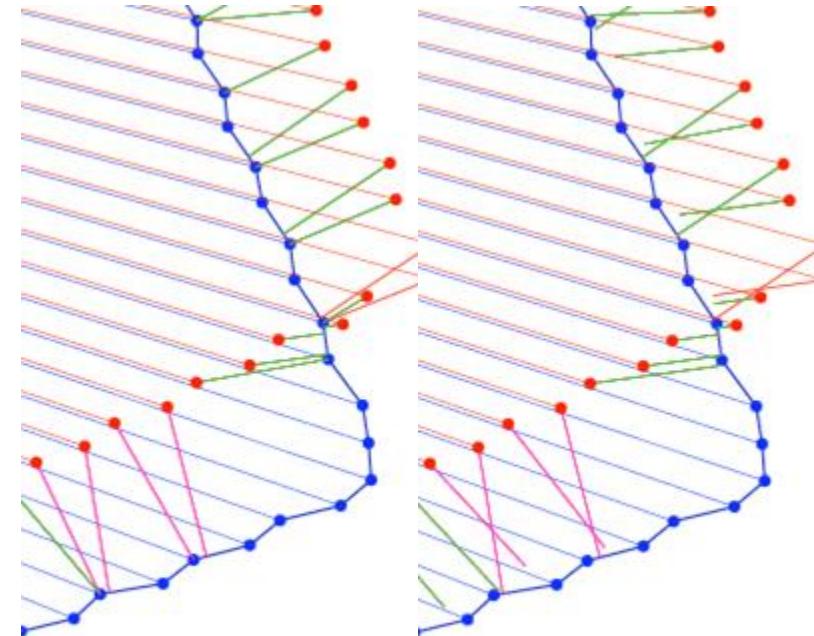


Localization: Scan Matching

Scan matching is a fundamental localization algorithm, and is used in most of the modern SLAM algorithms.

Students implement different metrics (point-to-point vs. point-to-line, shown on the right) for ICP from a research paper.

Students implement fast correspondence search to make ICP practical for a moving robot.





Simultaneous Localization and Mapping with Cartographer

Challenge:

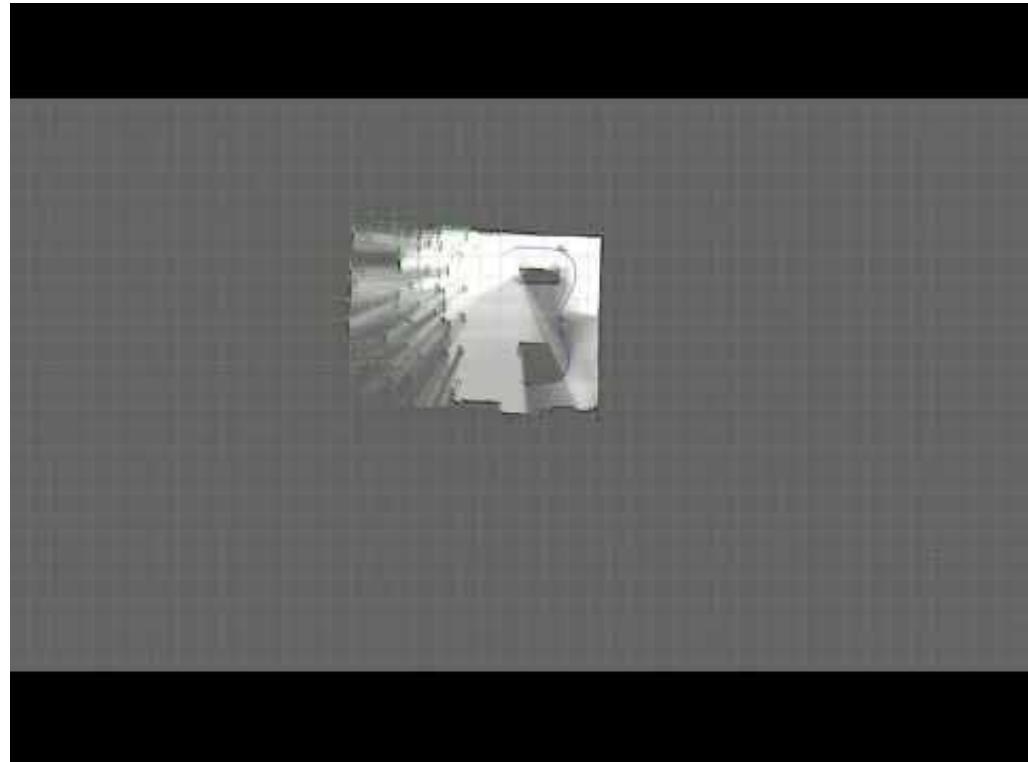
How to use state-of-the-art tools for map building.

Learning Outcome:

Understanding the Cartographer paper and how it relates to scan matching.

Assignment:

Build maps with Cartographer of race track on the car.





Localization: Particle Filter

Challenge:

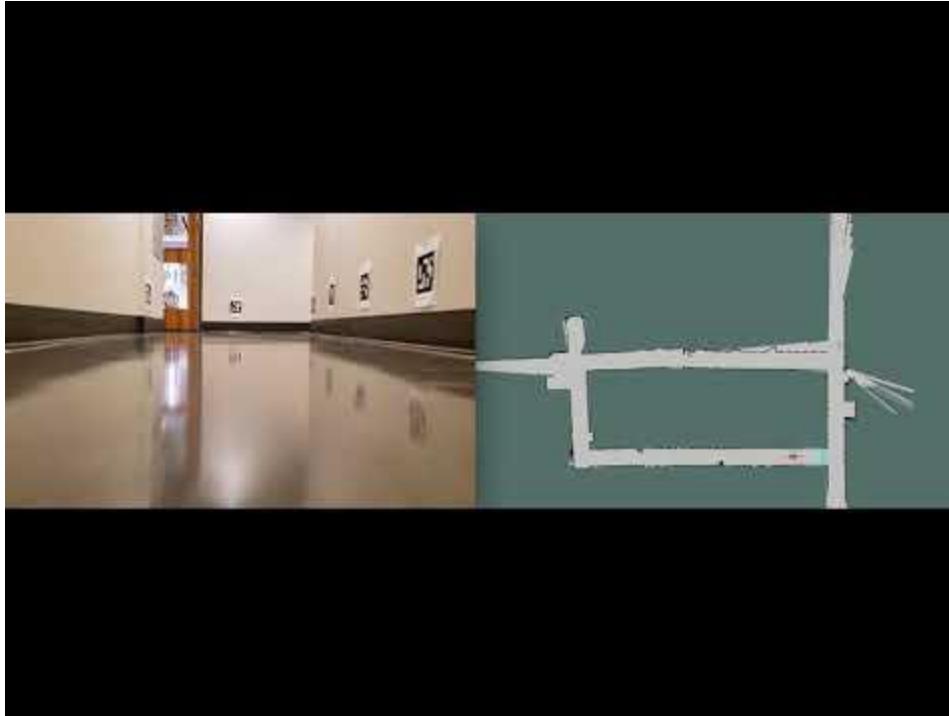
Given a map of the world and multiple sensor observations, what is the pose of my robot?

Learning Outcome:

Understanding particle filter, which is a version of a bayesian filter

Assignment:

Running Particle Filter to localize in the world





Pure Pursuit

Challenge:

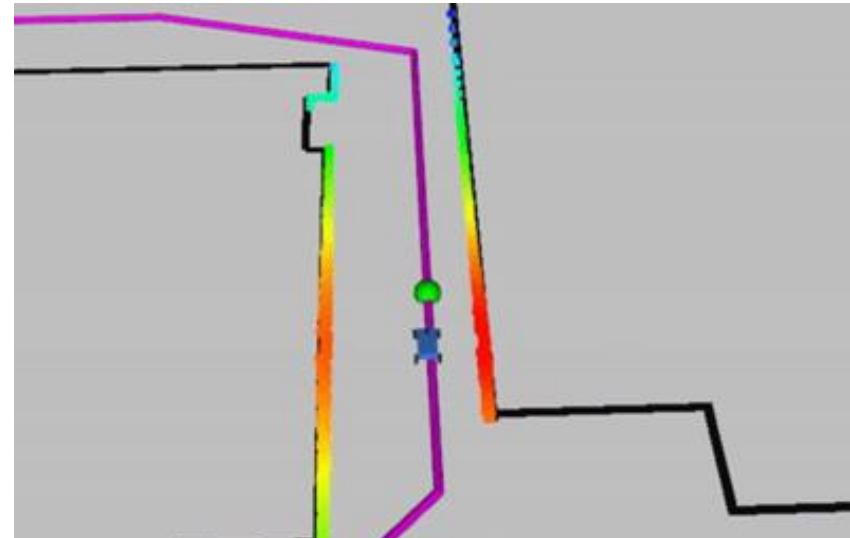
How to track a reference trajectory given a map the ability to localize?

Learning Outcome:

Closed form geometric approach and alternatives.

Assignment:

Implement pure pursuit waypoint tracker in the simulator and on the car.





Pure Pursuit

Challenge:

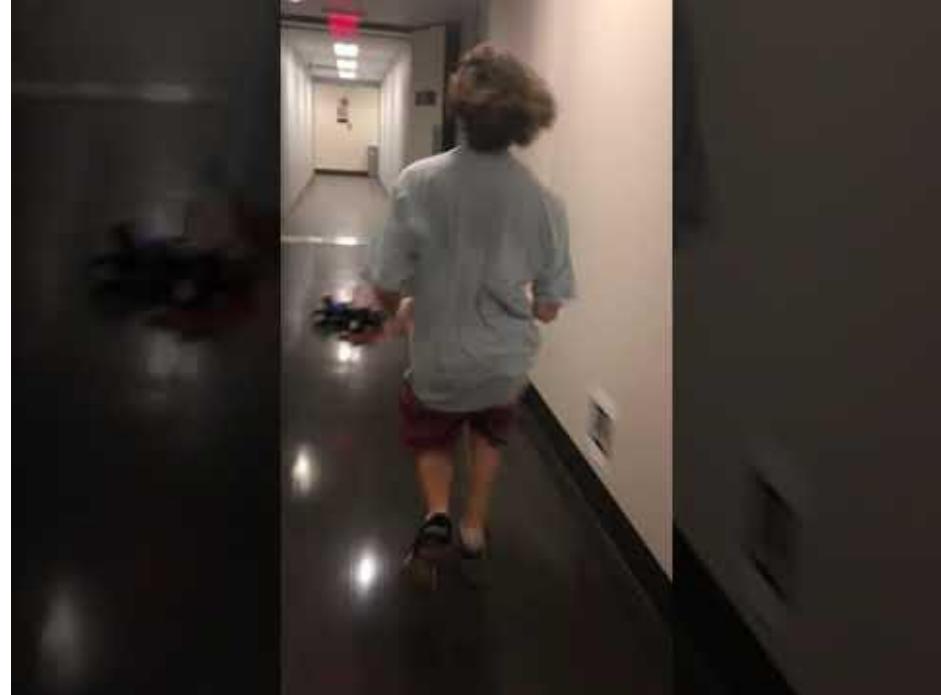
How to track a reference trajectory given a map the ability to localize?

Learning Outcome:

Closed form geometric approach and alternatives.

Assignment:

Implement pure pursuit waypoint tracker in the simulator and on the car.





Motion Planning

Challenge:

How do we combine the capabilities of map based methods while being able to avoid obstacles

Learning Outcome:

Understanding search-based motion planning, probabilistic planning methods, RRT and its variants

Assignment:

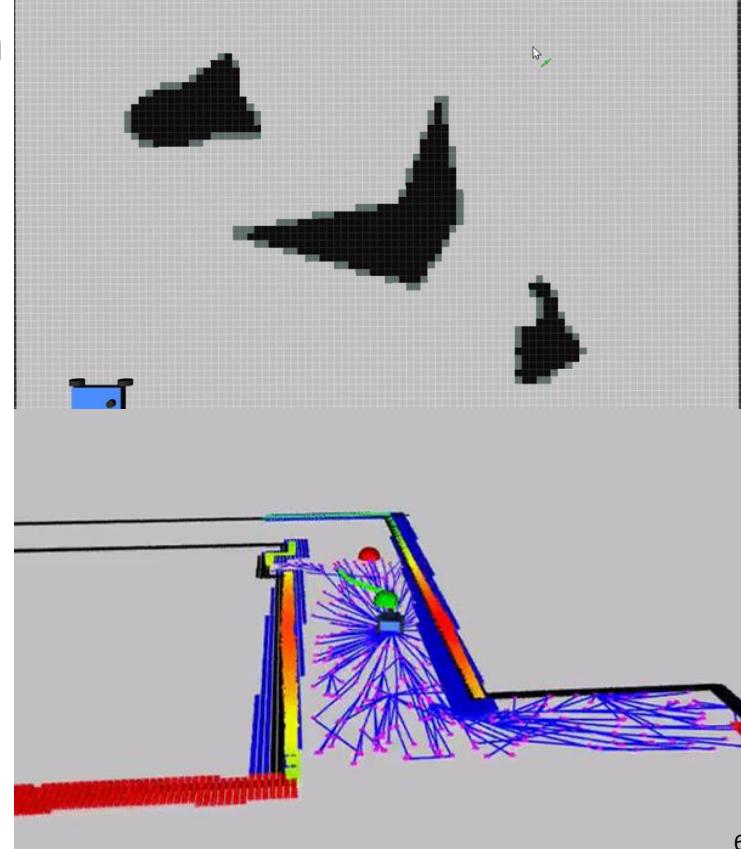
Implement RRT in the simulator and on the car.



Occupancy grid: approximating the real world with a discrete representation, also relates back to the SLAM lecture

Planning in discrete space with search-based planning methods (A*, Dijkstra's)

Planning in continuous space with probabilistic planning methods (RRT, RRT*)





Model Predictive Control

Challenge:

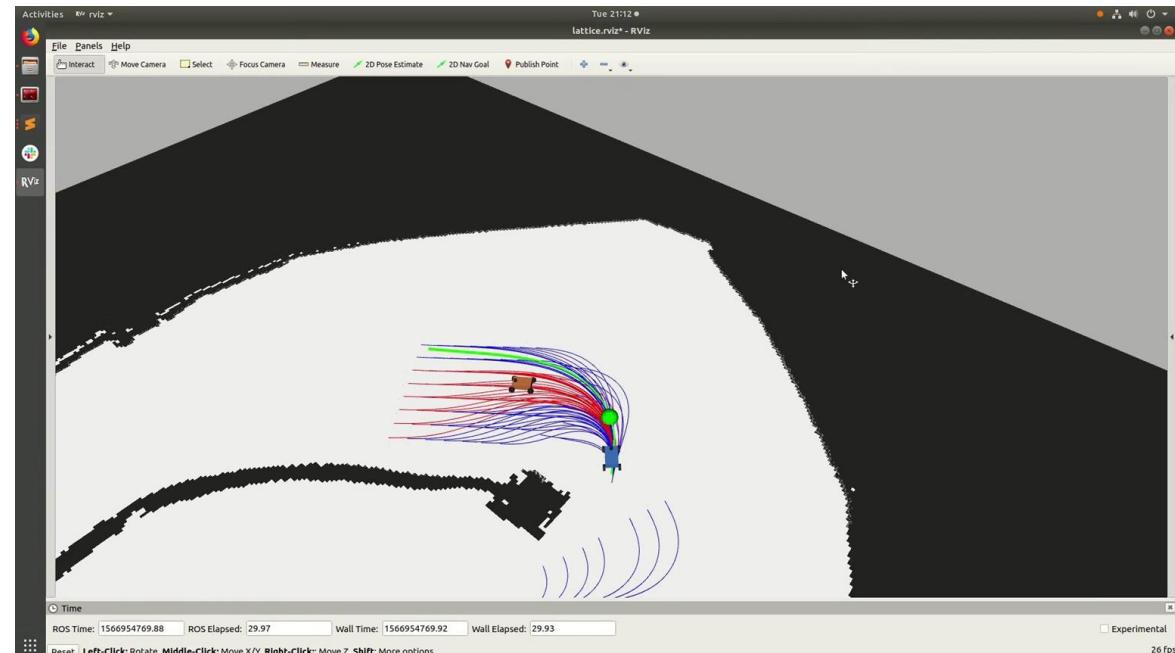
Create dynamically feasible
trajectories for overtaking

Learning Outcome:

Trajectory optimization &
sampling based MPC

Assignment:

As a project option





Race 3: Head-to-Head

Race Format: Round-robin, two cars

Penalties: Crashing

Baseline: Avoid static obstacles

Example Video: Follow-the-gap

CPSWeek'19





Automatic Emergency Braking

Challenge:

Prevent the car from crashing while trying new algorithms.

Learning outcome:

Real-life implementations, sensors, failure modes.

Assignment:

Time-to-collision based braking



Detection and Pose Estimation: classical methods

Challenge:

Where is the other car?

Learning Outcome:

Understanding camera model, single view geometry,
Homography, detecting features, and prediction.

Assignment:

Camera calibration, detecting poses of AprilTags,
predicting the trajectory of adversarial vehicle.



Detection and Pose Estimation: ML-based

Challenge:

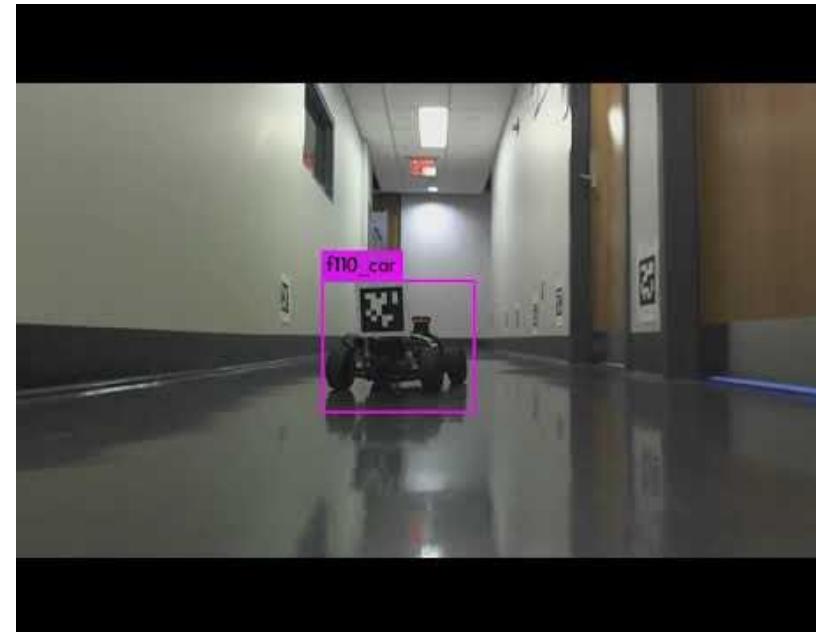
Where is the other car without using fiducial markers?

Learning outcomes:

Understanding multi-view geometry, the epipolar constraint, stereo vision, and using Convolution Neural Network detectors.

Assignment:

Making the detection pipeline fast





Reinforcement Learning

Challenge:

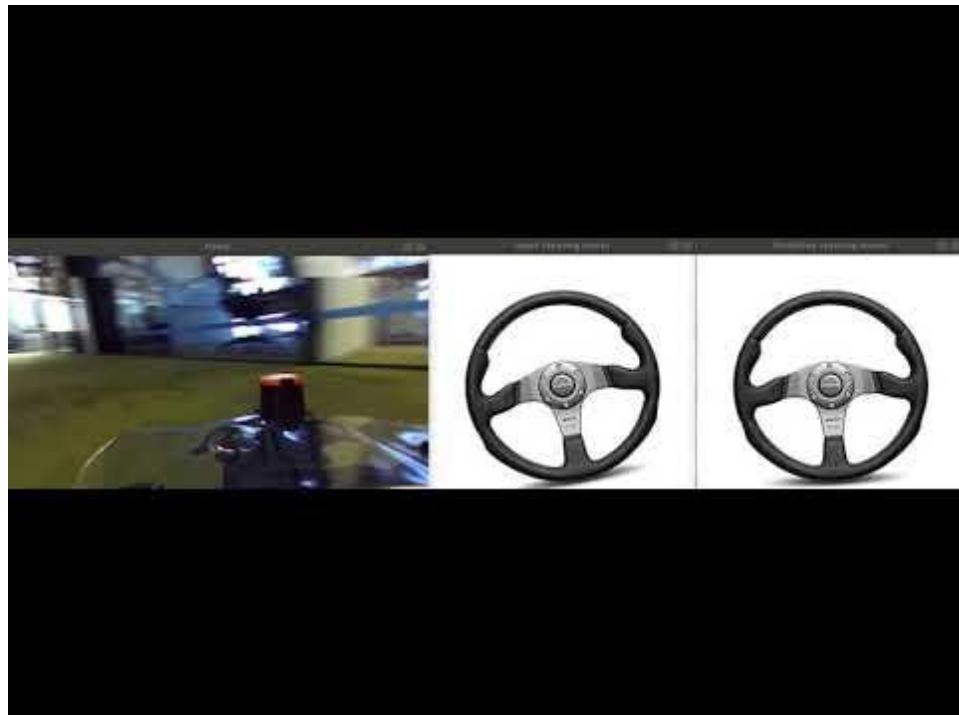
How to learn from human drivers?

Learning outcomes:

Understand imitation learning and implement it

Assignment:

RL as a project option.





f1tenth.org

F1

TENTH