Reinforcement Learning for Cooperative Adaptive Cruise Control

Dinesh Cyril Selvaraj
Shailesh Hegde

Prof. Carla Fabiana Chiasserini
Prof. Nicola Amati
Prof. Francesco Deflorio

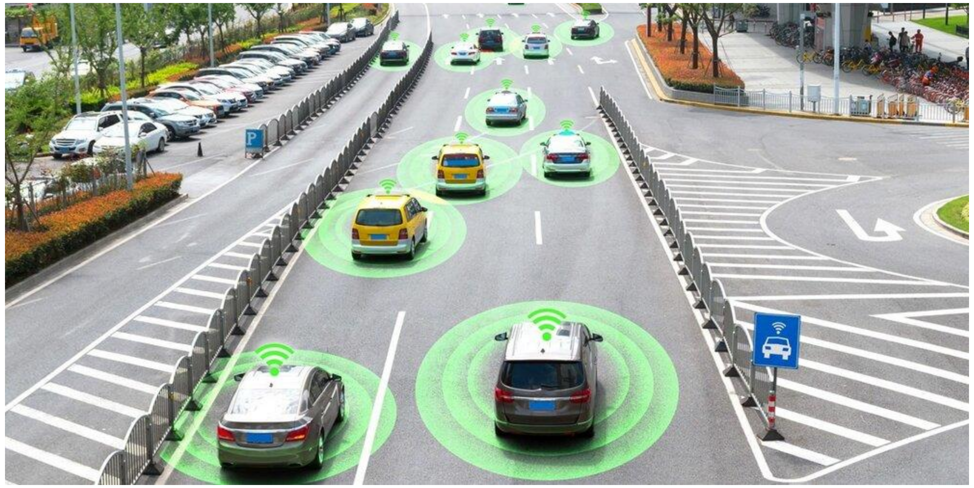
POLITECNICO DI TORINO

CARS
Center for Automotive Research and Sustainable mobility@PoliTO

Outline

Introduction

- Advancements in vehicle sensing capabilities, computational capabilities and wireless communication technologies allow for connected autonomous vehicles and transportation systems



- Vehicles are becoming data sources generating an enormous amount of information that can be used for the development of data-driven models

- Machine Learning (ML) techniques play a crucial role, as they have proved to be very effective in prediction and accurate decision making

Introduction

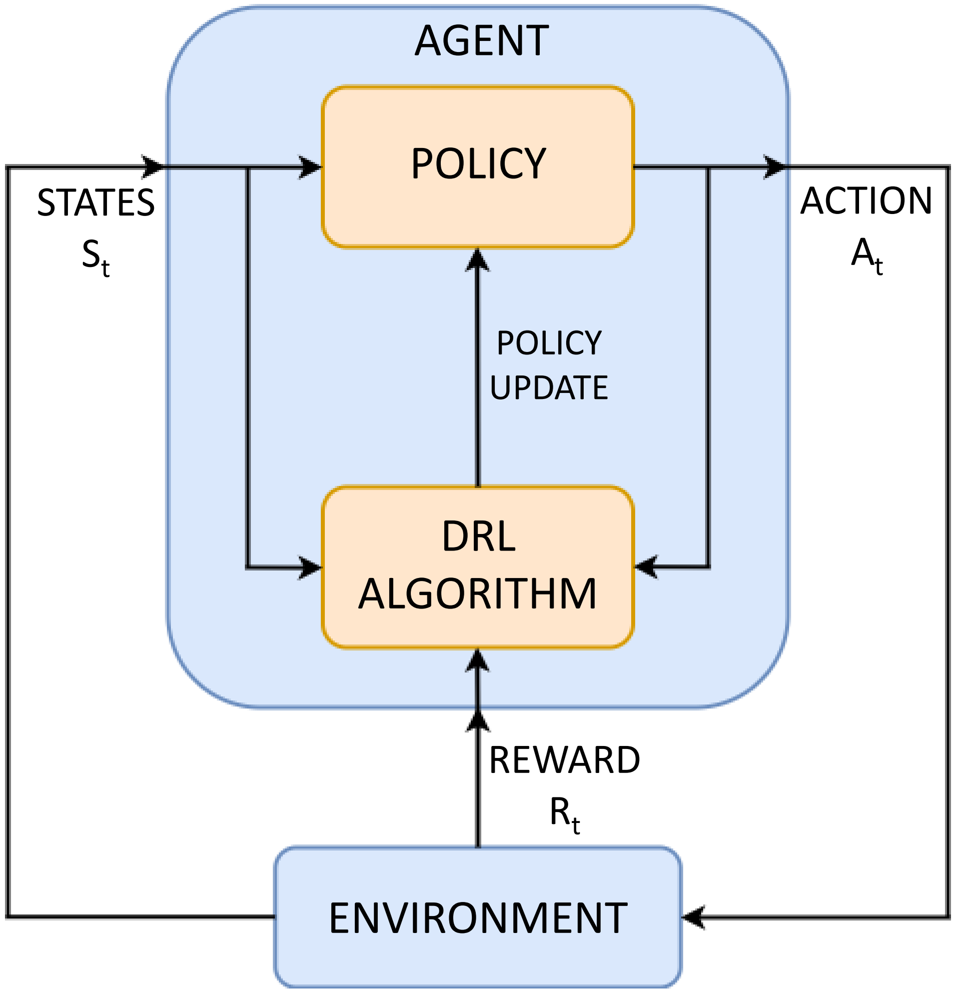
- Even with huge datasets, its highly unlikely to gather data for all possible road conditions we face in real life
- In a Dynamic ML model, data is continually collected and can be incorporated into the model with continuous updates
- We choose a Reinforcement Learning (RL) for this work due to its ability to swiftly react to the rapidly changing situations

Wet Patches

Oil Spill

Reinforcement Learning: Basics

Environment — Physical world in which the agent operates (Vehicle)
State — Current situation
Action — Contributes to determining the future state of the environment
Reward — Feedback from the environment
Policy — Method to map an agent's state to actions



Objective

To satisfy the requirements of Adaptive Cruise Control, Vehicle Stability and comfort by finding an optimal acceleration of the vehicle.

Deep Reinforcement Learning (DRL): States

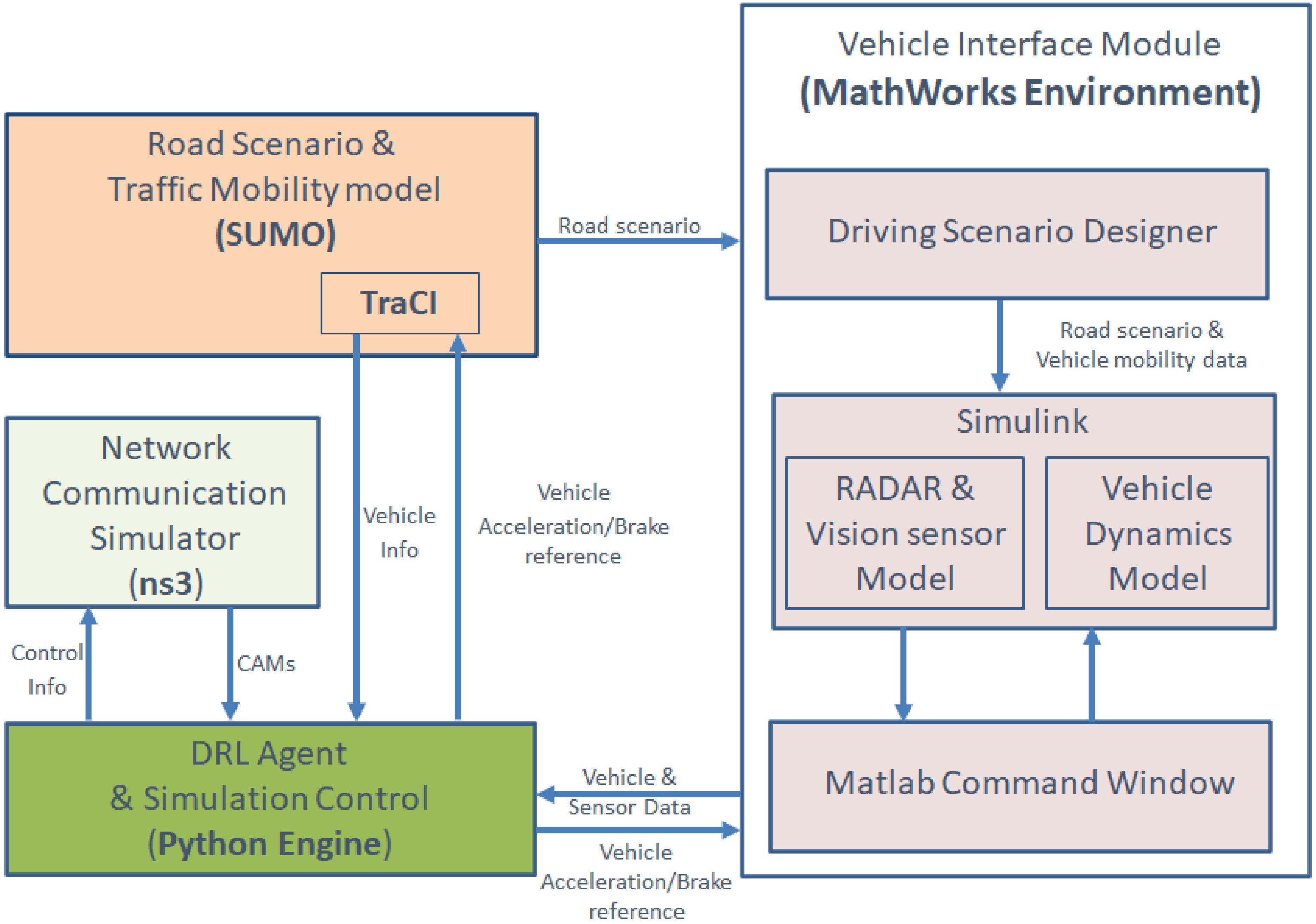
Every situation the Agent encounters in the Environment is formally called a state

Name	Representation	Formula	Definition
Acceleration (t)	$\alpha^{(t)}$	Observed	Acceleration of the preceding vehicle
Headway (t)	$\vartheta^{(t)}$	$\frac{Position_{Leader} - Position_{Follower}}{Velocity_{Follower}}$	Measurement of Inter vehicle spacing
Headway Delta (t)	$\Delta\vartheta^{(t)}$	$\vartheta^{(t)} - \vartheta^{(t-1)}$	Derivative of the headway. Gives us temporal information of the vehicle's status
Longitudinal Slip (t)	$S_l^{(t)}$	Decel: $S_l^{(t)} = \frac{v_R - v_W}{v_W}$; $Accel: S_l^{(t)} = \frac{v_R - v_W}{v_R}$	Difference between the tire tangential speed and the speed of the axle relative to the road
Friction Coefficient (t)	$\mu^{(t)}$	Given	Friction between the vehicle tires and the road

$$S \in \mathbb{R} \qquad s^{(t)} \coloneqq \{\alpha^{(t)}, \vartheta^{(t)}, \Delta\vartheta^{(t)}, S_l^{(t)}, \mu^{(t)}\}, \forall t \in \mathbb{N}$$

$v_R = \omega\, r_{stat}$, equivalent rotational wheel velocity.
$v_W = V_{CoG}$, wheel ground point velocity

CoMoVe Architecture



Integration of DRL:

- The DRL framework is integrated into the CoMoVe using Python Engine
- DRL State Variables values are gathered by Python Engine and given as observed state components to the DRL Framework
- The action (desired acceleration) of DRL Framework is given as a reference to lower-level controller (throttle/brake actuators) of the vehicle

Scenarios

- Adaptive Cruise control Scenario with an Ego Vehicle following a Lead Vehicle.
- DRL Agent will be running in the Ego Vehicle to control the velocity by satisfying all the objectives

Initial Condition

Ego Vehicle (DRL Agent)
Speed: 22.25 m/s

Lead Vehicle
Speed: 22.25 m/s

System Definition

- To learn an optimal policy for continuous action variables (Vehicle Acceleration) , Deep Deterministic Policy Gradient (DDPG) algorithm is used to train the DRL agent
- Both Actor and Critic network of DDPG have 2 hidden layers with 256 neurons for each layer

DDPG Hyperparameters	Value
Learning Rate (Actor & Critic)	0.001
Discount Factor	0.99
Mini-Batch Size	32
Experience Replay Memory Size	7000
Soft Target Update	0.001

Vehicle properties:

- Mass = 1181 [Kg]
- SI Engine
- Rear wheel driven
- 5 speed automatic transmission

Thank you