Genuense Athenaeum

Università degli Studi di Genova

Scuola Politecnica & Facoltà di Ingegneria

www.itim.unige.it/strategos

Laurea Magistrale in Ingegneria Strategica

International MSc in Engineering Technology for Strategy & Security of Genoa University

Impacts of Autonomous Vehicles on Urban Traffic and traffic control strategies

Advisors: **Prof. Agostino Bruzzone, Dr. Brunella Caroleo**

Candidate: **Mr. Javad Sadeghialavijeh**

Academic Year 2022-2023

STRATEGOS 3rd Edition

Acknowledgements

I am deeply grateful to express my heartfelt appreciation to my esteemed advisor, Professor Agostino Bruzzone, for their exceptional guidance and unwavering support throughout my master's program. Their wealth of expertise, mentorship, and unwavering encouragement have been invaluable in shaping my research and facilitating the successful completion of this thesis. Their dedication to my academic and professional growth has been an inspiration, as my professor.

Furthermore, I would like to extend my sincere gratitude to Dr. Brunella Caroleo and all my colleagues at the Links Foundation for their invaluable supervision and constructive feedback during my internship. Their mentorship and insightful inputs have significantly enriched my learning experience and contributed to my overall development. The opportunity to conduct my internship at the Links Foundation has been a privilege, and I am deeply appreciative of the knowledge and skills I have acquired through this experience.

Lastly, my heartfelt appreciation goes to my wife, Mina, for her steadfast love, encouragement, and support throughout this challenging yet ultimately rewarding journey. Her enduring belief in my abilities has been a constant source of inspiration and motivation. Her presence and support have been my pillars of strength, allowing me to overcome obstacles and aim for excellence in my academic endeavors. I am eternally grateful for her unwavering presence in my life.

I'd like to extend my sincere gratitude to everyone who has significantly influenced my academic and professional development. Their guidance and motivation have been invaluable to my progress, and I am eager to utilize the insights and expertise I've acquired to contribute meaningfully to my area of specialization.

Abstract

The rapid growth of autonomous vehicle (AV) technology holds potential to reshape urban mobility, necessitating research into its impact on traffic patterns and control strategies. This thesis focus into the interactions between Avs, the Navya autonomous minibus and conventional vehicles under various urban traffic control strategies. A key focus is assessing the effects of the Navya autonomous minibus under different AVs penetration levels using traffic simulation models. The Simulation of Urban MObility (SUMO) software forms the basis of these models.

To meet these research objectives, we had a review of the current literature on AVs. We also investigate traffic control strategies for AVs and assess their applicability in different urban settings. Moreover, we used the Simulation of Urban MObility (SUMO) software to produce a realistic environment for studying AV interactions.

Three traffic control strategies were explored: a mixed-traffic approach with no vehicular prioritization, an exclusive lane strategy for shuttles and public transport, and a lane reservation strategy targeted at specific network parts. Each strategy was tested under four AV penetration levels.

The simulation results, indicate that higher AV integration leads to reduced travel times, delays, and enhanced road efficiency. While simulations are valuable, real-world testing is essential before broad AV deployment. The findings underscore the need for cities to invest in AV-friendly infrastructure to achieve improved traffic efficiency, reduced congestion, and a modernized urban mobility landscape.

The findings of this research will offer valuable insights to policymakers, urban planners, and transportation specialists, guiding them in making informed decisions about AV deployment. Furthermore, the results will furnish practical suggestions for optimizing AV assimilation into the prevailing urban traffic framework, ultimately leading to more streamlined and sustainable urban mobility solutions.

This study was conducted during an internship at The LINKS Foundation in Torino, focusing on a specific simulation area within the city. The overarching objective is to provide insights to policymakers and urban planners on the benefits and challenges of integrating AVs into the urban fabric.

Contents

Content of figures

CHAPTER 1: INTRODUCTION

With the rapid advancements in autonomous vehicle (AV) technology and the growing interest in its potential to revolutionize urban mobility, understanding the impacts of AV integration on urban traffic and traffic control strategies has emerged as a crucial research area. As AVs move closer to widespread deployment on public roads, it is essential to comprehensively assess their effects on traffic patterns, congestion, and overall urban transportation systems.

This thesis aims to contribute to the emerging body of knowledge by examining the intricate interactions between AVs and conventional vehicles under various urban traffic strategies.

The primary objective of this research is to analyse the effects of operating the Navya autonomous minibus and introducing AVs at different penetration levels within designated testing zones. Using traffic simulation models, we aim to evaluate the effectiveness of various traffic control strategies in the context of AV presence. Moreover, this study seeks to highlight the impact of AV deployment during different times of the day and in diverse traffic conditions.

To meet these research objectives, we will embark on an extensive review of the current literature on AVs and their effects on traffic flow and congestion. We will also investigate traffic control strategies tailored for AVs and assess their applicability in different urban settings. Moreover, we will devise a traffic simulation model using the Simulation of Urban MObility (SUMO) software to produce a realistic environment for studying AV interactions.

Integrating AVs into urban settings holds the potential for enhanced traffic management, heightened safety, and a more efficient transportation network. However, reaping these benefits hinges on a thorough understanding of how AVs interact with the existing traffic and infrastructure. This thesis, therefore, addresses the pivotal research questions about the degree to which AV penetration affects traffic dynamics and how AV movement aligns with distinct traffic control strategies.

The anticipated findings of this research will offer valuable insights to policymakers, urban planners, and transportation specialists, guiding them in making informed decisions about AV deployment. Furthermore, the results will furnish practical suggestions for optimizing AV assimilation into the prevailing urban traffic framework, ultimately leading to more streamlined and sustainable urban mobility solutions.

In the subsequent sections of this thesis, we will explore the methodology, simulation results, discussions, and conclusions concerning the impacts of AVs on urban traffic and traffic control strategies. Through this study, our objective is to chart a path towards a safer, more efficient, and technology-centric urban transportation landscape.

CHAPTHER 2: Literature Review

This section provides a comprehensive overview of the traffic simulation and Autonomous Vehicles (AVs).

2-1 Traffic simulation

The continuous growth of road traffic volumes leads to significant environment and economic problems. For this reason, there have been efforts for more than four decades to understand the dynamics of traffic flow in order to find ways to optimize traffic. Traffic simulation is a valuable computational technique that offers a multitude of advantages for studying, analyzing, and optimizing transportation systems. As urban environments become increasingly complex and the demands on transportation infrastructure grow, the importance of understanding and managing traffic patterns becomes paramount. Traffic simulation serves as a powerful tool to address these challenges, providing insights and solutions that can lead to more efficient, safe, and sustainable mobility.

For the implementation of traffic management solutions accurate knowledge of the traffic conditions and dynamics is necessary. Traffic simulation frameworks provide a helpful tool to answer complex research questions, to evaluate or to test traffic management strategies and their impacts. (Pablo Alvarez Lopez, 2018)

There are several traffic simulation software, such as SUMO, MATSim, MITSIMlab, AIMSUN, CORSIM, Paramics, SimTraffic, VISSIM, TRANSIMS, etc. Some software focus on the behavior of the vehicle in detail, the others are not interested in because they insist much on the simulation of a wide area. (Mustapha Saidallah, 2016)

2-1-1 The traffic simulation groups

The traffic simulation tools can mainly be divided into four different groups. (Krauss, 1998)

1) Macroscopic: In Macroscopic softwares, average vehicle dynamics like traffic density are simulated. These models simulate the flow of traffic. They consider traffic characteristics (speed, flow and density) and their relationships. These models are making on the conservation equations of flows and traffic disturbances that spread in the traffic system. Consequently, they can be used to predict the spatial and temporal congestion that is caused by the traffic demand or incidents in a road network. (Mustapha Saidallah, 2016)

2) Microscopic: In Microscopic softwares, each vehicle and its dynamics are modeled individually. These models simulate the characteristics and interactions between individual vehicles. They produce essentially the trajectories of vehicles moving across the network. The processing logic includes algorithms and rules that describe how vehicles move and interact. It also includes acceleration, deceleration, lane changes and overtaking maneuvers. (Mustapha Saidallah, 2016)

3) Mesoscopic: In Mesoscopic softwares a mixture of macroscopic and microscopic model. These models simulate individual vehicles, to this end, traffic is represented by small groups of traffic entities, whose interactions are described in a medium level of detail. (Mustapha Saidallah, 2016)

4) Submicroscopic: In Submicroscopic softwares each vehicle and also functions inside the vehicle are explicitly simulated e.g. gear shift.

In this thesis our focus is on microscopic simulator. The advantage of macroscopic models is normally its fast execution speed. However, the detailed simulation of microscopic or submicroscopic models are more precise especially when emissions or individual routes should be simulated. (Pablo Alvarez Lopez, 2018) These tools provide a detailed view of individual entities within a system, allowing for the in-depth study of complex interactions scenarios. Microscopic simulator allows researchers and planners to understand specific behaviours and interactions at a detailed level. They can simulate a wide range of scenarios, from regular daily traffic patterns to uncommon events like accidents or road closures. This versatility aids in testing different solutions for diverse situations. With the ability to emulate real-world conditions, microstimulators can predict how individual entities react to specific stimuli or changes in the environment like a new traffic signal or pedestrian crossing, can impact the overall flow and efficiency of traffic, ensuring more accurate outcomes. (Barceló, 2010)

2-1-2 Category of Traffic Simulators

Regarding traffic simulation software accessibility, we can categorize them into two types: open-source and commercial. The open-source designation or "open-source code" applies to software whose license specifically meets the criteria established by the Open-Source Initiative; that is to say, the possibilities of free redistribution, access to source code and create derivative works. While a commercial software or proprietary means software that does not allow legally or technically, or by any other means whatsoever, to simultaneously perform four software freedoms which are running the software for any type of use, to study its source code (and therefore access to the source code), the distribution of copies, as well as modification and thus the source code improvement. (Mustapha Saidallah, 2016)

In this thesis we are using SUMO as open-source traffic simulation. Using open-source software can offer several benefits to companies across various industries. Here are some key advantages:

- **Cost Savings:** Open-source software is typically free to use, which can result in significant cost savings for companies. Instead of purchasing expensive licenses for proprietary software, businesses can allocate their budgets to other critical areas.
- **Flexibility and Customization:** Open-source software provides flexibility, allowing companies to tailor the software to their specific needs. Developers can modify the source code to add features, fix bugs, or adapt the software to unique requirements.
- **Rapid Development and Updates:** Open-source projects often evolve quickly because of contributions from a large community of developers. This means that bugs get fixed and new features get added more rapidly than in proprietary software, allowing companies to stay up to date with the latest advancements.
- **Interoperability:** Open-source software tends to adhere to open standards, making it easier to integrate with other tools and systems within a company's technology stack. This promotes interoperability and simplifies data exchange between different software solutions.
- **Licensing Freedom:** Open-source licenses generally provide more freedom and fewer restrictions than proprietary licenses. This means that companies can use, modify, and distribute open-source software without worrying about compliance issues or additional costs.
- **Innovation and Collaboration:** By using open-source software, companies can contribute to the development of the software and collaborate with other organizations that share similar needs. This fosters innovation and can lead to the creation of new features and capabilities that benefit all users.

open-source software can offer numerous advantages for companies, including cost savings, flexibility, community support, security, and the ability to drive innovation. However, it's important to carefully evaluate the specific needs and requirements of your organization when considering the adoption of open-source solutions.

2-1-3 Visualization of Traffic Simulators

Users can explore a range of scenarios and "what-if" analyses through visualizations, enabling them to test hypotheses and plan for various scenarios. Visualization makes easier to comprehend complex systems and processes. Visualizations enable researchers and analysts to gain insights from data that might be difficult to discern from raw numbers or textual descriptions. Patterns, trends, and anomalies become apparent through graphical representations. In traffic simulation, visualizations of simulation results can engage the public and foster informed discussions about proposed projects or policies. visualization in simulation software is essential for improving comprehension, analysis, decision-making, training, and communication across various domains. It enhances the value of simulations by making complex data and processes more accessible and actionable. As a result, it is a critical component in many industries and research disciplines.

Visualization can be two-dimensional (2D), three dimensional (3D) or both. 3D visualization allows to be closer to the real world. Also, it gives details of the simulation (eg the vehicle height, better visibility of traffic lights) (Mustapha Saidallah, 2016). SUMO is a two-dimensional (2D) simulator, but the connection between SUMO and a 3D simulator like Unity or Carla can be beneficial for creating more comprehensive and immersive simulations, particularly for scenarios involving urban traffic and autonomous vehicles. While there isn't a direct integration between SUMO and Unity, we can establish a connection between the two through various approaches.

In Unity, one can develop custom scripts to parse and visualize exported data. This capability facilitates the recreation of the traffic scenario from SUMO within Unity's 3D environment. For real-time simulations, a communication link can be established between SUMO and Unity using network protocols. This link permits data exchange between the two systems during the simulation, allowing Unity to receive, interpret, and dynamically visualize the traffic scenario based on real-time data from SUMO. (Chrysostomos, 2019)

2-2 SUMO (Simulation of Urban Mobility)

Eclipse SUMO (Simulation of Urban Mobility) is a free and open-source Microscopic Traffic Simulation package designed to handle large networks. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center. (https://www.dlr.de/ts, n.d.) It is available since 2001 and allows modelling of intermodal traffic systems - including road vehicles, public transport and pedestrians. Included with SUMO is a wealth of supporting tools which automate core tasks for the creation, the execution and evaluation of traffic simulations, such as network import, route calculations, visualization and emission calculation. SUMO can be enhanced with custom models and provides various APIs to remotely control the simulation. (https://sumo.dlr.de/docs/index.html, n.d.)

The development of SUMO started in the year 2000. The major reason for the development of an open source, microscopic road traffic simulation was to support the traffic research community with a tool with the ability to implement and evaluate own algorithms. The tool has no need for regarding all the needed things for obtaining a complete traffic simulation such as implementing and/or setting up methods for dealing with road networks, demand, and traffic controls. By supplying such a tool, the DLR wanted to i) make the implemented algorithms more comparable by using a common architecture and model base, and ii) gain additional help from other contributors. (https://sumo.dlr.de/docs/index.html, n.d.)

SUMO allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route, and moves individually through the network. Simulations are deterministic by default but there are various options for introducing randomness.

2-2-1 Software design criteria

In Sumo two major design goals are approached: the software shall be fast, and it shall be portable. Due to this, the very first versions were developed to be run from the command line only - no graphical interface was supplied at first and all parameters had to be inserted by hand. This should increase the execution speed by leaving off slow visualization. Also, due to these goals, the software was split into several parts. Each of them has a certain purpose and must be run individually. This is something that makes SUMO different to other simulation packages where, for instance, the dynamical user assignment is made within the simulation itself, not via an external application like here. This split allows an easier extension of each of the applications within the package because each is smaller than a monolithic application that does everything. Also, it allows the usage of faster data structures, each adjusted to the current purpose, instead of using complicated and ballast-loaded ones. Still, this makes the usage of SUMO a little bit uncomfortable in comparison to other simulation packages. As there are still other things to do, we are not thinking of a redesign towards an integrated approach by now. (https://sumo.dlr.de/docs/index.html, n.d.)

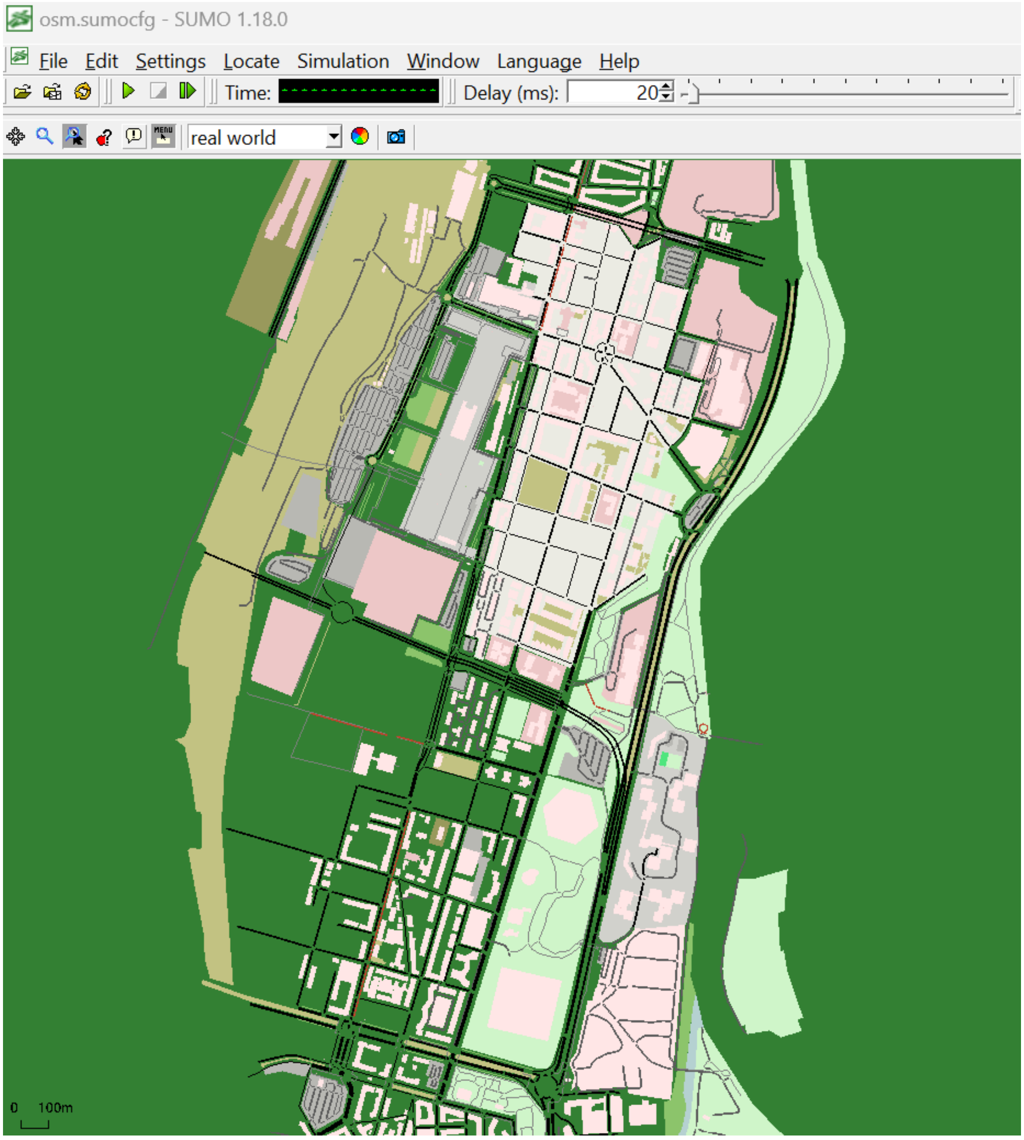


Figure 1: Simulation Area in SUMO

2-2-2 Features of sumo:

There are many features for sumo that make it a good traffic simulator. In below we speak about some of them.

- Simulation
    - Space-continuous and time-discrete vehicle movement
    - Different vehicle types
    - Multi-lane streets with lane changing
    - Different right-of-way rules, traffic lights
    - A fast openGL graphical user interface
    - Manages networks with several 10.000 edges (streets)
    - Fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine)
    - Interoperability with other application at run-time
    - Network-wide, edge-based, vehicle-based, and detector-based outputs
    - Supports person-based inter-modal trips
- Network Import
    - Imports VISUM, Vissim, Shapefiles, OSM, RoboCup, MATsim, OpenDRIVE, and XML-Descriptions
    - Missing values are determined via heuristics
- Routing
    - Microscopic routes - each vehicle has an own one
    - Different Dynamic User Assignment algorithms
- High portability
    - Only standard C++ and portable libraries are used
    - Packages for Windows main Linux distributions exist

2-2-3 TraCI

Running Sumo is heavily depended to python code and specially using TraCI library. TraCI (short for Traffic Control Interface) is an API that provides access to a SUMO traffic simulation, enabling controlling the behavior of multiple simulation objects during a live simulation. TraCI serves as a bridge between Python and SUMO, enabling you to control and interact with the simulation from within your Python scripts. It allows to retrieve values of simulated objects (vehicles, pedestrians, traffic lights, lanes, edges, and traffic control strategies, and other infrastructure) and to manipulate their behavior "on-line". TraCI uses a TCP based client/server architecture to provide access to sumo. Thereby, sumo acts as server. (https://sumo.dlr.de/docs/index.html, n.d.)

To use the library, is needed to install it using pip install Traci or add the <*SUMO_HOME*>/tools directory to your Python load path. This is typically done with a stanza like this:

```
import os
import sys
if 'SUMO_HOME' in os.environ:
    sys.path.append(os.path.join(os.environ['SUMO_HOME'], 'tools'))
import traci
```

After connecting to the simulation, we can emit various commands and execute simulation steps until you want to finish by closing the connection. By default, the close command will wait until the sumo process really finishes.

TraCI can be used either as an interface for other programs or by means of python scripts that interact with the simulation for a given scope. As already mentioned, TraCI works through a series of commands split into 14 different domains corresponding to the individual modules: Gui, poi, simulation, lane, edge, route, traffic light, junction, induction loop, multi entry-exit, polygon, person, vehicle and vehicle type. (https://sumo.dlr.de/docs/TraCI.html, n.d.) The TraCI python library can be used to control multiple simulations at the same time with a single script. The TraCI allows to call it multiple times with different simulation instances and labels. It is a great advantage of SUMO for complex simulation scenario.

2-2-4 SUMO research

SUMO has been used within several projects for answering a large variety of research questions there are a brief explanation about sumo work in recent years and ability of sumo about simulation of traffic:

- Evaluate the performance of traffic lights, including the evaluation of modern algorithms up to the evaluation of weekly timing plans.
- Vehicle route choice has been investigated, including the development of new methods, the evaluation of eco-aware routing based on pollutant emission, and investigations on network-wide influences of autonomous route choice.
- SUMO was used to provide traffic forecasts for authorities of the City of Cologne during the Pope's visit in 2005 and during the Soccer World Cup 2006.

- SUMO was used to support simulated in-vehicle telephony behaviour for evaluating the performance of GSM-based traffic surveillance.
- SUMO is widely used by the V2X community for both, providing realistic vehicle traces, and for evaluating applications in an on-line loop with a network simulator.
- AI training of traffic light plans.
- Simulation of the traffic effects of autonomous vehicles and platoons.
- Simulation and validation of autonomous driving function in cooperation with other simulators.
- Simulation of parking traffic.
- Simulation of railway traffic for AI-based dispatching of vehicles.
- Traffic safety and risk analysis.
- Calculation of emissions (noise and pollutants).

2-3 Autonomous Vehicles

Autonomous vehicles, also known as self-driving cars or driverless vehicles, refer to vehicles equipped with advanced sensors, artificial intelligence (AI), and control systems that enable them to navigate and operate on roads without direct human input. (James M. Anderson, 2014) Autonomous vehicles encompass a range of vehicle types, from passenger cars to trucks and buses, that integrate sensor technologies, machine learning algorithms, and connectivity to interact with their surroundings and operate autonomously without human intervention.

Autonomous vehicles can perceive their environment, make decisions, and execute driving tasks without human intervention. They rely on a combination of sensors, cameras, radar, lidar, and sophisticated algorithms to interpret their surroundings and navigate safely. (Automated Driving Systems 2.0: A Vision for Safety, 2021)

AVs represent a technological evolution in transportation, embodying the capability to operate autonomously in various driving scenarios, ranging from routine commuting to complex urban environments. These vehicles have the potential to enhance road safety, reduce traffic congestion, and revolutionize mobility.

The research and development of Autonomous Vehicles (AVs) have seen rapid advancements in recent years. Traffic simulation plays a pivotal role in modeling and analyzing the interaction between autonomous and human-driven vehicles. Through simulation, researchers can create virtual environments that replicate real-world traffic scenarios, enabling them to explore different interaction patterns and test various control strategies. These simulations allow researchers to assess the impact of autonomous vehicle behavior on overall traffic flow, safety, and efficiency.

Behavior modeling, involving the development of algorithms that capture the decision-making processes of both autonomous and human drivers. This includes modeling factors such as lane changes, merging, overtaking, and yielding behaviors. Accurate behavior modeling enables researchers to investigate potential conflicts, evaluate traffic management interventions, and optimize the design of autonomous vehicle control systems. (Xiangquan Chen, 2023)

It's important to note that no simulation can perfectly replicate real-world conditions, so while simulation tools are invaluable in the research phase, on-road testing under controlled conditions is essential before wide-scale deployment of AV technologies.

2-4 Interaction between Autonomous and Human-Driven Vehicles

The interaction between autonomous and human-driven vehicles presents a complex challenge in the integration of autonomous technology into existing traffic environments. Autonomous vehicles are equipped with advanced sensors and communication technologies that enable them to perceive their surroundings and interact with other vehicles, pedestrians, and infrastructure in real-time. Human drivers, relying on visual cues and past experiences, communicate and make decisions through a combination of signals, gestures, and vehicle movements.

As AV technology continues to develop, ongoing research and real-world testing are essential to ensure that AVs can safely and efficiently interact with human-driven vehicles.

AVs not only need to follow traffic rules but must also anticipate the often-unpredictable behaviours of human drivers. (Shladover, 2016) Human drivers don't always behave logically or follow traffic rules to the letter. Anticipating such behaviours like sudden lane changes or aggressive driving, is a significant challenge for AVs. (Ashesh Jain, 2015)

Cooperative driving and Vehicle-to-Everything (V2X) communication can help in improving the interaction between human-driven and autonomous vehicles. V2X allows vehicles to communicate with each other and with infrastructure, thereby providing a better understanding of the surrounding environment. (Kenney, 2011)

For smooth interactions, human drivers need to trust and accept autonomous vehicles on the road. Studies have been done to understand public perceptions and the factors that influence trust in AVs. (M. Kyriakidis, 2015)

Autonomous vehicles must anticipate the behavior of human drivers and respond appropriately to ensure smooth and predictable interactions. studies are shedding light on how human drivers perceive and interact with autonomous vehicles. Factors influencing human trust, acceptance, and reactions to autonomous vehicles are being examined to inform the design of autonomous behaviors that resonate well with human drivers. This understanding is crucial for the successful integration of autonomous vehicles into the existing transportation ecosystem and ensuring a harmonious coexistence between different vehicle types.

2-5 Innovative Traffic Control Strategies

In the rapidly evolving landscape of transportation, innovative traffic control strategies are pivotal in addressing the challenges posed by increasing urbanization and the integration of emerging technologies. These strategies encompass a range of approaches that leverage advanced technologies, data-driven insights, and adaptive systems to optimize traffic flow, enhance safety, and improve overall transportation efficiency. Here are some key innovative traffic control strategies:

1. Dynamic Traffic Control strategy: is an advanced method of traffic management that adapts to real-time traffic conditions to optimize the flow of vehicles through intersections. This system often uses a combination of inductive loop detectors, infrared sensors, and cameras to detect vehicles. Advanced systems might integrate with connected vehicle technologies, where vehicles communicate directly with infrastructure, providing another data source for traffic management. (Ma, 2019). It Reduced congestion, improved traffic flow, lowered emissions due to reduced idling at intersections, and enhanced safety. On the other hands, High initial infrastructure costs, the need for ongoing maintenance, potential privacy concerns (with camera systems), and the requirement for continuous data and network reliability.
2. Connected Vehicle Technology: Connected vehicle technology enables vehicles to communicate with each other and with infrastructure, such as traffic signals and road signs. This communication facilitates real-time exchange of information about traffic conditions, road hazards, and upcoming signal changes. Cooperative adaptive cruise control and platooning are examples of connected vehicle applications that can improve traffic flow and reduce congestion. (By Harding, Powell, & Yoon, 2014)
3. Traffic Flow Optimization through Data Analytics: Data analytics techniques, including machine learning and big data analysis, are employed to extract insights from various data sources, such as traffic sensors, GPS data, and social media. By analyzing historical and real-time data, transportation agencies can identify traffic patterns, predict congestion, and proactively implement strategies to mitigate bottlenecks and improve traffic flow.

4. Dynamic Route Navigation: Advanced navigation systems provide real-time route guidance to drivers, considering current traffic conditions and congestion levels. These systems offer alternative routes and dynamically adjust recommendations based on changing traffic patterns, helping drivers make informed decisions that optimize travel time and reduce congestion. By integrating data-driven insights and adaptive systems, these strategies offer promising solutions to the complex challenges of urban mobility. By being well-informed of the outcomes the implementation of these strategy, urban planners can make more strategic decisions, optimizing the benefits and mitigating any potential adverse effects. Simulation of these strategy gives this essential information to them.

2-7 The Navya autonomous shuttle

As part of The LINKS Foundation's project, they launched two Navya minibuses in Turin. The Navya minibus is an autonomous shuttle capable of carrying up to 15 passengers. This cutting-edge vehicle can operate on both public and private roads with its high-performance guidance and detection system. Thanks to its advanced sensors, the Navya shuttle can precisely locate itself within its environment and adapt its navigation to various situations, such as obstacles, other vehicles, and pedestrians. Fitted with cutting-edge technology, this vehicle can run on public or private roads and paths. (https://www.navya.tech/en/, n.d.)

Figure 2: Navya minibuses

The main goal was to develop scenarios that could be used to safely and efficiently integrate autonomous vehicles into the urban environment. This project is corporation between The LINKS Foundation, Politecnico di Torino and GTT (Gruppo Torinese Trasporti).

The Navya shuttle was tested on Turin streets for several months, and data was collected using different types of sensors to analyze its behavior, interactions, and performance. The collected data analysis by The LINKS Foundation to extract the input data for traffic simulations. We will use these data as input data for future simulation of shuttle in different traffic scenario and also for validation of our simulation in future works.

Figure 3: Testing Navya minibuses in Turin Street

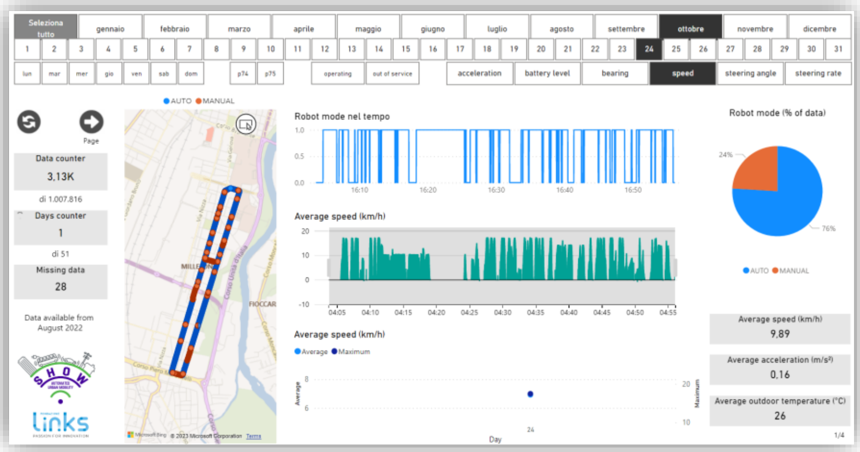


Figure 4: Data Analysis of testing Navya minibuses

2-8 Traffic Behaviour models

Behavior models in traffic simulations, such as lane-changing and car-following models, are essential components that define how vehicles interact with each other and navigate the road network. These models aim to replicate real-world driving behaviors, enabling realistic and accurate traffic simulations. These models could change according to the driving mechanisms. The primary distinction between lane changing models for human-driven vehicles and AVs is how decisions are made and executed by humans versus machines. Here are some of the reasons why different models exist for the two types of vehicles:

- **Decision-making Mechanism**: In Human-driven Vehicles decisions are based on human perception, intuition, experience, and sometimes emotions. Humans have a natural ability to judge situations and can sometimes make decisions based on "gut feelings" or unquantified observations. But in Avs decisions are based on algorithms, sensor data, and predefined rules. AVs lack intuition and emotion, relying instead on logical and deterministic processes.

- **Sensing Capabilities**: Humans primarily rely on visual cues, which can be subjective and sometimes influenced by cognitive biases or distractions. But Avs use an array of sensors (LiDAR, radar, cameras, ultrasonic sensors, etc.) that provide objective and quantified data about the surroundings. This rich dataset can be processed rapidly for decision-making.

- **Reaction Times**: the Human reaction times can vary and may be influenced by fatigue, distractions, or other factors. But AVs can react almost instantaneously to sensor data without the delays inherent in human perception and physical response.

- **Predictability**: Humans can be unpredictable in their actions, influenced by factors like mood, aggressiveness, or even inexperience. While AVs operate based on algorithms, making them more predictable. However, they need to account for the unpredictability of human drivers around them.

- **Communication**: Drivers might use hand signals, eye contact, or other non-verbal cues to communicate intentions to other drivers. Avs currently, AVs lack these nuanced communication methods, but there's ongoing research into vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication that could compensate for this.

- **Learning and Adaptation**: Drivers learn from experience, and their driving behaviour can evolve over time. While AVs can be updated with new algorithms, and with the incorporation of machine learning, they can adapt to new driving scenarios or improve based on accumulated data.

Given these differences, it's clear that while there may be some overlap, lane changing models tailored specifically for AVs need to account for their unique decision-making processes, capabilities, and constraints. On the other hand, models for human-driven vehicles need to capture the complexities, nuances, and sometimes irrationality of human behavior. Moving forward, we will discuss two significant models in traffic simulation behaviors: car-following models and lane-changing models.

2-8-1 car-following models

Car-following models are mathematical representations used in traffic flow theory and simulation to describe how individual vehicles adjust their speed and spacing relative to other vehicles on the road. These models are crucial for understanding and predicting traffic behavior in various road conditions and are a fundamental part of transportation research and simulation. Common Car-Following Models:

1. Intelligent Driver Model (IDM):

IDM is a widely used car-following model that considers factors like desired speed, minimum gap, time headway, and comfortable deceleration. It aims to replicate human driving behaviour by modelling how drivers adjust their speeds to maintain safe distances. (Martin Treiber A. H., 2000)It is designed to replicate the behaviour of human drivers by considering various factors that influence their decisions and actions while driving.

2. Krauss Car-Following Model (Krauss-McFadden Model):

The Krauss model is known for its simplicity and effectiveness in representing car-following dynamics. It focuses on drivers' behaviour in queues and is often used in traffic simulation.

3. Gipps Model:

The Gipps model incorporates factors such as desired speed, reaction time, and a safety time gap. It aims to capture driver behaviour in different traffic conditions and is particularly useful for modelling congested traffic.

4. Follow the Leader (FTL) Model:

The FTL model is a simple car-following model where each vehicle tries to maintain a fixed gap with the leading vehicle by adjusting its speed. It is a fundamental model used for understanding traffic flow principles.

SUMO is a dynamic and evolving software package, and new models and improvements are regularly added by the development community. The specific car-following models available in SUMO can vary and are typically updated in newer versions of the software. In this thesis we implemented IDM Car-following model. There are some compelling arguments for using the Intelligent Driver Model (IDM) in my research or simulations, along with relevant references to support these points:

Realistic Representation of Human Behavior: IDM is designed to capture the behaviours of real-world drivers by considering factors like desired speed, minimum gap, time headway, and comfortable deceleration. It offers a realistic representation of how drivers adjust their speeds in response to traffic conditions. (Martin Treiber A. H., 2000)

Versatility Across Traffic Conditions: IDM is versatile and can be used to model a wide range of traffic conditions, from free-flowing traffic to congested scenarios. Its parameters can be adjusted to simulate various driving behaviours. (Arne Kesting, 2010)

Well-Documented and Widely Accepted: IDM is one of the most well-documented car-following models in the literature. It has been widely accepted and used in the field of traffic flow research, making it a reliable choice for your simulations. (Martin Treiber A. K., 2013)

Integration with Simulation Tools: IDM is integrated into various traffic simulation software packages, including SUMO, making it accessible and practical for simulating traffic scenarios.

Parametric Adjustability: IDM allows you to adjust its parameters to match specific real-world scenarios or to test different hypotheses. This flexibility can be valuable for conducting sensitivity analyses in your research.

Established Use in Traffic Flow Research: IDM has been extensively used in traffic flow research and has contributed to a better understanding of congestion, traffic stability, and flow dynamics. Utilizing IDM can allow you to build on a substantial body of existing research.

These arguments, emphasize the strengths and advantages of using the Intelligent Driver Model (IDM). IDM's realistic representation of human driving behaviours, versatility, wide acceptance, and parametric adjustability make it a valuable choice for studying various aspects of traffic flow and congestion. IDM model plays a pivotal role in understanding and simulating traffic behaviours in a wide range of scenarios. Its realistic representation of human driving behaviours, versatility, and integration with simulation tools make it a valuable asset for traffic research, management, and the development of intelligent transportation systems (ITS).

Model Principles of IDM:

IDM is based on the premise that drivers aim to maintain a desired speed while keeping a safe following distance from the vehicle ahead. Its key principles include:

- Desired Speed (v0): Each driver has a desired speed that they would like to maintain when traffic conditions permit.
- Minimum Gap (s0): IDM considers a minimum gap that drivers aim to maintain to ensure safety. This gap represents the minimum spacing between vehicles.
- Time Headway (T): Drivers strive to keep a desired time headway to the vehicle in front, which is the time it would take to travel the minimum gap at the desired speed.
- Comfortable Acceleration (a): This parameter represents the comfortable or preferred acceleration of a vehicle when approaching its desired speed.
- Comfortable Deceleration (b): Comfortable deceleration is the rate at which a vehicle prefers to decelerate when slowing down. It ensures smooth braking behaviour. (Martin Treiber A. K., 2013)
- Exponent (delta): The exponent parameter determines the sensitivity of the IDM's acceleration to changes in the headway. A higher exponent makes acceleration more sensitive to headway variations. (Arne Kesting, 2010)
- Minimum Spacing (s0_min): This parameter sets the lower bound for the minimum gap, allowing for the possibility of drivers accepting smaller safe distances in certain situations.
- Maximum Spacing (s0_max): Similar to the minimum spacing, this parameter sets the upper limit for the minimum gap.

The IDM Model is often employed to model the behaviours of both automated vehicles (AVs) and conventional (common) vehicles, but there can be differences in the parameters used for each. Here are some of the key parameters in the IDM model and how they might differ for AVs compared to common vehicles:

Desired Speed (V0):

Common Vehicles: For conventional vehicles, the desired speed is typically set by the driver based on factors like speed limits, traffic conditions, and personal preferences.

AVs: Autonomous vehicles may have a pre-defined desired speed based on traffic regulations and system settings. This speed can be adjusted by the AV's control algorithms to optimize traffic flow or ensure safety.

Time Headway (T):

Common Vehicles: The time headway represents the minimum time gap a driver maintains with the vehicle in front and is often influenced by the driver's reaction time and comfort level.

AVs: Autonomous vehicles can have much shorter time headways because they can react faster than human drivers. AVs may also coordinate with each other to maintain closer spacing, improving traffic flow.

Acceleration Exponent (β):

Common Vehicles: The acceleration exponent in the IDM model accounts for how drivers adjust their acceleration concerning the desired speed and the presence of other vehicles. Common vehicles may have a typical value for β based on human driving behaviours.

AVs: Autonomous vehicles can have different values of β that are optimized for efficient and safe driving. These values may change depending on traffic conditions and objectives.

Comfortable Deceleration (b):

Common Vehicles: The comfortable deceleration parameter is influenced by the driver's comfort and braking habits.

AVs: Autonomous vehicles can have precise control over their deceleration, often exceeding the limits of human comfort for maximum safety and efficiency.

Minimum Gap (s0):

Common Vehicles: The minimum gap is the desired following distance a driver maintains under free-flow conditions.

AVs: Autonomous vehicles can safely operate with smaller minimum gaps due to their faster reaction times and ability to communicate with other AVs, enabling platooning and closer spacing.

Maximum Acceleration (a_max) and Maximum Deceleration (a_min):

AVs: Autonomous vehicles can have defined maximum acceleration and deceleration limits that may be higher than those of common vehicles, allowing for quicker response and manoeuvring.

the specific parameter values for AVs can vary depending on the manufacturer, the type of autonomous system, and the goals set for the AV in a given traffic scenario. Additionally, AVs can adapt their parameters dynamically based on real-time sensor data and communication with other AVs and infrastructure, whereas common vehicles rely on human drivers' reactions and decisions.

2-8-2 Lane changing models

Lane changing is a critical component in traffic flow and microscopic traffic simulation. Various models have been proposed to describe and predict lane-changing behaviour based on driver psychology, vehicle dynamics, and traffic conditions. Here's a brief overview of some lane changing models with references:

1. **Gipps' Lane Changing Model**: Gipps proposed a model where the decision to change lanes is based on safety criteria and incentive criteria. The safety criteria ensure that a lane change will not result in a collision, while the incentive criteria evaluate if the lane change will provide a benefit, such as reaching a destination faster.
2. **MOBIL (Minimizing Overall Braking Induced by Lane Changes)**: Developed by Kesting, Treiber, and Helbing, MOBIL focuses on the minimization of braking manoeuvres induced by lane changes. It evaluates both the safety and incentive for lane changing, emphasizing the effects of a lane change on other vehicles.
3. **Fuzzy Logic-based Lane Change Models**: These models use fuzzy logic to model the uncertainty and variability in human driver decision-making during lane changes. Factors like relative speed, gap acceptance, and urgency are considered in these models.
4. **Discrete Choice Models**: These models approach lane changing as a choice problem where drivers evaluate the utility of different lanes based on various factors. The choice with the highest utility is selected. (Tomer Toledo, 2003)
5. **Operational, Tactical, and Strategic Model**: This model categorizes lane changes into three levels: operational (immediate necessity, e.g., avoiding an obstacle), tactical (short-term benefits, e.g., faster-moving lane), and strategic (long-term objectives, e.g., highway exit). Each level has different decision-making criteria. (Hidas, 2002)

These models aim to capture the complexity and variability of lane-changing behaviour of human-driven vehicles. When integrated into traffic simulators, they help provide a more accurate representation of real-world driving dynamics. However, it's crucial to note that lane-changing behaviour is influenced by numerous factors, and no model can capture all its intricacies perfectly. For AVs models will change. The primary distinction between lane changing models for human-driven vehicles and Avs stems from the fundamental differences in how decisions are made and executed by humans versus machines.

Lane changing for Autonomous Vehicles (AVs) is a critical research area since the decision-making process involves not only the vehicle's sensors and algorithms but also interactions with human-driven vehicles. This decision-making is more complex than for human drivers, as it requires the integration of sensor data, prediction of the surrounding environment, and safe execution of the manoeuvre. SUMO supports customizable vehicle behaviour, which means that many lane changing models can potentially be implemented in SUMO. Here's a breakdown regarding the lane-changing models for AVs and their feasibility within SUMO:

- **Reinforcement Learning-based Models**: Researchers have explored using reinforcement learning for lane-changing decisions in AVs. These models allow vehicles to learn optimal strategies over time by interacting with their environment. (Hanna Krasowski, 2020) SUMO can be interfaced with reinforcement learning algorithms using tools like Python's TraCI (Traffic Control Interface). It allows for dynamic control of simulation parameters, making it feasible to use reinforcement learning for lane-changing decisions in AVs.

- **Probabilistic Prediction Models**: These models aim to predict the behaviour of surrounding vehicles using probabilistic frameworks. By understanding the likely actions of others, the AV can make safer lane-changing decisions. SUMO itself may not have built-in support for complex probabilistic models, but you can combine SUMO with external prediction tools or algorithms via TraCI.

- **Model Predictive Control (MPC) for Lane Changing**: MPC can be used to make lane-changing decisions by optimizing a set of objectives (like safety and comfort) over a prediction horizon. It accounts for the constraints of the vehicle and the road. While MPC logic would likely be external to SUMO, you can apply MPC-based decision-making in real-time using TraCI to influence vehicle manoeuvres in the simulation.

- **Deep Neural Networks and Imitation Learning**: Using deep neural networks, AVs can learn lane-changing behaviours by imitating human drivers. This approach is data-driven and requires extensive data sets of real-world driving. (Carl-Johan Hoel, 2018) With external tools and TraCI, you can implement neural network-based lane changing in SUMO. You'd train your model outside of SUMO and then use the trained model to make decisions during the simulation.

- **Game-Theoretic Models**: Game theory can model interactions between the AV and other road users. In a lane-changing scenario, the AV tries to anticipate the reactions of other drivers. (Meng Wang, 2015)SUMO doesn't have built-in game theory models for lane changing, but you can implement such logic externally and interface with SUMO using TraCI.

The development of lane-changing models for AVs is still an active research area, as the integration of different technologies, methods, and approaches presents challenges. As

AV technology matures, these models will play a crucial role in ensuring that these vehicles can navigate complex traffic situations safely and efficiently. while SUMO provides basic traffic dynamics and lane-changing models out of the box, for specialized models, particularly those tailored for AVs, integration using the TraCI interface is a common approach. This flexibility is one reason why SUMO is a popular choice for traffic simulation in academic and industrial research. In this thesis we use basic model of Reinforcement Learning-based Models as lane changing model.

CHAPTER 3: Methodology

Microsimulation is a tool for analyzing and understanding the behavior of complex systems, such as those found in the field of intelligent mobility. This technique involves creating a digital model of a real-world system and using it to simulate the interactions and behaviors of its components. (Alberto Bellini, 2023)

This research was part of my research in my internship at The LINKS Foundation in Torino. The proposed methodology involves studying the interactions between the Autonomous Shuttle of Navya, AVs and other road users in a simulated environment. Sumo used as a micro-simulator, to simulate the interactions between AVs and other road users in a simulated environment. SUMO used to develop a traffic simulation model. Subsequently, a dynamic control strategy was implemented to enhance automated shuttle operations across road networks. This meant searching for the effects of using the dynamic control strategy at various AV penetration levels, under diverse traffic conditions, and during different times of the day.

The simulation performed to evaluate the effect of using dynamic lanes can mitigate their impacts due to interactions with other vehicles in reducing the congestion while guaranteeing safety.

The selected area for simulation is part of center of Turin. The area selected encompasses a stretch of road that is bounded to the south by Corso Maroncelli, spanning between the intersections of Via Ventimiglia and Via Genova, and to the north by Corso Spezia (refer to Figures 1 and 2). This route included Corso Maroncelli, Via Genova, Corso Spezia, and Via Ventimiglia, covering a total distance of 4630 meters. We deliberately selected this network as it incorporates adjacent streets, ensuring the inclusion of real traffic dynamics from intersecting roads along the desired route. Due of the shuttle's brief maneuvers and the limited number of lanes in the northern section of the simulation, the shuttle's route is more confined. It is bounded to the south by Corso Maroncelli, between the intersections of Via Ventimiglia and Via Genova, and to the north by Via Valenza.

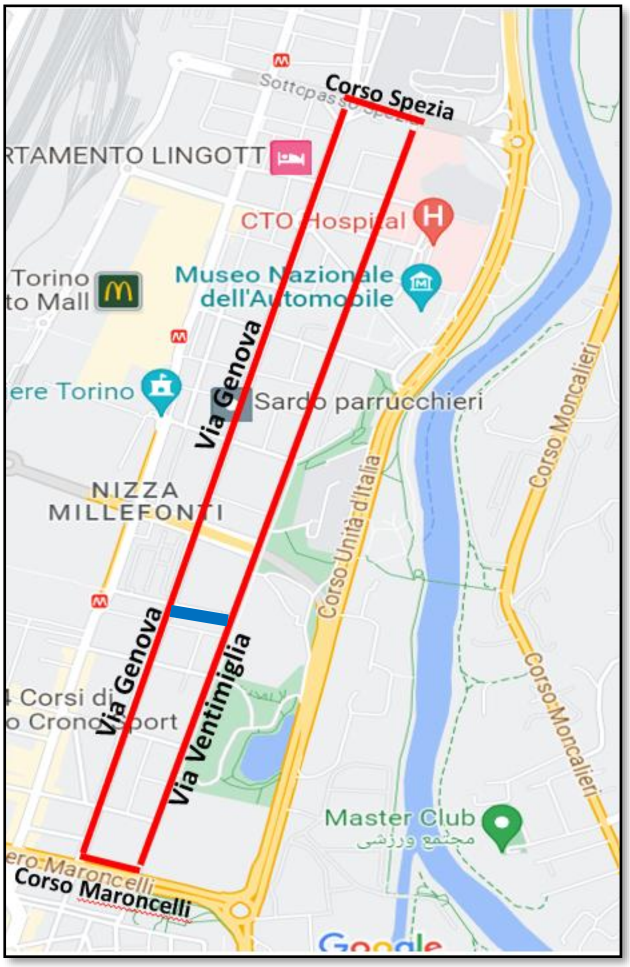


Figure 5: Simulation area in Torino

For the research, we executed a simulation encompassing three distinct traffic control strategies, each evaluated with four different penetration levels of Autonomous Vehicles (AVs). Every simulation run 10 times with different seeds according to existing random parameters in tripe generation and simulation. Aggregation of these results presented as output of simulation.

Traffic efficiency for vehicles is observed using delay time, speed and total travel time. parameters are disaggregated by vehicle type to understand whether the control strategies made had a positive effect on traffic in general or only on automated shuttles.

In our first traffic control strategy, we embraced a mix-traffic approach, featuring shuttles, AVs, and manually driven cars. This approach was devoid of any overt prioritization or restrictions on any vehicle type. All vehicle categories shared the road space harmoniously, without any one type receiving preferential treatment over the others. We selected this strategy as our base scenario, thereby setting it as a benchmark which the efficacy of the succeeding scenarios could be comparison.

In the second traffic control strategy is earmarking exclusive lanes for shuttles and public transportation in Corso Maronceli and Via Mentimilia, stretching from Via Valenza to Corso Maronceli. One notable repercussion of this lane allocation was the consequent reduction in lane availability for other vehicular types. The overarching goal was to amplify the priority of shuttles and public transport within the traffic network, acknowledging their pivotal role in sustainable urban mobility.

In the third traffic control strategy took a more targeted approach by implementing a lane reservation strategy. This was specifically applied to parts of our network, notably in Corso Maronceli and Via Mentimilia, stretching from Via Valenza right up to Corso Maronceli. This strategy can give a protection to the shuttle in order to increase the interest of the users, maintain smooth the operational speed of the shuttle and decrease the possibility of congestion. A tangential but equally crucial benefit was the anticipated reduction in congestion possibilities, ensuring a smoother traffic flow for all road users.

Through these meticulously designed strategies, our simulation aspired to unravel the complexities and interplay of various traffic dynamics in a rapidly evolving urban transportation landscape.

For different penetration levels of Avs, we choose scenarios, 1-without having AVs, 2-having 25% AVs in simulation 3-having 50% AVs in simulation and 4- Having 75% AVs in simulation.

Table 1: Parameters of the driver model used in SUMO simulations (Qiong Lu, 2019)

	Min Gap (m)	Accel (m/s^2)	Decel (m/s^2)	Emergency decel (m/s^2)	Sigma	Headway (s)	Max Speed (km/h)
Human Driver	**1.5**	**3.5**	**4.5**	**8**	**0.5**	**1.5**	**50**
Avs	**0.5**	**3.8**	**4.5**	**8**	**0**	**0.5**	**50**
Shuttle	**0.5**	**0.8**	**1.5**	**3**	**0**	**0.5**	**18**

- Mingap: the offset to the leading vehicle when standing in a jam
- Accel: the acceleration ability of vehicles
- Decel: the deceleration ability of vehicles
- Emergency Decel: the maximum deceleration ability of vehicles
- Sigma: the driver imperfection (between 0 and 1).
- Tau: the driver’s desired (minimum) time headway (reaction time) (in s).

One of the most significant contributors to traffic flow fluctuations in urban areas is rush hour traffic. Typically, there are two peak periods during the day when traffic congestion is at its highest: the morning rush hour (when people commute to work) and the evening rush hour (when they return home). These patterns are well-documented in urban transportation research. (Genevieve Giuliano, 2017) For our traffic simulations, we base our data on manual traffic counting. Using this real-world count, we then set different traffic inputs for the network. We use the 'jtrrouter.py' tool to create random trips. Based on this method, we input

10,000 vehicles per hour into the simulation network. On the main route, we've defined two separate traffic flow: one for AVs and another for vehicles driven by humans. Together, these flows account for a total flow of 2,000 vehicles per hour.

3-1 Simulation Framework and Tools

Besides the simulator availability, a very delicate aspect of a traffic simulation is related to the generation or availability of the scenario. Often SUMO is used to test particular situations and solutions that do not require a scenario bigger than some blocks or one or few junctions. For small scenarios like these ones, there is usually no particular problem in fixing the map and generating a synthetic traffic for the scope of the study. When instead the study necessitates to deal with a big scenario the preparing of the simulation gets much more complicated. Since many of the problems encountered and solutions found have been useful for the scope of the thesis, it is worth to rapidly discuss about the most significative ones. Most of the works dedicated to the big scenario generation were focused on the gathering of input data in order to build a realistic simulation, particularly they were focused on map and traffic generation. In the following parts simulation steps explained.

3-1-1 Selecting the simulation network

A SUMO network file describes the traffic-related part of a map, the roads and intersections the simulated vehicles run along or across. At a coarse scale, a SUMO network is a directed graph. Nodes, usually named "junctions" in SUMO-context, represent intersections, and "edges" roads or streets.

- every street (edge) as a collection of lanes, including the position, shape and speed limit of every lane,
- traffic light logics referenced by junctions,
- junctions, including their right of way regulation,
- connections between lanes at junctions (nodes).

SUMO road networks are encoded as XML files. The contents are grouped by the instances in the following order:

- cartographic projection valid for this network
- edges: at first, internal edges are given, then plain edges; each edge contains the list of lanes that belong to it
- traffic light logics

- junctions, including their right-of-way definitions; plain junctions first, then internal junctions
- connections, plain first, then internal
- optionally roundabouts

Generating road networks for SUMO (Simulation of Urban Mobility) traffic simulations can be accomplished in several ways, depending on the complexity of the network you need and the data sources available. there are some different methods for generating road networks for SUMO simulations like manual creation, using OSM, import map from other simulation software. (https://sumo.dlr.de/docs/index.html, n.d.)

Manual Creation:

SUMO uses XML files to define road networks. It is possible to manually create and edit these XML files to define roads, junctions, traffic lights, and other network elements. This method is suitable for small and simple road networks.

OpenStreetMap (OSM) Data:

OpenStreetMap is one of the most common methods to import road network data in sumo. OSM is a collaborative mapping platform that empowers individuals and communities to create, edit, and share detailed geographic information. OpenStreetMap, often referred to as the "Wikipedia of maps," is a collaborative project that enables users to contribute and edit geographic data. The map data on OSM is created by volunteers who survey, digitize, and input information about roads, buildings, landmarks, natural features, and more. This collaborative approach has led to the creation of a global map database that is open and accessible to everyone. (Haklay, 2010)

SUMO provides tools to convert OSM data into SUMO network files (e.g., .net.xml). OSM data often contains detailed road network information, including road types, lanes, and intersections. SUMO provides a tool called NETCONVERT that can convert network data from various formats (including OSM, Visum, VISSIM, and more) into SUMO network files. In our simulation we know our area and import the network from OSM map. a complete scenario can be built quickly and comfortably. The network will be imported with options and type maps suitable for the selected traffic modes. If more control is needed the options described below can be used.

NETEDIT:

SUMO's graphical user interface (GUI) includes a tool called NETEDIT, which allows to create, edit, and visualize network elements interactively. It can be a helpful tool for both manual creation and modification of networks.

Some traffic simulation software tools, like AIMSUN or VISSIM, allow to create road networks and export them in formats compatible with SUMO. This can be useful if you are already using these tools for traffic modeling.

Complex road networks often involve a combination of methods. For example, you might import a base road network from OSM data and then manually edit or extend it to suit your simulation requirements.

When generating road networks for SUMO, it's crucial to consider the specific goals of your simulation and the available data sources. The choice of method will depend on factors such as the network's size, complexity, realism, and the level of control you require over its details.

Import the map with OSM:

Import the map with OSM Web Wizard, (for the operation of this tool it’s necessary an old version of Python → version 2.7)

Passages to import the OSM map:

- open OSM Web Wizard
- select the area
- select the option and the characteristics of the network that are necessary to import
- click generate scenario

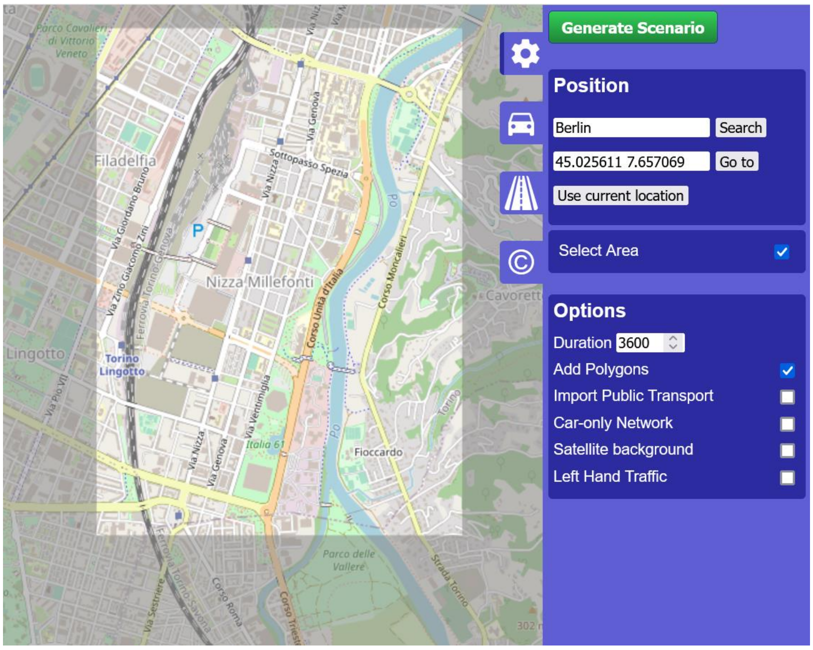


Figure 6: Screen of OSM Web Wizard for exporting the map of selected area

After the generation of the scenario with OSM Web Wizard, a new traffic simulation folder will be saved in the sumo folder address.

3-1-2 Modify network details

After importing a map from OpenStreetMap (OSM) into SUMO With OSM Web Wizard), may needs to make various modifications to the imported network to ensure it aligns with your simulation goals and requirements. Here are some common modifications you might need to make:

Lane Width: it needs to adjust lane widths to match real-world conditions or your specific simulation needs.

Lane Types: Modify Lane types to indicate different road types (e.g., residential, highway, etc.).

Speed Limits: Set speed limits for each road segment based on real-world data or desired simulation parameters.

Add or Modify Traffic Lights: SUMO may not always correctly detect and import traffic lights from OSM data. You may need to manually add or adjust traffic light positions and timings to accurately simulate traffic signal control.

Define Turn Lanes and Restrictions: Specify turn lanes and turning restrictions at intersections to control vehicle behaviour. You may need to define which lanes are allowed to turn in specific directions.

Customize Junctions: Junction geometries imported from OSM may not always be ideal for your simulation. You can modify junction shapes, angles, and sizes to better match real-world conditions or your research objectives.

Define Traffic Demand: need to specify traffic demand, including vehicle routes, origins, and destinations. You can use tools like "DUAROUTER" to generate demand files based on real-world traffic data or custom scenarios.

Remove Unnecessary Elements: OSM data can contain non-road elements like buildings, parks, and rivers. You should remove or ignore these non-road elements unless they are essential for your simulation.

Network Scaling and Translation: Adjust the scale and translation of the imported network to ensure it aligns with your simulation's coordinate system and geographic location.

Connectivity Checks: Validate the connectivity of your network to ensure that all roads and lanes are properly connected. SUMO requires a fully connected road network for accurate simulations.

Adding Custom Features: If the simulation requires custom features, such as bus stops, tram tracks, or pedestrian crossings, you'll need to add them manually.

Traffic Demand Adjustment: Fine-tune the traffic demand parameters, such as the number of vehicles, arrival rates, and vehicle types, to match your research objectives or real-world conditions.

The specific modifications required will depend on the complexity of your simulation and the quality of the OSM data you imported. It's essential to thoroughly review and validate the imported network and make adjustments as needed to ensure the accuracy and reliability of your SUMO simulation.

To modify the selected area, it is needed to go to *Edit → Open in netedit*. With this tool, it’s possible to modify the network and add features.

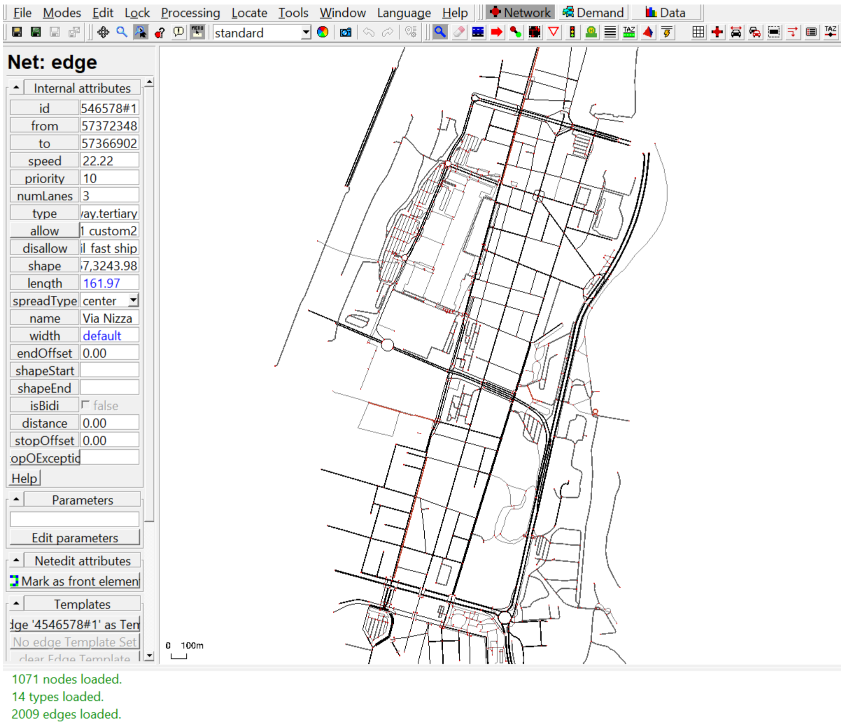


Figure 7: Visualisation of the osm map

3-1-3 Set the boundary conditions

To define the boundary condition, it's necessary to define the area of study. Then you have to delete all the objects outside the study area.

When you cut an object, you must pay attention to repristinate the conditions present outside the boundary. For example, if immediately outside the border there is an intersection, it's necessary to recreate a simple intersection inside the study area to say to the program that at that point are allowed some types of manoeuvres and there isn't a free flow condition but there are some traffic signals.

3-1-4 Set the maneuvers

To modify the allowed manoeuvres, it is needed to go *netedit*, Click on *Network* Network → *Set connection mode*

On the left of the screen will appear a series of commands with which it will be possible to program the connection of an edge.

To do this is necessary to select a lane on one edge and the lane in which you want to arrive on the other edge, defining the manoeuvre.

The starting lane will appear in light blue and the final one in green, some time will be not possible to select a final destination if the program sees a conflict point with another manoeuvre and this specific lane will appear in yellow.

To add a manoeuvre even if there is a conflict point click on the starting lane, press *ctrl,* and while holding down select the destination.

After any creation of a new connection, press the *OK* button. After the creation of all the connections press save.

To remove the connection, apply the same procedure. The program when you click on the destination selects the final lane if you click one and deselect if you click twice.

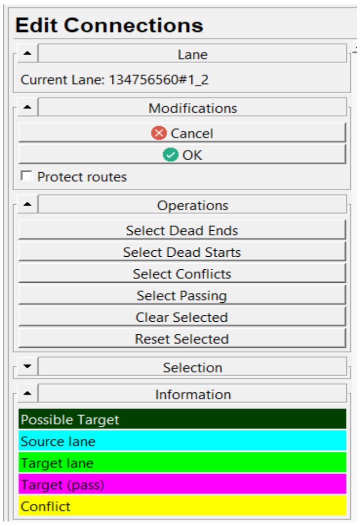


Figure 8: Modifying Intersection Maneuvers with NetEdit

3-1-5 Defining different types of vehicles

In SUMO, defining vehicle types is pivotal for generating accurate and realistic traffic simulations. These defining, which can be specified using XML files or NetEdit, represent different vehicles like cars, trucks, or specialized units like buses. Key attributes of these types include the vehicle's dimensions, speed limits, acceleration and deceleration rates, lane-changing behaviour, fuel consumption, emissions, and preferred routes. SUMO offers customization to simulate distinct behaviours of various vehicles, from passenger cars to autonomous vehicles. Each vehicle's behaviour, such as acceleration, affects its interaction within the simulation.

These may include:

- **Length:** The length of the vehicle.
- **Width:** The width of the vehicle.
- **Maximum Speed:** The maximum speed the vehicle can achieve.
- **Acceleration and Deceleration Rates:** Rates at the vehicle types.
- **Lane Change Parameters:** Parameters governing lane-changing behaviour.
- **Lane change model:** Fuel efficiency and consumption rates
- **Emissions:** Emission factors, representing pollution levels.
- **Routing Preferences:** Preferences for certain types of roads or routes.

SUMO also provides the flexibility to define custom vehicle types to meet specific research or scenario requirements. For instance, researchers can define autonomous vehicle types to simulate the behavior of self-driving cars.

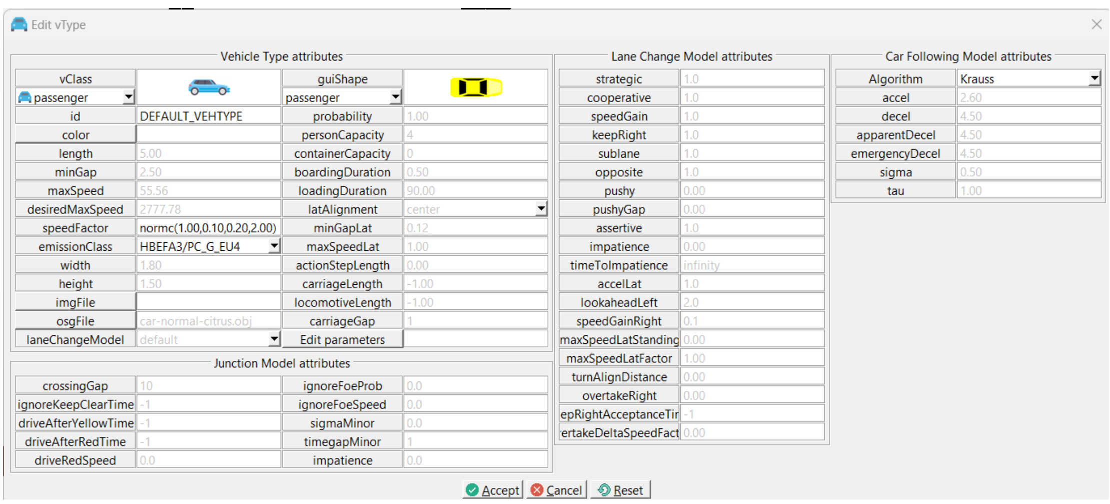


Figure 9: attributes editor of sumo's vehicle type in NetEdite

3-1-6 Modify the traffic signals

To modify the traffic light, it is needed to go to Edit → Open in netedit. Click on Network Network → Set traffic light mode .

So, in this way, all the signalized intersections will appear on the map. By clicking on this intersection, a series of commands will appear on the left of the screen that make it possible to program the intersection's phases.

It’s possible to set the type of traffic light (actuated, static, delay based), the offset, the duration of each phase and the maneuvers allowed in each signal group, and the min and max value of each interval.

Each letter written in the State column represents a single maneuver of that specific signal group. The maneuvers are reported counterclockwise, and each letter specifies if that specific maneuver has a green, yellow, or red phase.

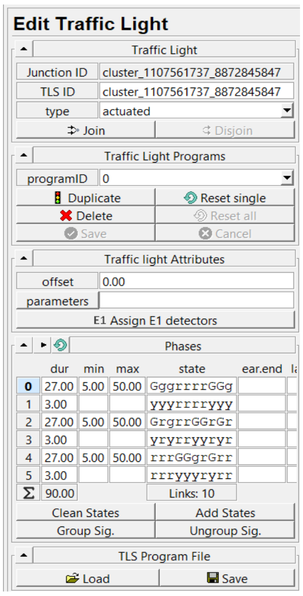


Figure 10: Tool to edit traffic lights in NetEdit

3-2 Definition of flow

Traffic flow are defined in SUMO's route definition files (typically with a .rou.xml extension). it specifies the type of each vehicle in this file, which associates them with a particular traffic flow type. There are some important parts in traffic flow:

- **Duration**: the time that vehicles will enter the simulation. This can be set using the **begin** and **end** attributes.
- **Frequency**: The number of vehicles introduced per given time unit can be defined. Instead of defining individual vehicles, flows allow you to introduce, say, 10 vehicles per minute between a specific time interval.
- **Route**: The path that the vehicles in the flow will take through the network. This is often defined separately in a route file and then referenced within the flow definition.

- **Vehicle Type**: Flows can also specify the type of vehicle (car, truck, bike, etc.) being introduced, including its maximum speed, acceleration, and other dynamic properties. These types can be predefined and then referenced in the flow.
- **Other Attributes**: There are many other attributes that can be specified for a flow, such as depart lane, depart position, depart speed, and more, which provide greater detail and specificity about how the flow behaves.

Vehicles assigned to different traffic flow types may exhibit distinct driving behaviors. For example, you can define a "car" traffic flow type with typical car-like behavior, a "bus" traffic flow type with characteristics specific to buses, and an "emergency" traffic flow type with behavior suited for emergency vehicles.

The behavior of each traffic flow type is determined by the parameters specified in the vehicle type definitions (usually in a add.xml file), such as maximum speed, acceleration, and deceleration. By categorizing vehicles into different traffic flow types, it is possible to control how they interact with the road network.

SUMO allows you to set specific routing rules for each traffic flow type to model their preferred routes and behaviors. This can be particularly important for public transport simulations, where buses or trams may have predefined routes and stops.

There is the flexibility to customize traffic flow types to represent various vehicle categories and their unique characteristics. This customization allows you to model a wide range of scenarios, from regular urban traffic to specialized situations like public transportation, emergency services, and freight transport.

there are multiple methods to define and generate traffic flows in sumo. These methods are chosen based on the nature of the traffic study and the availability of data.

1. **Direct Flow Definition:** This is one of the simplest ways to define traffic flows. In this approach, traffic flows are defined directly for specific routes in the simulation. You specify attributes like the starting time, ending time, the number of vehicles, and the route they should follow. This method is ideal when the exact paths and the number of vehicles that will traverse them over time are known.
2. **Origin-Destination (OD) Matrix:** An OD matrix is a representation of demand between various origin-destination pairs in the network. Each cell in the matrix indicates the number of trips (or vehicles) starting from a particular origin and heading towards a specific destination. SUMO provides tools to convert an OD matrix into a set of individual flows. This conversion considers the matrix values and possible routes between each

origin-destination pair. This method is particularly valuable when dealing with larger networks and when traffic demand data is available in an OD matrix format, which is common in transportation studies.

3. **Turn Percentages:** Instead of defining the exact routes or the OD pairs, this method focuses on specifying the percentage of vehicles that should take a specific turn at intersections. At every junction or intersection, you define the fraction (or percentage) of vehicles that should go straight, turn right, turn left, etc. these percentages and the input flow at the upstream edges, SUMO can distribute the vehicles at intersections accordingly. This approach is especially useful when detailed turn movement data is available at intersections, but exact route or OD pair data is not.
4. **Random Routes:** Defines a set number of routes randomly for each vehicle. Particularly useful when exploring various possible routes, a vehicle might take.
5. **Flow Distribution over Time:** Allows defining how traffic flow varies over time. In these methods, can simulate rush hours, off-peak times, etc.
6. **Public Transport Flows:** Define bus, tram, and other public transport lines. Specify timetables, stops, and frequencies.
7. **Traffic Assignment:** Based on iterative assignment approaches that consider congestion and route choices. Uses tools like DUE (Dynamic User Equilibrium) to adjust flows based on traffic conditions.
8. **Importing Real-world Data:** Incorporate datasets like mobile phone traces, GPS data, or traffic count data. Convert this data into flows or routes for the simulation.

Each method has its unique applications and is chosen based on the simulation's objectives and the available input data. SUMO's flexibility in handling diverse traffic demand definition methods makes it a powerful tool for a wide range of traffic simulation scenarios. Combining methods or refining them can also lead to more realistic and complex traffic simulations in SUMO.

3-3 Simulation time and step

The simulation time refers to the overall duration for which the simulation is set to run. Depending on the specific study or analysis, this could range from a short span of a few minutes to extended periods covering several hours of traffic behaviours. On the other hand, the simulation step, often termed as the time step, dictates the intervals at which the simulation's state updates. By default, SUMO updates its state every second, but this is adjustable based on the researcher's requirements.

In traffic simulations, ensuring accurate representation of real-world traffic behaviour requires careful attention to several temporal parameters. Among these, reaction time and action step length play a critical role.

Reaction time represents the delay between the occurrence of an event and the response to that event. In traffic simulations, this typically denotes the time it takes for a driver or an autonomous driving system to respond to changes in the traffic environment. It plays a crucial role in determining the responsiveness and realism of simulated entities. In human drivers, reaction times can vary based on factors like fatigue, distractions, or age. In simulations, this variance can be modelled to introduce a range of driver behaviours. Reaction time is crucial in scenarios like sudden braking, obstacle avoidance, or response to traffic signals.

The action step length works similar to a reaction time (vehicle will not react immediately to changes in their environment) but it also differs from a "true" reaction time because whenever a vehicle has its action step it reacts to the state in the previous simulation step rather than to the state that was seen in their previous action step. (https://sumo.dlr.de/docs/index.html, n.d.).

Action step length defines the discrete time intervals at which decisions or actions are evaluated and taken within the simulation. For instance, if the action step length is set to 1 second, the simulation will evaluate the status and make decisions every second. It dictates the granularity and precision of the simulation. A smaller action step length can result in a more detailed and fine-grained simulation but at the cost of higher computational demands. Conversely, a larger action step length simplifies the simulation, making it computationally efficient but potentially less accurate. Action step length is crucial in scenarios where rapid decisions are essential, such as complex intersections, merging onto busy highways, or in situations with closely spaced vehicles. Both reaction time and action step length need to be carefully selected based on the simulation's objectives. For studies focusing on detailed driver behaviour analysis, it might be beneficial to have a shorter action step length. On the other hand, for large-scale or macroscopic simulations, a more extended action step length might suffice.

By default, the action step length is equal to the simulation step length which works well for the default step length of 1s. When performing sub-second simulation by setting a lower step-length value, it may be useful to maintain a higher action step length in order to model reaction times and also in order to reduce computational demand and thus speed up the simulation. (https://sumo.dlr.de/docs/index.html, n.d.)

```
<time>
    <begin value="0"/>
    <end value="3600"/>
    <step-length value="0.8"/>
</time>
```

Figure 11: duration of simulation and simulation step

3-4 Additional files in SUMO

To add various infrastructures such as detectors, traffic lights, bus stops, etc., into a SUMO simulation, one must write the appropriate XML code for each element. Subsequently, the filename containing this code should be added to the 'additional_file' section of the SUMO configuration file.

```
<input>
    <seed value="5"/>
    <net-file value="osm.net.xml.gz"/>
    <route-files value="My_genereted_trips.rou.xml,route_new.rou.xml"/>
    <additional-files value="osm.poly.xml.gz,ptstops.add.xml"/>
</input>
```

Figure 12: additional file in Config file

CHAPTHER 4: Results

In this chapter we present the output of the simulation. the simulation time is 1 hour that with changing the simulation speed, each simulation takes around 2 minutes. By Using Traci library, we run each simulation 10 times with random seeds according to existing random parameters in trip generation and simulation. Output files of sumo simulation are in XML format. It presented according to the type of output for each vehicle, each simulation steps, each trip, each edge or each lane.

Then we use a python script to analysis and aggregation of these output to get the traffic monitoring parameter of average speed, total travel time, Total delay.

4-1 simulation outputs

SUMO is renowned for its extensive output capabilities that cater to a myriad of traffic and mobility analyses. One of its standout features is the detailed data it provides on individual vehicle movements during the simulation, commonly referred to as Floating Car Data (FCD). This data encompasses position, speed, lane, and other attributes at each time step. SUMO can output emission data, offering insights into the emissions generated by different vehicle types, such as CO2 and NOx. This is particularly invaluable for environmental impact studies.

Additionally, SUMO equips its users with data on traffic counts, showcasing the number of vehicles that pass through specified detectors or induction loops. Comprehending the efficiency and time consumption of various routes is also made possible through SUMO's ability to record travel times, detailing the duration taken by vehicles to commute between specific entry and exit points. For a broader perspective, the software provides statistics about traffic on network edges, encapsulating metrics like vehicle counts, average speeds, and waiting times. Moreover, those curious about the intricacies of vehicle routes will find SUMO's route data listing to be insightful, as it delves into the chosen routes of vehicles, aiding in analyzing route choice behaviors.

A pivotal aspect of traffic analysis, especially in urban settings, revolves around understanding congestion. SUMO facilitates this through its queue data, which provides intricate details about vehicle queues, especially at intersections or traffic hotspots.

The breadth and depth of these outputs, combined with the simulator's flexibility, empower stakeholders to undertake a variety of analyses. This includes understanding and pinpointing traffic congestion, evaluating the environmental consequences of vehicular flow, planning and forecasting infrastructure needs, determining the efficiency of public transport systems, and even testing the impacts of diverse traffic strategies. All in all, the expansive output capabilities and granular details offered by SUMO make it an exceptional tool in the realm of traffic simulation. (https://sumo.dlr.de/docs/index.html, n.d.)

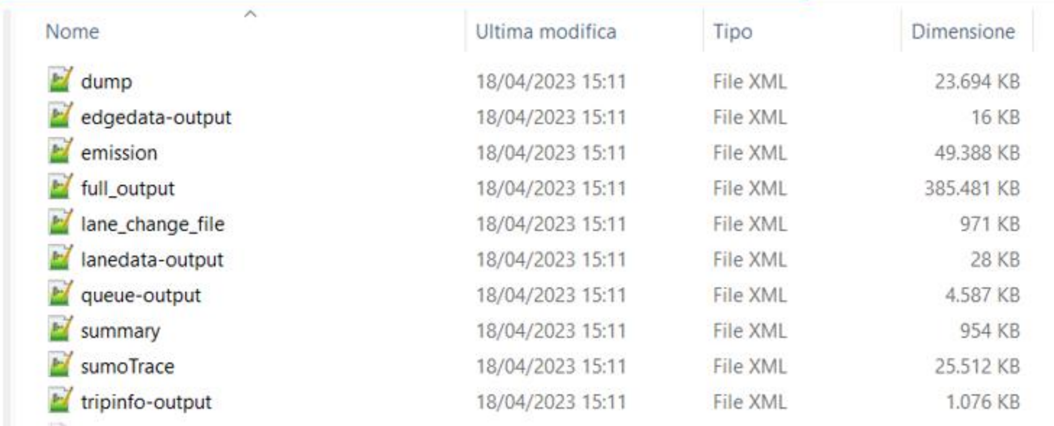
For getting output result of SUMO, it needed to insert type of the output in Config file to generate them after each run.

```
<report>
    <verbose value="true"/>
    <duration-log.statistics value="true"/>
    <no-step-log value="true"/>
</report>
<time>
    <begin value="0"/>
    <end value="3600"/>
</time>
<output>
    <netstate-dump value="dump.xml" />
    <fcd-output value="sumoTrace.xml"/>
    <emission-output value="emission.xml"/>
    <full-output value="full_output.xml"/>
    <lanechange-output value="lane_change_file.xml"/>
    <queue-output value="queue-output.xml"/>
    <tripinfo-output value="tripinfo-output.xml"/>
    <edgedata-output value="edgedata-output.xml"/>
    <lanedata-output value="lanedata-output.xml"/>
    <summary value="summary.xml"/>
    <!--
    <person-summary-output value="person-summary-output.xml"/>
    <movereminder-output value="movereminder-output.xml"/>
            -->
    <!--
    <vtk-output value="vkt_file.xml"/>
    -->
</output>

<gui_only>
    <gui-settings-file value="osm.view.xml"/>
</gui_only>
```

Figure 13: Code, to obtain the results, in Config file

All output files written by SUMO are in XML-format by default. However, with the python tool xml2csv.py allow to convert them to CSV format which can be opened with most spreadsheet software. After launching a sumo simulation, the file of the results will appear in the sumo folder in xml files.



Nome	Ultima modifica	Tipo	Dimensione
dump	18/04/2023 15:11	File XML	23.694 KB
edgedata-output	18/04/2023 15:11	File XML	16 KB
emission	18/04/2023 15:11	File XML	49.388 KB
full_output	18/04/2023 15:11	File XML	385.481 KB
lane_change_file	18/04/2023 15:11	File XML	971 KB
lanedata-output	18/04/2023 15:11	File XML	28 KB
queue-output	18/04/2023 15:11	File XML	4.587 KB
summary	18/04/2023 15:11	File XML	954 KB
sumoTrace	18/04/2023 15:11	File XML	25.512 KB
tripinfo-output	18/04/2023 15:11	File XML	1.076 KB

Figure 14: output Files of Sumo simulation

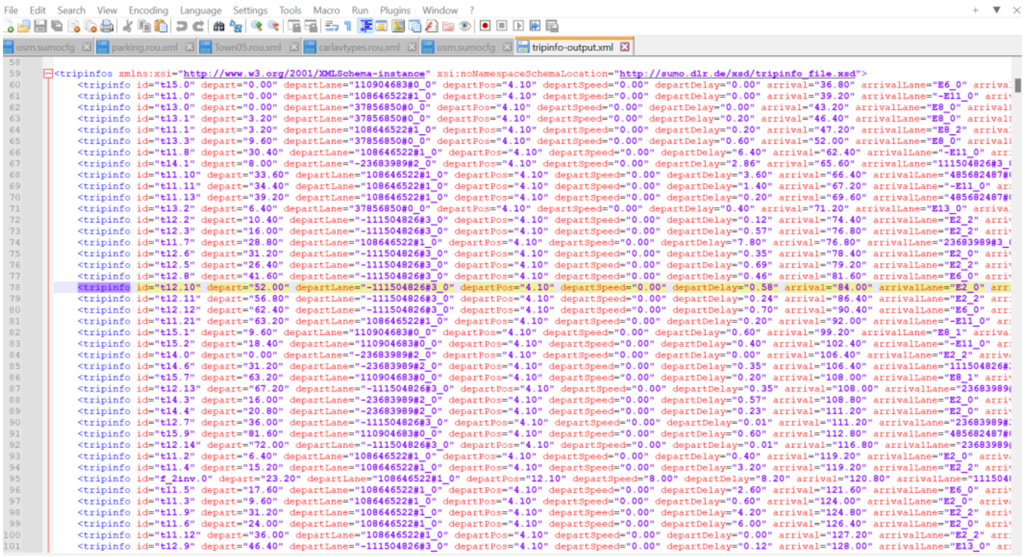


Figure 15:the XML output of the simulation – trip information for each vehicle

4-2 Run multiple with random seeds

In traffic simulations, various parameters may follow distinct distributions. To achieve greater accuracy in results, it is imperative to execute the simulation multiple times and then collate the outcomes. Multiple runs in traffic simulations are essential to capture the variability and uncertainty inherent in real-world traffic conditions. This approach offers a more robust understanding of potential scenarios and aids in the validation of the model. (Jaime Barceló, 2005)

Utilizing varied seeds for traffic simulations ensures the extraction of different yet replicable pseudo-random outcomes, a critical component in understanding the range of potential responses within the system. (Dynamic Network Traffic Assignment and Simulation

Methodology for Advanced System Management Applications, 2001) We implemented this by running SUMO with a range of seed values. utilizing varied seeds in traffic simulations significantly enhances the robustness of the findings, providing a more comprehensive understanding of potential traffic scenarios.

to obtain results from multiple simulation you have to write a code in the command prompt.

runSeeds.py -a sumo -k osm.sumocfg --seeds 1:10

the other way is using random seeds in Python script by Traci library and run multiple simulation.

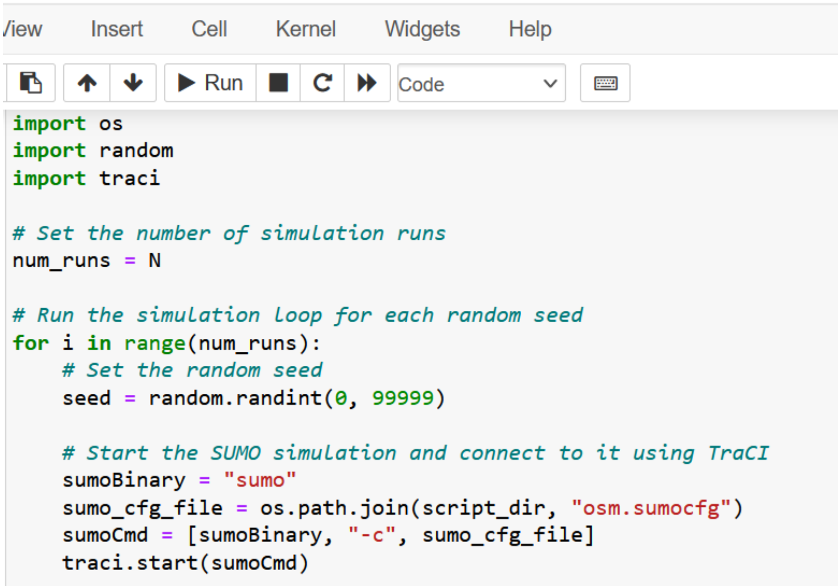
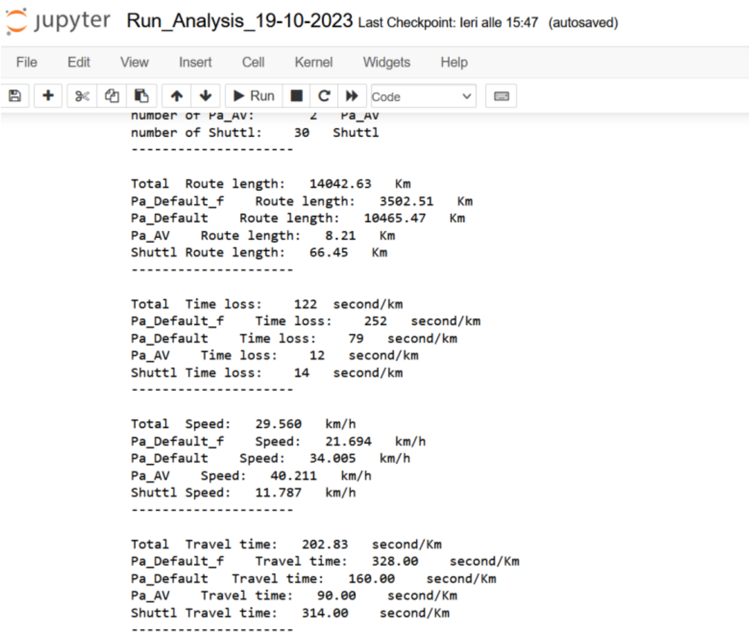


Figure 16: Python script of running multiple simulation and random seeds, without visualization

Also, by using "**Sumo**" instead of "**Sumo-gui**" in command "**sumoBinary**" we could get the results of multiple simulation without running the visual face of simulation and just export the results file.

4-3 Aggregate results

After executing several SUMO simulations and obtaining results, it's essential to consolidate the outputs. To achieve this, we developed a Python script to analyze the data, aggregate it, and measure key traffic parameters such as average speed, total travel time, Total delay and the number of lanes changing.




```
number of Pa_AV:       2    Pa_AV
number of Shuttl:     30    Shuttl
--------------------

Total  Route length:    14042.63   Km
Pa_Default_f    Route length:   3502.51   Km
Pa_Default    Route length:   10465.47   Km
Pa_AV    Route length:   8.21   Km
Shuttl Route length:   66.45   Km
--------------------

Total  Time loss:    122  second/km
Pa_Default_f    Time loss:    252   second/km
Pa_Default    Time loss:    79   second/km
Pa_AV    Time loss:    12   second/km
Shuttl Time loss:    14   second/km
--------------------

Total  Speed:   29.560   km/h
Pa_Default_f    Speed:   21.694   km/h
Pa_Default    Speed:   34.005   km/h
Pa_AV    Speed:   40.211   km/h
Shuttl Speed:   11.787   km/h
--------------------

Total  Travel time:   202.83   second/Km
Pa_Default_f    Travel time:   328.00   second/Km
Pa_Default    Travel time:   160.00   second/Km
Pa_AV    Travel time:   90.00   second/Km
Shuttl Travel time:   314.00   second/Km
--------------------
```

Figure 17: Analysis output of the simulation

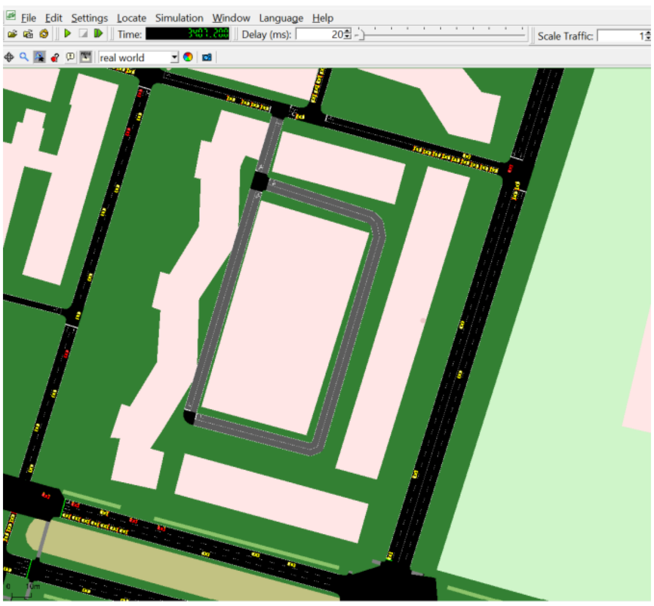


Figure 18: Sumo simulation – Red vehicle show AVs in simulation

4-4 Simulation Results

In Table 2 the results of Travel time of sumo simulation presented. The presented table, Table 2, offers invaluable insights into the travel time for different vehicle types under various traffic strategies and AVs integrations. The scenarios span three distinct traffic strategies: Mixed Traffic, Separated Lane, and Dynamic Lane, and within each, the penetration of AVs is varied from 0% to 75%.

In the Mixed Traffic approach, there's an observable trend: as the percentage of AVs increases, the travel time for human-driven vehicles consistently decreases. Starting at 152 seconds/km with no AVs, the travel time goes down to 111 seconds/km at 75% AV integration. This suggests that AVs might be facilitating smoother traffic flow in the mixed scenario. For shuttles, a slight decline in travel time is also noted, moving from 348 to 309 seconds/km. AVs, when introduced, maintain a fairly consistent travel time, fluctuating around the 106 seconds/km mark, indicating their ability to sustain consistent speeds regardless of their numbers.

Moving to the Separated Lane strategy, there's a stark difference when compared to the mixed traffic. At 0% AVs, the travel time for human-driven vehicles soars to 328 seconds/km. However, as the AV percentage increases, there's a substantial decrease in their travel time, dropping to almost half at 50% and then levelling off around 166 seconds/km by 75%. The shuttle travel time remains largely unchanged across all scenarios, while AVs experience varied travel times, starting at a high of 150 seconds/km at 25% and then decreasing to around 110 seconds/km by 75%. This might hint at the effective utilization of the separated lanes by the AVs as their numbers grow, improving their overall efficiency.

Table 2:Simulation Results for Travel time (Second/km)

Scenario	Traffic Strategy	AVs Percentage	Human-Driven	Shuttle	AVs
1	Mixed traffic	0%	152	348	-
2	Mixed traffic	25%	120	317	107
3	Mixed traffic	50%	116	313	106
4	Mixed traffic	75%	111	309	104
5	Separated lane	0%	328	314	-
6	Separated lane	25%	314	310	150
7	Separated lane	50%	198	308	122
8	Separated lane	75%	166	314	110
9	Dynamic Lane	0%	247	325	-
10	Dynamic Lane	25%	224	312	126
11	Dynamic Lane	50%	210	305	112
12	Dynamic Lane	75%	183	303	105

Lastly, within the Dynamic Lane strategy, human-driven vehicles start at 247 seconds/km at 0% AVs. As AV integration increases, their travel time sees a consistent drop, reaching 183 seconds/km by 75%. Shuttles, on the other hand, maintain their travel time around the 310 seconds/km mark, showcasing a minor decline. AVs, once introduced, show a diminishing travel time trend, going from 126 to 105 seconds/km as their presence grows.

Table 3 provides a comprehensive look at the delay times (seconds per km) experienced by various vehicle types under different traffic strategies and degrees of Avs penetration. The data spans across three distinct traffic strategies: Mixed Traffic, Separated Lane, and Dynamic Lane, with AV percentages ranging from 0% to 75%.

In the Mixed Traffic strategy, there's a clear trend for human-driven vehicles. Starting at a delay time of 77 seconds/km when no AVs are present, this delay consistently drops as more AVs are integrated, reaching a low of 36 seconds/km at 75% AV penetration. This indicates that as AVs are introduced into mixed traffic, they may help in decreasing congestion or facilitating smoother traffic flow. Shuttles see a significant decrease in delay time, starting from 42 seconds/km at 0% AVs and plunging to just 9 seconds/km at 75% AVs. AVs, when part of the traffic, have delay times hovering between 32 and 29 seconds/km, suggesting their consistent performance across varying penetration levels.

Under the Separated Lane strategy, human-driven vehicles experience a pronounced delay at 0% AV integration, with a time of 252 seconds/km. As AVs are gradually introduced, there's a significant drop in these delays. By the time AVs constitute 75% of the traffic, the delay for human-driven vehicles stands at 91 seconds/km. Shuttle delays remain relatively constant, hovering around the 10 to 14 seconds/km range, while AVs experience varied delay times, which decrease from 55 seconds/km at 25% AV presence to 33 seconds/km at 75%.

For the Dynamic Lane approach, human-driven vehicles start with a delay of 110 seconds/km at 0% AVs. As the AV integration increases, these delays consistently diminish, landing at 83 seconds/km by the 75% AV scenario. Shuttles experience a marked decrease in delay, moving from 37 seconds/km to 17 seconds/km as the AV percentage grows. AVs, once part of the traffic, have delay times decreasing from 43 seconds/km at 25% to 30 seconds/km at 75%.

Table 3: Simulation Results for Delay (Second/km)

Scenario	Traffic Strategy	AVs Percentage	Human-Driven	Shuttle	AVs
1	Mixed traffic	0%	77	42	-
2	Mixed traffic	25%	44	17	32
3	Mixed traffic	50%	40	12	31
4	Mixed traffic	75%	36	9	29
5	Separated lane	0%	252	14	-
6	Separated lane	25%	239	10	55
7	Separated lane	50%	122	8	43
8	Separated lane	75%	91	13	33
9	Dynamic Lane	0%	110	37	-
10	Dynamic Lane	25%	97	26	43
11	Dynamic Lane	50%	87	23	37
12	Dynamic Lane	75%	83	17	30

Table 4 offers presented the average speeds (km/h) of different vehicle types under three specific traffic strategies: Mixed Traffic, Separated Lane, and Dynamic Lane. This is viewed against various AVs penetration percentages.

Within the Mixed Traffic strategy, the average speed of human-driven vehicles reveals a pattern of improvement. Beginning at 28.8 km/h without AVs on the road, there's a noticeable increase in speed as more AVs join the traffic, culminating in an average speed of 34.6 km/h at 75% AV penetration. This suggests that AVs, when incorporated into mixed traffic, can lead to smoother traffic flows and higher speeds for human-driven vehicles. The speed for shuttles also sees a positive trend, from 10.0 km/h at 0% to 11.9 km/h at 75% AVs. AVs, once introduced, exhibit speeds ranging from 35.0 to 35.9 km/h, indicating a consistent performance across varying percentages of AV presence.

In the Separated Lane strategy, human-driven vehicles have a lower average speed at 21.3 km/h when no AVs are included. However, as AVs penetration increases, the speeds increased, reaching an average of 28.1 km/h at 75% AV integration. Shuttle speeds remain fairly stable across this strategy, lingering around the 11.7 to 11.8 km/h range. AVs, upon their inclusion, have speeds that progressively rise from 28.6 km/h at 25% to 34.3 km/h at 75% AV presence.

The Dynamic Lane strategy sees human-driven vehicles starting at an average speed of 26.5 km/h in a no-AV scenario. This figure steadily rises as AVs are integrated, with the speed hitting 31.4 km/h by the time AVs constitute 75% of traffic. Shuttles remain quite consistent in this strategy, with speeds fluctuating minimally around the 11.6 to 11.7 km/h mark. AVs, once introduced into the mix, display a speed growth from 32.2 km/h at 25% AV penetration to 35.0 km/h at 75%.

Table 4: Simulation Results for Average Speed in network (Km/h)

Scenario	Traffic Strategy	AVs Percentage	Human-Driven	Shuttle	AVs
1	Mixed traffic	0%	28.8	10.0	-
2	Mixed traffic	25%	32.8	11.6	35.0
3	Mixed traffic	50%	33.5	11.7	35.2
4	Mixed traffic	75%	34.6	11.9	35.9
5	Separated lane	0%	21.3	11.7	-
6	Separated lane	25%	21.7	11.8	28.6
7	Separated lane	50%	24.6	11.8	32.7
8	Separated lane	75%	28.1	11.8	34.3
9	Dynamic Lane	0%	26.5	11.6	-
10	Dynamic Lane	25%	28.3	11.6	32.2
11	Dynamic Lane	50%	30.3	11.6	33.1
12	Dynamic Lane	75%	31.4	11.7	35.0

CHAPTHER 5: Conclusions:

AVs represent a frontier of innovation in urban mobility, promising transformational change in the dynamics of urban traffic. This research, conducted through a partnership with The LINKS Foundation in Torino, employed microsimulation tools, specifically SUMO, to analyze the interplay between AVs and conventional vehicles under diverse traffic control strategies within a defined area of Turin's city center. The study's central focus was on evaluating the implications of integrating the Navya autonomous minibus and different AV penetration levels by using SUMO microstimulator.

The results presented across Tables provide the influence of AVs on urban traffic dynamics across various traffic strategies.

1. Travel Time: Within the Mixed Traffic strategy, as the AV percentage increases, there's a noticeable decrease in travel time for both human-driven vehicles and shuttles. This trend suggests that incorporating more AVs into a mixed traffic setting can lead to enhanced road efficiency and reduced travel times. Similarly, in the Separated and Dynamic Lane strategies, higher AV percentages tend to result in reduced travel times for human-driven vehicles. However, the shuttle timings are more consistent, and AVs show variability based on the specific strategy adopted.

2. Delay: The delay data is particularly illuminating. For Mixed Traffic, as AV penetration grows, there's a clear reduction in delay for human-driven vehicles, which is a strong testament to the potential of AVs to reduce congestion and streamline traffic. Shuttles, too, benefit from reduced delays as AV percentages rise. In the Separated Lane strategy, while human-driven vehicles experience a pronounced delay at 0% AV presence, this delay considerably drops as more AVs are integrated. The Dynamic Lane strategy reflects a similar pattern, underscoring the efficiency gains realized from the strategic management of AVs in traffic.

3. Average Speed: The average speed of vehicles provides insights into road utilization efficiency. Under Mixed Traffic conditions, the speed of human-driven vehicles improves significantly with increasing AV percentages. This trend is mirrored in the Separated and Dynamic Lane strategies as well. The consistency of the shuttle speeds across all strategies suggests they're less influenced by AV percentages, whereas AVs tend to exhibit higher speeds, reflecting their capacity for optimal navigation and movement in diverse traffic scenarios.

These tables collectively emphasize the transformative potential of AV integration in urban traffic scenarios. Across all examined metrics – be it travel time, delay, or average speed – the presence of AVs generally correlates with improved traffic efficiency and flow. The data thus highlights the imperative for cities and urban planners to invest in infrastructure and strategies that facilitate the seamless integration of AVs. Such measures not only cater to the technological evolution of transportation but also promise tangible benefits in terms of reduced congestion, efficient road use, and a more streamlined urban mobility landscape.

Different control strategies, from mixed traffic without any prioritization to dedicated lanes for AVs and public transportation, were assessed. While each strategy has its merits, the data indicates that strategies which leverage AV's inherent capabilities to reduce congestion and maintain consistent speeds lead to the most significant improvements in overall traffic flow.

It's evident that as the percentage of AVs increases, the traffic system's efficiency improves. This accentuates the need for progressive AV integration in urban settings, advocating for a gradual shift towards a higher AVs presence on roads.

It's important to note that no simulation can perfectly replicate real-world conditions, so while simulation tools are invaluable in the research phase, on-road testing under controlled conditions is essential before wide-scale deployment of AV technologies.

CHAPTHER 6: Future work

In our upcoming projects at Links Foundation, we are pivoting towards the reception and analysis of real-time data sourced from both the Traffic Centre of Torino and various connected vehicles. This proactive shift is aimed at formulating and implementing strategies for Traffic Flow Optimization by harnessing the power of Data Analytics. Moreover, we are aspiring to enhance Dynamic Route Guidance and Navigation systems. Our primary objective in this endeavour is to analyse real-time traffic data to identify the most efficient routes for vehicles.

The real-time data provide, stemming from connected vehicle and their continuous interaction with other vehicles, infrastructure, and even pedestrians, offers a more dynamic approach to simulating and managing urban traffic scenarios. When integrated into SUMO, this real-time data can yield more accurate, responsive, and adaptive traffic models.

SUMO simulations can be calibrated and validated using the real-time data streams. By continuously comparing simulation outputs with real-time data. It can ensure that the simulation remains an accurate reflection of the current traffic scenario. Also, by analysing patterns from historical and real-time data, SUMO can predict potential traffic bottlenecks, rush hour surges, or even identify areas prone to frequent accidents.

To achieve this, and given the computational demands of such real-time analytics, we foresee a requirement for parallel simulations on High-Performance Computers (HPC). High-Performance Computing (HPC) is at can process vast amounts of data at paralleled speeds, which means complex computations that might take days or even months on standard computers can often be completed in hours or minutes on HPC systems. These simulations can provide insights that are impossible or impractical to obtain through real-world experiments.

The other steps will be connected simulation of CARLA and Sumo. CARLA (Car Learning to Act) is an open-source platform designed to assist the research and development of autonomous driving systems. CARLA developed to support development, training, and validation of autonomous urban driving systems, by providing a rich variety of complex environments and traffic situations.

While CARLA excels at high-fidelity vehicle dynamics and sensor simulation, SUMO excels at modelling large-scale traffic scenarios, vehicle behaviours, and traffic management strategies. The integration or connected simulation of CARLA and SUMO brings together the strengths of both simulators. By connecting the two, it's possible to embed a detailed autonomous vehicle simulation (from CARLA) within a broader urban traffic context (from

SUMO). the connected simulation of CARLA and SUMO provides a powerful and flexible platform for autonomous vehicle research, bridging the gap between individual vehicle behaviours and city-scale traffic dynamics.

Figure 19: Mix simulation of Sumo and Carla

References

1. Alberto Bellini, A. C. (2023). The use of dynamic lane reservation for autonomous shuttles in urban traffic. 15th ITS European Congress.

2. Arne Kesting, M. T. (2010). Enhanced Intelligent Driver Model to Access the Impact of Driving Strategies on Traffic Capacity. *Philosophical Transactions of the Royal Society A 368, 4585-4605* .

3. Ashesh Jain, H. S. (2015). Car that Knows Before You Do: Anticipating Maneuvers via Learning Temporal Driving Models.

4. *Automated Driving Systems 2.0: A Vision for Safety.* (2021). National Highway Traffic Safety Administration (NHTSA).

5. Barceló, J. (2010). *Fundamentals of Traffic Simulation.* Springer.

6. By Harding, J., Powell, G., & Yoon, R. (2014). *Vehicle-to-vehicle communications : readiness of V2V technology for application.* United States. National Highway Traffic Safety Administration.

7. Carl-Johan Hoel, K. W. (2018). Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning. *IEEE International Conference on Intelligent Transportation Systems (ITSC)*.

8. Chrysostomos, C. (2019). *3D Traffic Simulation In Unity: How to Use Sumo to Produce 3D Traffic in Unity.* Cyprus.

9. Dynamic Network Traffic Assignment and Simulation Methodology for Advanced System Management Applications. (2001). *Networks and Spatial Economics*.

10. Genevieve Giuliano, S. H. (2017). *The Geography of Urban Transportation.* Guilford Publications.

11. Haklay, M. (2010). How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets. *Environment and Planning B: Urban Analytics and City Science*.

12. Hanna Krasowski, X. W. (2020). Safe Reinforcement Learning for Autonomous Lane Changing Using.

13. Hidas, P. (2002). Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C Emerging Technologies 10(5-6): 351-371*.

14. *https://sumo.dlr.de/docs/index.html*. (n.d.).

15. *https://sumo.dlr.de/docs/TraCI.html*. (n.d.).

16. *https://www.dlr.de/ts*. (n.d.).

17. *https://www.navya.tech/en/*. (n.d.).

18. Jaime Barceló, J. C. (2005). *Simulation Approaches in Transportation Analysis.* Simulation Approaches in Transportation Analysis.

19. James M. Anderson, N. K. (2014). *Autonomous Vehicle Technology: A Guide for Policymakers.* RAND Corporation.

20. Kenney, J. B. (2011). Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proceedings of the IEEE*.

21. Krauss, S. (1998). Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics.

22. M. Kyriakidis, R. H. (2015). Public opinion on automated driving: Results of an international questionnaire among 5000 respondents. *Transportation Research Part F: Traffic Psychology and Behaviour*.

23. Ma, J. G. (2019). A Review of Dynamic Traffic Signal Control Methods. *IEEE Access, 7, 125236-125254*.

24. Martin Treiber, A. H. (2000). Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E 62, 1805-1824*.

25. Martin Treiber, A. K. (2013). *Traffic Flow Dynamics Data, Models and Simulation.* SpringerLink.

26. Meng Wang, S. H. (2015). Game theoretic approach for predictive lane-changing and car-following control. *Transportation Research Part C Emerging Technologies*.

27. Mustapha Saidallah, A. E. (2016). A Comparative Study of Urban Road Traffic Simulators. *5th International Conference on Transportation and Traffic Engineering (ICTTE 2016).*

28. Pablo Alvarez Lopez, ,. M.-W. (2018). Microscopic Traffic Simulation using SUMO. *21st International Conference on Intelligent Transportation Systems (ITSC).*

29. Qiong Lu, T. T. (2019). The impact of autonomous vehicles on urban traffic network capacity: an experimental analysis by microscopic traffic simulation. *Transportation Letters The International Journal of Transportation Research*.

30. Shladover, S. E. (2016). *The Truth about “Self-Driving” Cars.* Scientific American.

31. Tomer Toledo, H. N.-A. (2003). Modeling Integrated Lane-Changing Behavior. *Transportation Research Record: Journal of the Transportation Research Board*.

32. Xiangquan Chen, Z. W. (2023). Modeling Mixed Traffic Flow with Connected Autonomous Vehicles and Human-Driven Vehicles in Off-Ramp Diverging Areas. *Intelligent Vehicle-Infrastructure System and Sustainable Transportation*.