# ****Abstract****

For this study…

# ****Introduction****

**Intro here…**

# ****System Model****

## Vehicle Model

The vehicle emission model used in these scenarios is a simplified version of PHEM (Passenger Car and Heavy Duty Emission Model) developed by the Graz University of Technology in 1999, which is based on extensive empirical data collected from passenger cars, light duty and heavy duty vehicles. The model is based on the vehicle's speed and acceleration, which are calculated at a frequency of one second intervals. The mapping of these parameters to fuel efficiency is illustrated in Figure ???. The importance of this model lies in its ability to provide an accurate representation of fuel consumption, allowing for meaningful comparisons to be made between different simulation runs.

[PHEMlight - SUMO Documentation (dlr.de)](https://sumo.dlr.de/docs/Models/Emissions/PHEMlight.html)

Chart, histogram

Description automatically generated

## Environment

SUMO (Simulation of Urban MObility) was used to create the simulation environment. SUMO is an open-source traffic simulation tool that is used to model and analyze the behavior of road traffic in urban environments.

A signalized corridor of traffic lights was generated with randomized light positions for every simulation run. The SPaT data for the studied traffic flow direction was kept at the values listed in Table 1 to simplification purposed. Future work includes adding variable SPaT data and feeding it as observation to the reinforcement learning agent.

|  |  |
| --- | --- |
| **Light Phase** | **Light Timing** |
| Red | 20 seconds |
| Green | 30 seconds |
| Yellow | 3 seconds |

The speed limit for all roads in the simulation was kept to 20 m/s (45 mph).

Traffic flow was also randomly generated at the beginning of each simulation. Vehicle number and positions were all randomly determined at the initiation of the scenario. These simulated vehicles attempted to maintain a constant speed of 20 m/s, but this speed was governed by the proximity to other vehicles and the intersections. When approaching another vehicle or red light, the vehicles would command a constant deacceleration of ??? m/s2. To increase speed when not at 20 m/s, a constant acceleration of ??? m/s2 was used.

A baseline vehicle will be referenced that refers to a vehicles spawned into the scenario with the same vehicle dynamics as the surrounding traffic flow, and would therefore attempt to drive 20 m/s. This vehicle is meant to simulate human-driver behavior, as the baseline vehicle has no knowledge of the SPaT data, only knowledge of the current phase color.

# Control Algorithms

Control algos…

## Training Process

1. Build a traffic corridor with randomized intersection positions
2. Deploy a randomized number of cars on the corridor
3. Run a simulation step (the time between intersection)

DQN Steps

1. Initialize Q-Network and Target network with randomize weights (θ)
2. Select either the optimal action or a random action based on the greedy epsilon factor (ε).
3. Observe the next state and reward
4. Store the experience in the replay memory
5. Sample a random batch of experiences from the replay memory
6. Compute the target Q-value for the next states using the target network
7. Update the Q network using the computed target Q-values and loss function
8. Update the target network by copying the weight from the Q network every N episodes to stabilize the learning process.
9. Repeat steps 2 through 8 until the maximum number of episodes is reached.
10. Use the Q network to predict the optimal action based on unseen states.

# Simulation Results and Discussion

## Scenario without Traffic

Two RL agent implementations were tested on the simulations without traffic. These two vehicles were compared to the baseline.

Both agents received the same observations. However, the first agent commanded an acceleration for the vehicle at every 1 second interval.

Timeline

Description automatically generated

The downside of the first agent is that it has to output many accelerations throughout the course of the scenario, so training the agent to an optimal values take a lot more simulation time.

The second agent commanded a target velocity (action) at every intersection (Figure ???). The vehicle will attempt to hit this target velocity using an acceleration of ??? and deacceleration of ???. It is governed by the same dynamics as the surrounding vehicles, so it will slow down before colliding with another vehicle or running a red light. This behavior provides a sandbox for the RL agent to explore where it will not be trying fruitless actions.

Timeline

Description automatically generated

The downside to the second agent is that is only makes decisions at each intersection, so if something happens between intersections, a more efficient speed will not be commanded for the new situation.

## Scenario with Traffic

# Conclusion

# References

SUMO:

[“Microscopic Traffic Simulation using SUMO”](https://ieeexplore.ieee.org/document/8569938) ; Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. IEEE Intelligent Transportation Systems Conference (ITSC), 2018.

PHEMlight:

https://www.itna.tugraz.at/assets/files/areas/em/PHEM\_en.pdf