Università di Pisa
Dipartimento di Ingegneria dell'Informazione

Performance Evaluation of Computer Systems and Networks

Project 1 – Vehicular network: car sensing

Tesfaye Yimam Mohammad

Introduction

Project Description

Part I:- A vehicular system is composed of N vehicles that move randomly within a 2D floorplan of size L× H, according to a waypoint model.

Part II:- Every T seconds each vehicle checks how many cars are within its transmission range. The relationship between T and M is expressed as $T = \propto \times M*M$.

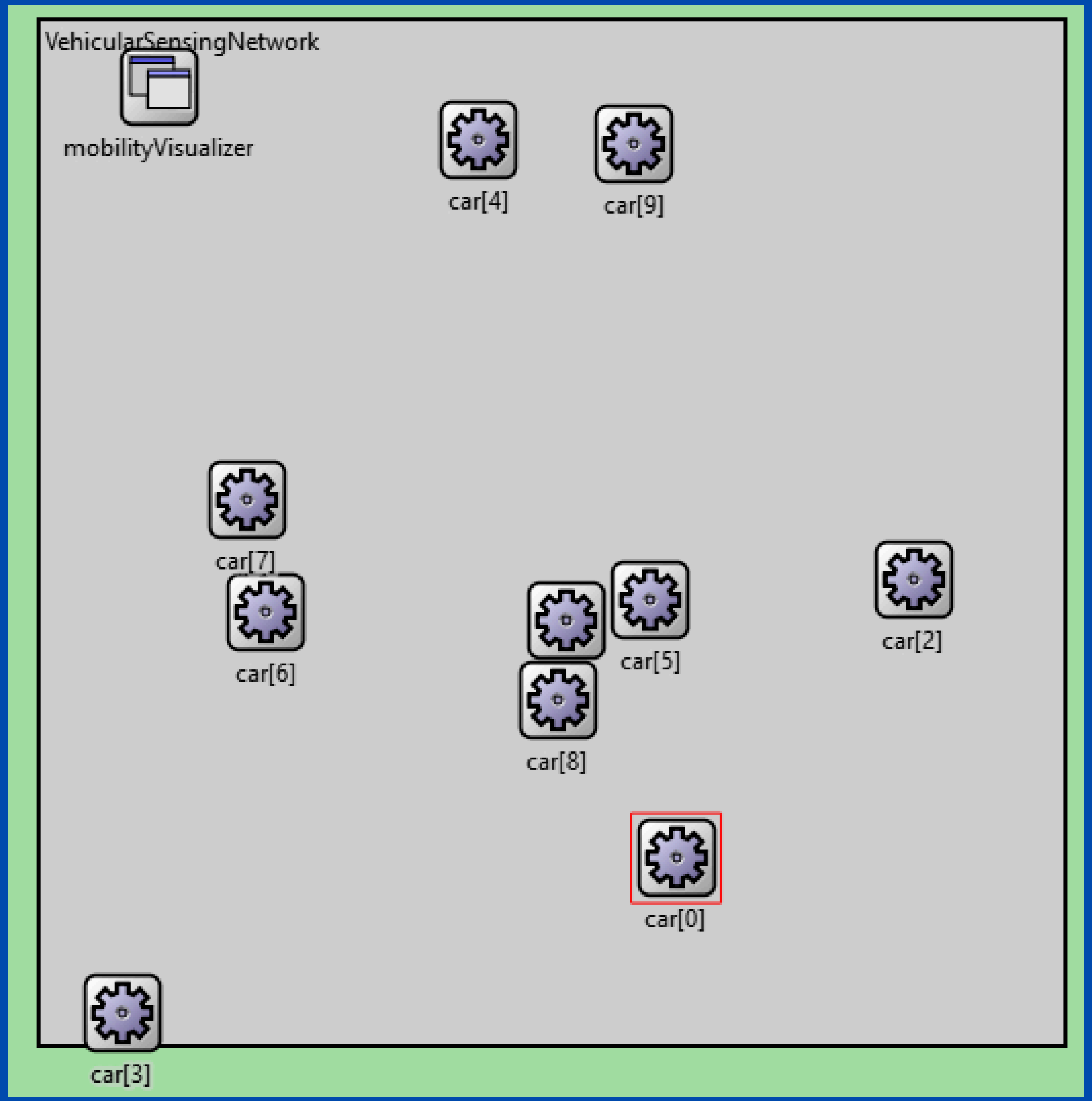
-> α assume values between 0 and 1.

Confidence Interval

-> 95% of confidence level is used in the analysis of the simulation results.

Objective of the Project

Evaluate at least the overall rate of vehicles sensed per second for various values of M and α.



Implementation: Modules

Car:- A compound module that represents the system that we are dealing with in the project. It has the following sub-modules.

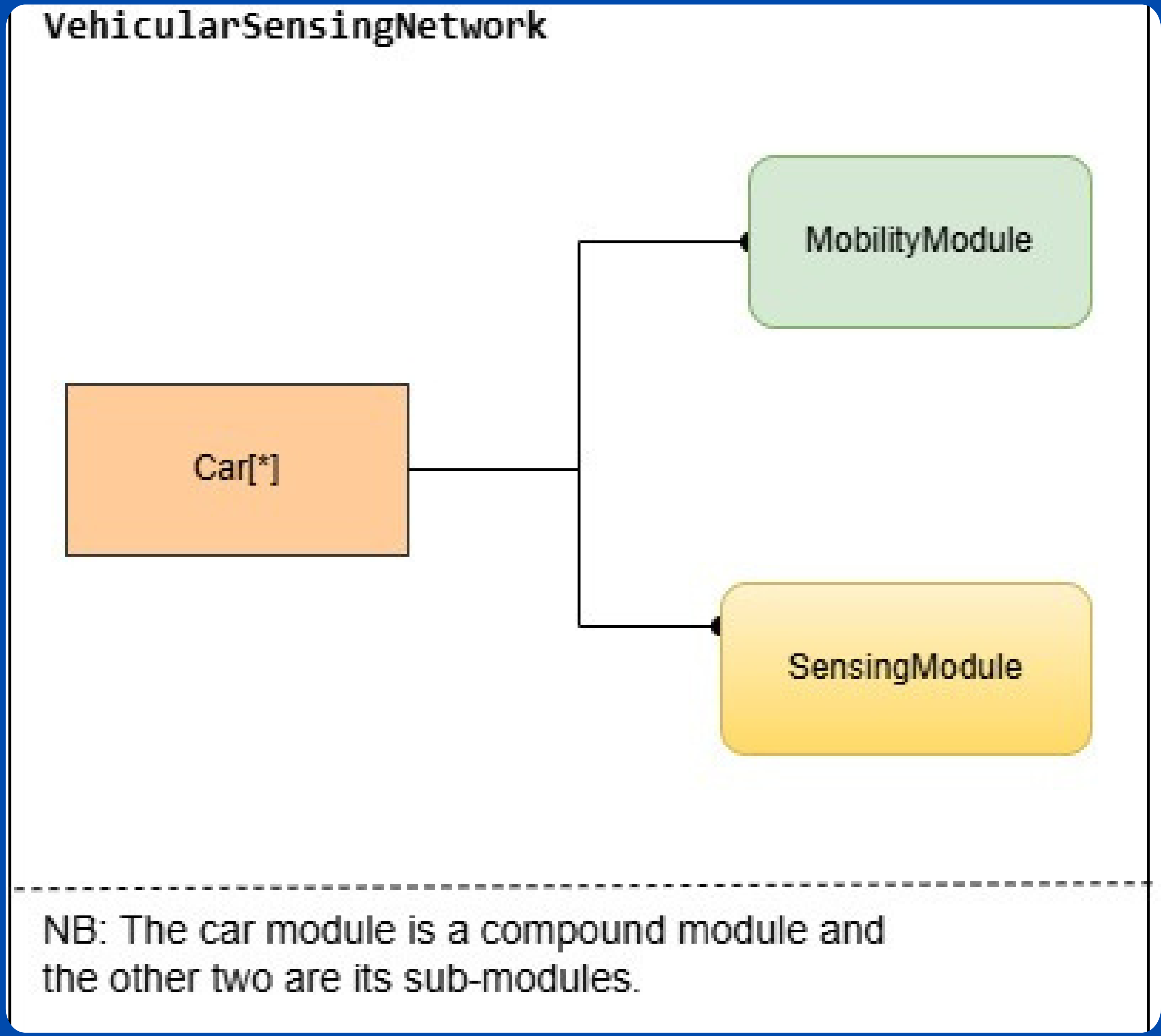
– **MobilityModule** :- a simple module.

-> Extends the **RandomWaypointMobility** from INET to accesses all the necessary parameters to be used in the project.

– **SensingModule** :- is a simple module

-> provides all the parameters that are needed to implement the sensing logic.

-> Simulation results are saved in **.csv** files.



Implementation: Rate Calculation

The **rate** is the reciprocal of the the sensing Interval,**T**.

-> The instantaneous rate(r) for every sensing event is calculated based on the **count** of cars sensed to the sensing interval.

general approach of rae calculation

$$r \approx \frac{1}{T} \tag{2}$$

instantaneous rate calculation

$$r = \frac{count}{T} \tag{3}$$

overall rate calculation

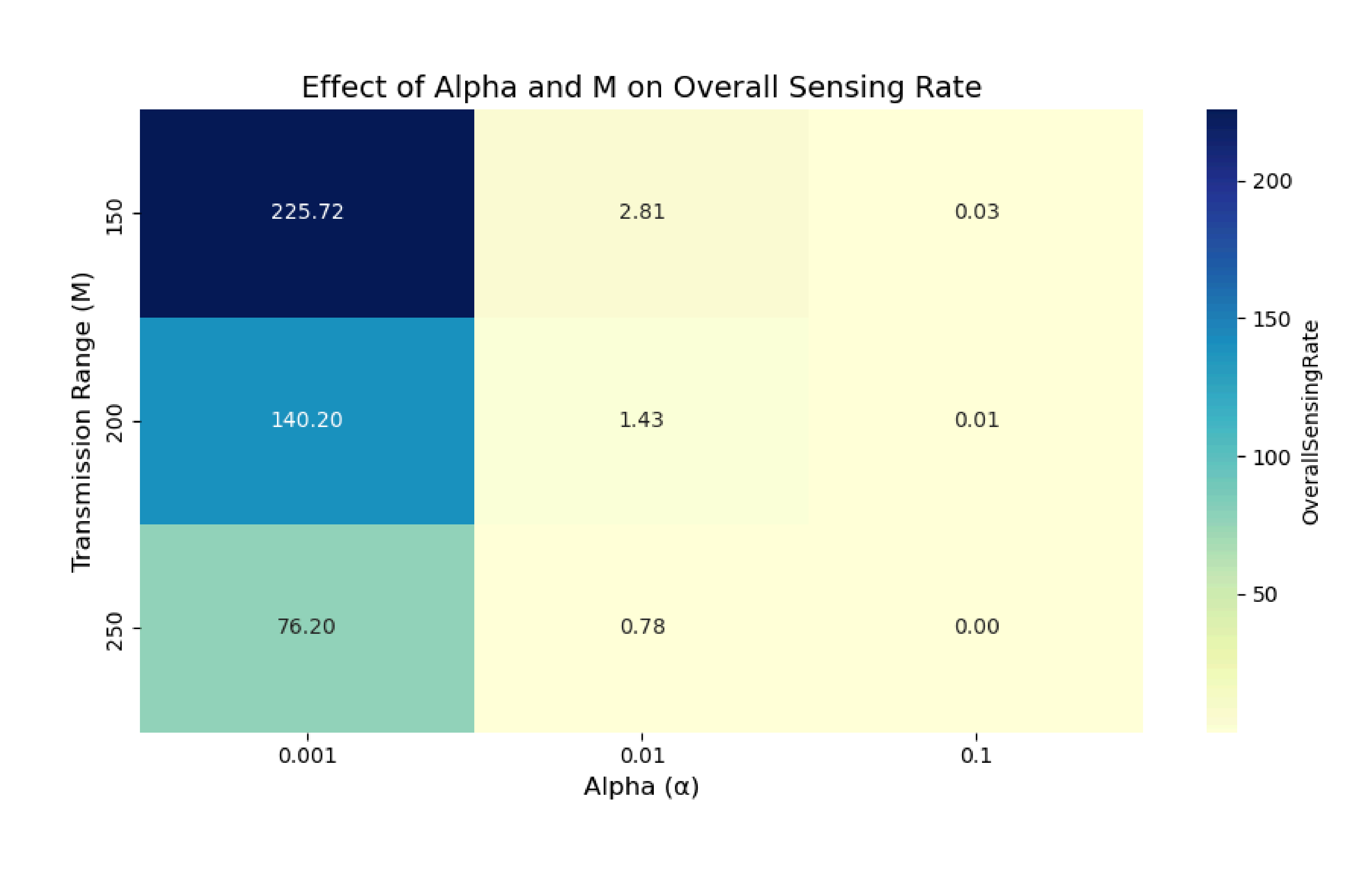
$$R = \frac{1}{N}\sum_{i=1}^{N} r_i \tag{4}$$

-> The **over all sensing rate (R)** is the the sum of the instantaneous rates for each sensing event of all the vehicles and normalized by N.

Results: Impacts of ∝ and M on R

-> The **sensingInteval,T,** determines how frequently the sensinging events are scheduled.

- If T is larger, sensing events occur less frequently, resulting in a lower value of R.
- If T is smaller, sensing events occur more frequently, leading to a higher value of R.
- It is recommended to choose parameter combinations of M and **α** that yield moderate results, ensuring that R remains neither too low nor too high.



Verification: Consistency test

Test:- 1

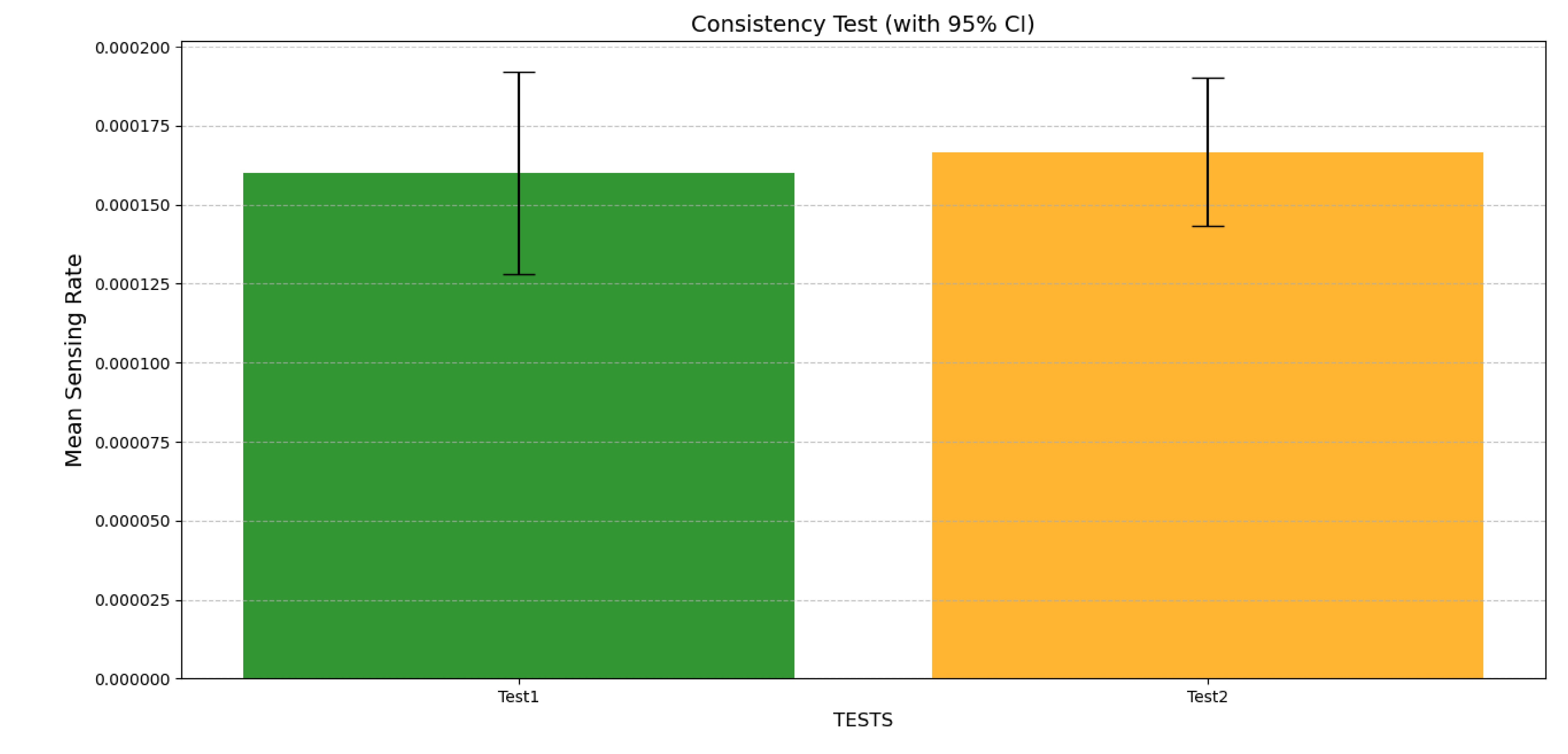
sim-time-limit = 6000s

*.car[*].customSensing.sensingInterval = 100s

Test:-2:

sim-time-limit = 12000s

*.car[*].customSensing.sensingInterval = 200s



Verification: Continuity test

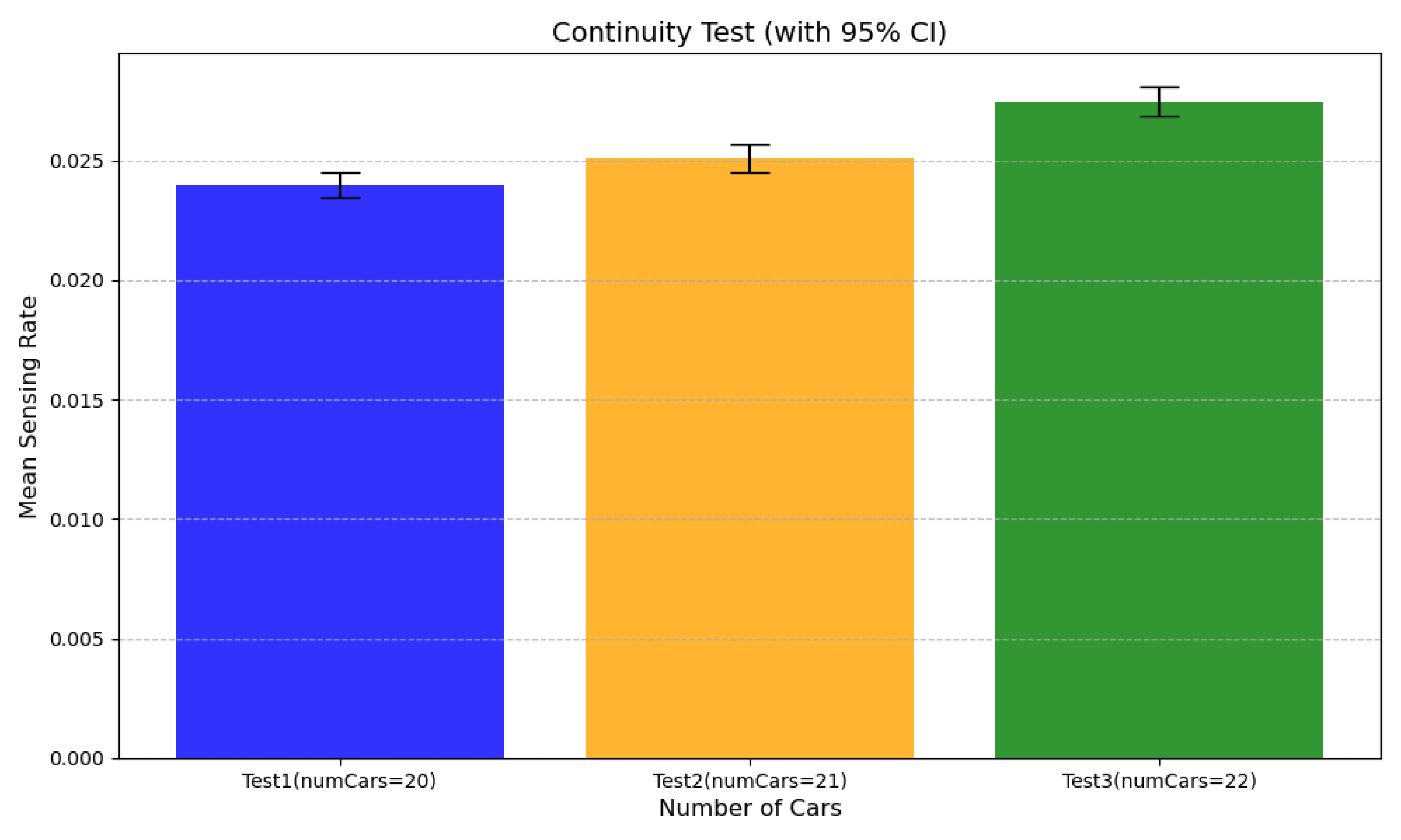
This test is all about to verify if changing slightly the input affects slightly the output.

Test1:- N = 20

Test2:- N = 21

Test3 :- N = 22

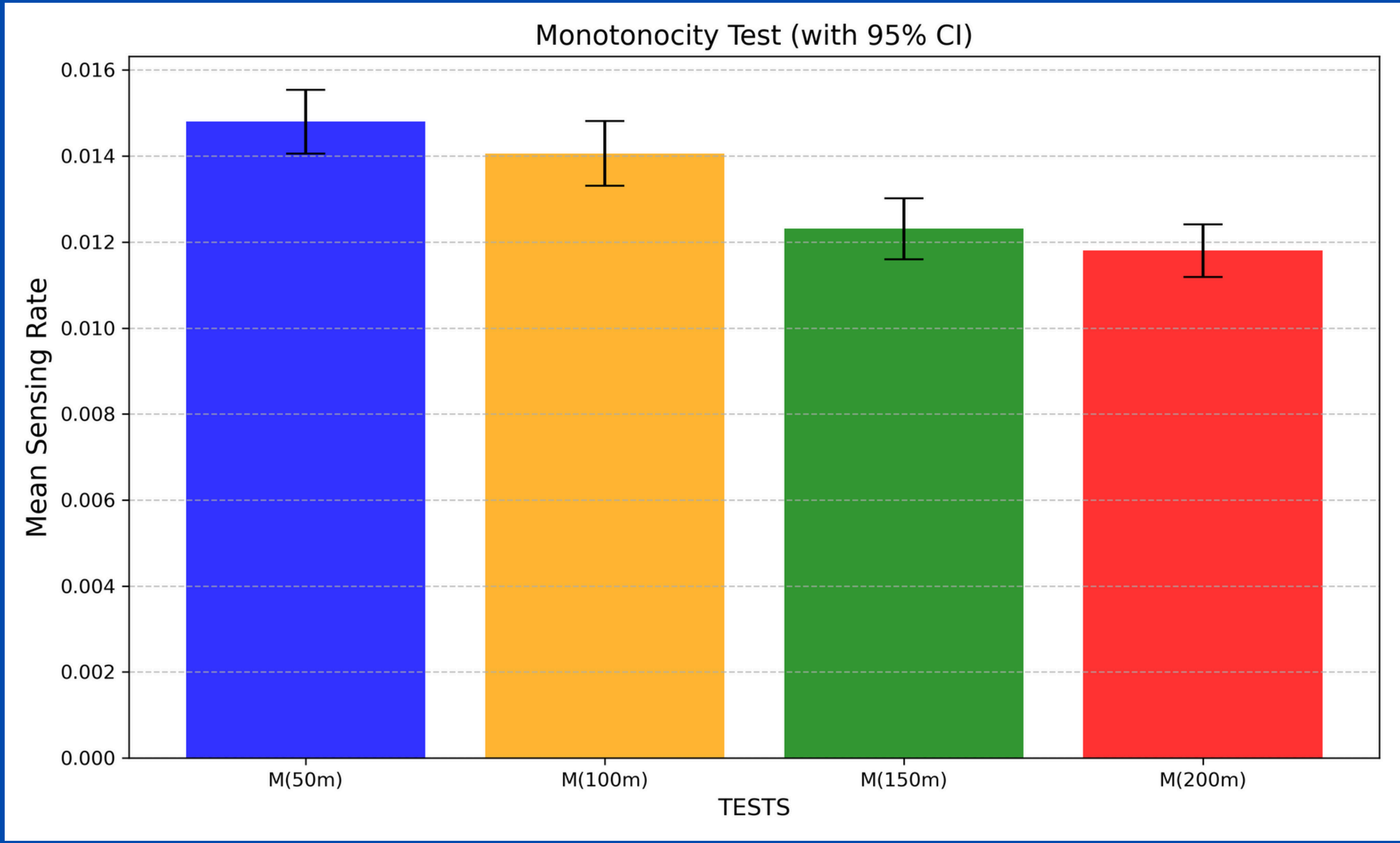
As the number of vehicles (**input**) increases slightly the mean sensing rate (**output**) also increases. Moreover, there are no outliers from the results, so the model passes this test.



Verification: Monotonicity test

This test consists in assessing the monotonicity of some performance indexes using different combinations of factors.

In this test we are going to analyze if the overall sensing rate decrease as we increase **M** in 4 different tests.



Verification: Degeneracy test

This test is used to analyze the behavior of the system when the parameters are set to 0 or to other extreme values.

Test1:- set the N = 1 i.e there is only one car in the simulation. Based on the logic of the sensing module, there is no self-sensing so all the results that we got forthe overall sensing rate is zero in every 'repetition' as expected.

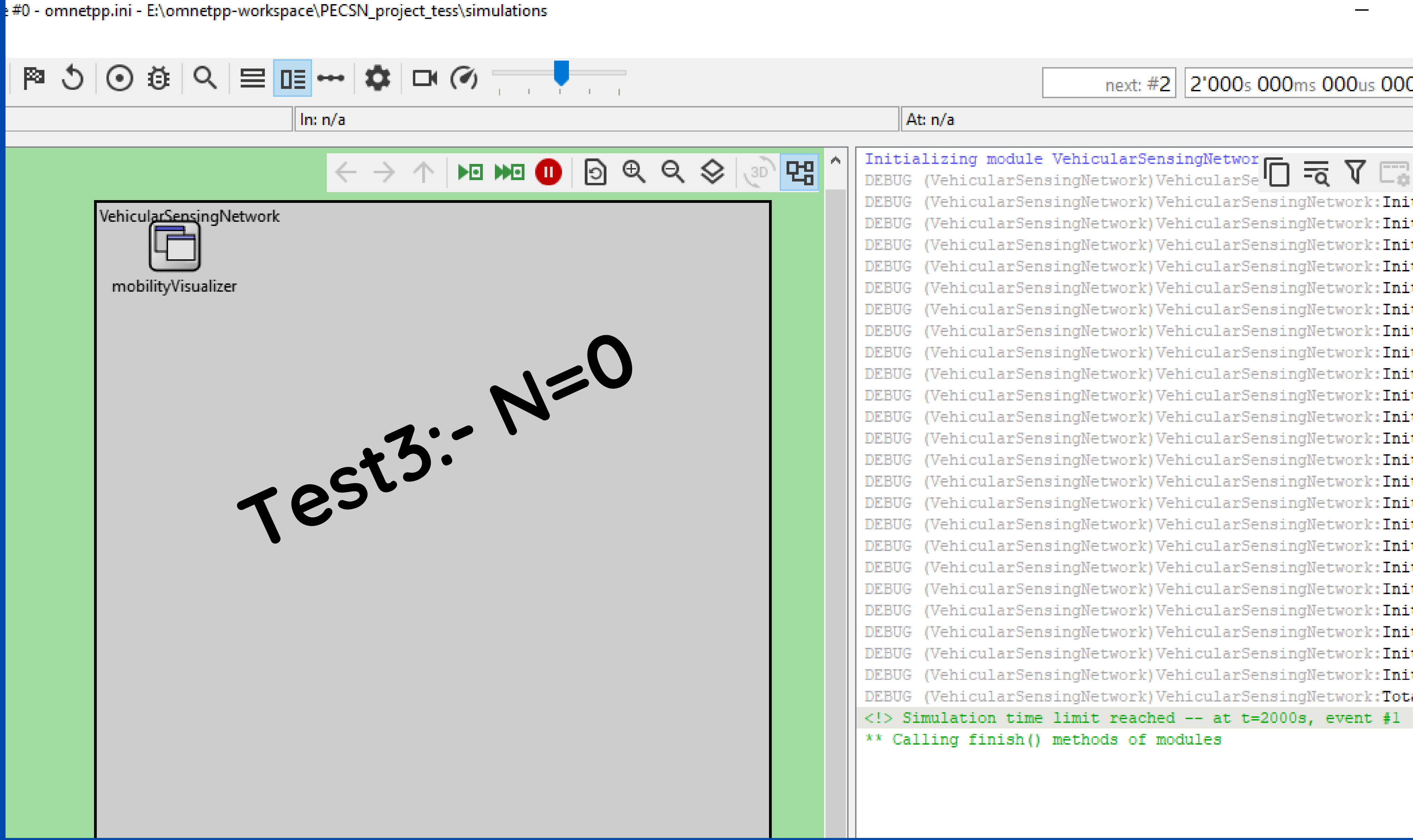
Test2: In this test, the simulation area is reduced from 500m × 500m to just 1m × 1m, effectively making it a single point play-ground. As a result, every vehicle is expected to detect all other vehicles within the simulation environment.

Test3:- set N=0, the system works as expected.

Test1:- N=1

Simulatio	runNumb	CarID	count	rate
225	1	0	0	0
450	1	0	0	0
675	1	0	0	0
900	1	0	0	0
1125	1	0	0	0
1350	1	0	0	0
1575	1	0	0	0
1800	1	0	0	0
2025	1	0	0	0
2250	1	0	0	0
2475	1	0	0	0
2700	1	0	0	0
2925	1	0	0	0
3150	1	0	0	0
3375	1	0	0	0
3600	1	0	0	0
3825	1	0	0	0
4050	1	0	0	0
4275	1	0	0	0
4500	1	0	0	0

CarID	count	rate
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0



Test3:- N=0

```
INFO:Car[0] sensing car[1] at posit
INFO:Car[0] sensing car[2] at posit
INFO:Car[0] sensing car[3] at posit
INFO:Car[0] sensing car[4] at posit
INFO:Car[0] sensing car[5] at posit
INFO:Car[0] sensing car[6] at posit
INFO:Car[0] sensing car[7] at posit
INFO:Car[0] sensing car[8] at posit
INFO:Car[0] sensing car[9] at posit
INFO:runNumber is :0
INFO:alpha value :0.001
```

SimulationTime	runNumb	CarID	count	rate
225	0	0	9	0.04
450	0	0	9	0.04
675	0	0	9	0.04
900	0	0	9	0.04
1125	0	0	9	0.04
1350	0	0	9	0.04
1575	0	0	9	0.04
1800	0	0	9	0.04
2025	0	0	9	0.04
2250	0	0	9	0.04

450	0	1	9	0.04
675	0	1	9	0.04
900	0	1	9	0.04
1125	0	1	9	0.04
1350	0	1	9	0.04
1575	0	1	9	0.04
1800	0	1	9	0.04
2025	0	1	9	0.04
2250	0	1	9	0.04
2475	0	1	9	0.04
2700	0	1	9	0.04
2925	0	1	9	0.04
3150	0	1	9	0.04

Test2:- 1mx1m play-ground

Thank you