Milestone 5: Simulation Runs and Data Collection

Fatih Kurt

April 28, 2025

Contents

1 Introduction

Project Overview:

This simulation models a traffic system in a grid-based urban environment using agent-based modeling. Each car is an autonomous agent making localized decisions about acceleration, deceleration, and stopping at traffic lights based on nearby conditions. The project investigates how varying the acceleration behavior of car agents impacts traffic smoothness, congestion, and overall flow efficiency.

Purpose of Simulation Runs:

These simulation runs are critical for analyzing how different car agent reaction settings (acceleration values) affect traffic outcomes. By systematically varying acceleration, we expect to observe changes in key traffic metrics such as average speed, average wait time, and number of stopped vehicles. The results will directly inform the project hypothesis regarding the relationship between agent smoothness and system-level traffic flow.

Key Questions Addressed:

- What happens to overall traffic smoothness when individual car agents accelerate more aggressively versus more gradually?
- How does car agent behavior contribute to traffic congestion or efficiency?
- How does this milestone provide initial validation and inform future refinements of the simulation model?

2 Simulation Outline

2.1 Simulation Description

The simulation models a city traffic grid using NetLogo.

Key Components

- Car agents that accelerate, decelerate, or stop based on local traffic conditions and traffic lights.
- Traffic light agents that automatically switch based on a fixed cycle.
- Road grid where car agents navigate between randomly assigned houses and work locations.

Real-world Behaviors

- Stop-and-go movement at traffic lights.
- Formation of congestion due to localized slowdowns.
- Acceleration smoothing and its impact on overall traffic efficiency.

Unique Feature:

- Ability to modify the car agents' acceleration parameter easily to simulate different driver reaction behaviors.

2.2 Execution Instructions

To run the NetLogo code on your Web Browser, please do the following:
1. Download the NetLogo file in my GitHub directory (namely my source code in the src branch) to your computer.
2. Go to the NetLogo's website through the following link (https://ccl.northwestern.edu/netlogo/).
3. Click the Go to NetLogo Web to go to NetLogo Web.
4. On the top right corner of the screen, there is an option to upload a model. Click where it says "Choose File" and find and open my NetLogo code.
5. Now you will be able to see variable adjustment section and the simulation display side by side as well as couple of useful graphs at the bottom.

6. Adjust the desired variables. When you are done with your setup, click "Setup" to save your changes to the current iteration of the simulation.

7. Click "Go" to run the simulation.

Required Inputs, Configurations, Parameters

To produce similar results as me, set the following variables as instructed: ticks=500, grid-size-x = 3, grid-size-y = 3, num-cars = 50, power? = true, ticks-per-cycle = 20, speed-limit = 1, current-auto? = true, current-phase = 0

Afterwards, adjust the acceleration variable as desired (this action can't been done through the variable interaction UI right next to the simulation screen. Instead, please use the NetLogo Code section at the bottom of the screen to search for "set acceleration" and set the acceleration through the value next to this code in line 102)(for my tests, I have set the acceleration to 0.049, 0.099, 0.149, and 0.199 (I was advised on not making acceleration 0.1 since we could get a rounding error and end up on a patch boundary).

Output formats

The data is written to the graphs in the middle of the screen. I ran my tests to 500 ticks to get a good, healthy amount of data. Afterwards, I hit the 3 vertical lines on the top right of all graph and chose the Download CSV option to extract the data into my records.

Other

No external libraries are required beyond NetLogo.

Example simulation run

2.3 Assumptions and Limitations

Assumptions:

- All cars have the same behavioral parameters (homogeneous agents).
- Traffic lights operate on fixed timing cycles.
- Cars always move toward goals without lane-changing.

Limitations:

- No explicit modeling of road types (all roads assumed identical).
- No realistic car-to-car communication.
- Pedestrian behavior, weather, and accidents not modeled.

Trade-offs:

- Simple acceleration modeling allows focus on isolated agent behavior without external distractions.

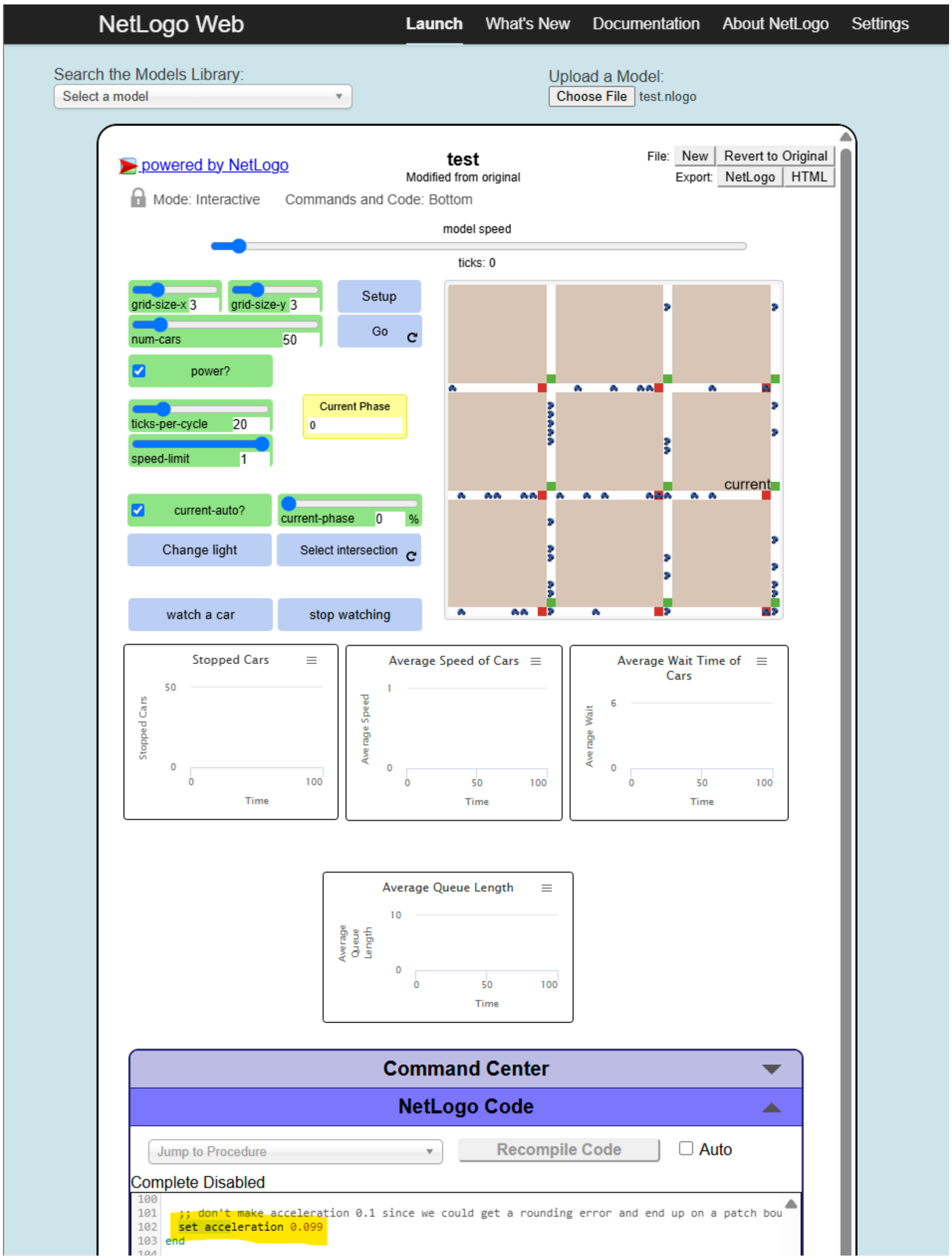

Figure 1: Example: Set Acceleration = 0.099

3 Simulation Run Summary

Overview of Runs:

For this stage of the project, I conducted initial simulation runs to explore how varying the car agents' acceleration rates affects traffic smoothness and congestion. The primary parameter varied was the acceleration value, while all other simulation settings remained constant.

Each simulation run was conducted under the following controlled conditions: ticks=500, grid-size-x = 3, grid-size-y = 3, num-cars = 50, power? = true, ticks-per-cycle = 20, speed-limit = 1, current-auto? = true, current-phase = 0

With different test runs, different acceleration values were tested. To be more specific,

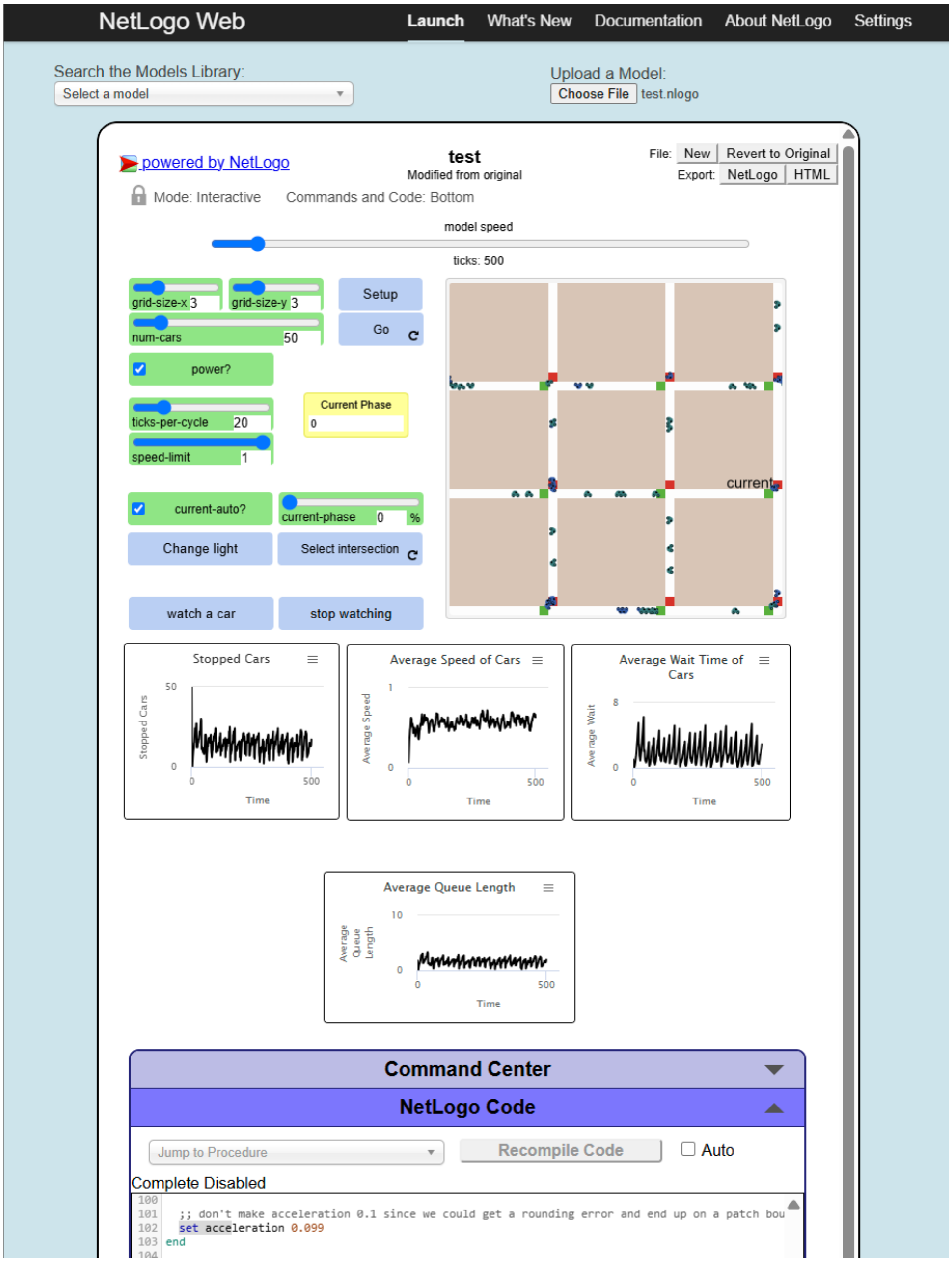

Figure 2: Example: After 500 ticks

I have tested my model with acceleration set to the following values: 0.049 (gradual acceleration), 0.099 (default acceleration), 0.149 (semi-aggressive acceleration), and 0.199 (aggressive acceleration).

For each run, the key data I have collected was the following: the average speed of cars over time, the average wait time of cars at red lights or in traffic, and number of stopped cars per tick.

The purpose was to analyze how smoother or more aggressive agent behaviors impacted the overall flow of traffic.

Simulation Run Table:

Key Insights

- Summary of Simulation Performance and Outcomes
  - Across the four simulation runs, varying the acceleration behavior of car agents produced clear and meaningful differences in traffic dynamics. Default acceleration provided balanced and moderate traffic flow, while smoother acceleration, moderate aggressiveness, and high aggressiveness each impacted traffic congestion, average speed, queue lengths, and wait times differently.
- Trends, Patterns, and Significant Results
  - Default acceleration achieved the most balanced traffic flow, with moderate queue lengths, wait times, and minimal stoppages.
  - Smoother acceleration slightly increased queue lengths and wait times, leading to slightly more congestion than anticipated.
  - Moderate acceleration (0.149) showed little improvement over the default, and actually slightly worsened wait times and stoppages.
  - Very aggressive acceleration (0.199) produced the worst traffic outcomes, including the highest queue lengths, stoppages, and wait times, despite an increase in average speed.
  - A key trend observed is that excessive aggressiveness in acceleration worsens traffic flow, rather than improving it.
- Expected Outcomes vs. Reality
  - While the overall direction of effects was generally aligned with expectations (e.g., smoother driving causing smoother but slower traffic), the extreme negative impacts of very aggressive acceleration were more severe than originally expected. Some assumptions about faster acceleration leading to faster system performance were disproven.

4 Data Collection and Storage

Data Collection Methodology

During each simulation run, several key traffic metrics were recorded to monitor the performance and behavior of the system under different acceleration settings:

- Data Points Recorded:
  - Number of stopped cars
  - Average speed of cars
  - Average wait time of cars
  - Average queue length at intersections
- Frequency of Collection:

  - Data was collected at every simulation tick (i.e., once per discrete time step).
  - This provided a fine-grained, continuous view of how traffic patterns evolved over time.

- Methods of Data Capture:
  - NetLogo's built-in plotting tools were used to log simulation variables in real time.
  - After each run, the plot data was exported directly from the NetLogo interface into separate data files.
  - No manual recording was involved; all data was captured automatically by the simulation environment.

- Preprocessing or Filtering:
  - Minimal preprocessing was applied.
  - Raw values were directly exported without smoothing or filtering to maintain the integrity of the original simulation outputs.
  - Minor cleaning (such as removal of placeholder or redundant columns) was performed after export to focus analysis only on relevant variables.

Data Formats and Storage

- File Formats:
  - All collected data was exported as CSV (Comma-Separated Values) files.
  - This format was chosen because it is simple, widely supported, and ideal for later analysis in tools like Python, Excel, or statistical software.

- Organization and Naming Conventions:
  - Files were named systematically to clearly indicate the corresponding test and the type of data they contain. (Example filenames: test 1 average-speed-of-cars.csv, test 2 stopped-cars.csv, etc.)
  - Files for each run were grouped together logically to ensure easy retrieval and tracking.

- Data Access for Analysis:
  - The CSV files were imported into Python using Pandas for cleaning, summarizing, and visualizing the data.
  - All analysis scripts were designed to automatically read the structured CSV files, minimizing manual intervention and error.

- Data Integrity and Reliability:
  - Data exports were verified immediately after each simulation run to ensure no corruption or missing values.
  - File backups were maintained throughout the milestone to protect against data loss.

Additional Questions Addressed

- Alignment with Simulation Goals
  - The collected data directly measures core traffic dynamics such as congestion (stopped cars, queue length), efficiency (average speed), and system responsiveness (wait time), perfectly aligning with the goal of understanding how agent acceleration behavior impacts overall traffic performance.
- Limitations or Challenges
  - One limitation encountered was the need to ensure that exported data accurately matched the intended test parameters (e.g., early runs mistakenly saved the wrong test labels). Careful documentation and re-exporting were needed to correct these issues. Additionally, NetLogo's export format required minor post-processing to make datasets ready for structured analysis.
- Structuring Data for Efficient Analysis
  - By using uniform CSV structures and consistent naming conventions, the collected data could be easily ingested into Python for automated summarization and plotting. This allowed efficient statistical comparisons between different simulation runs and rapid generation of analytical charts.

5 Data Analysis

5.1 Analysis Approach

The collected traffic simulation data was analyzed systematically using descriptive statistical techniques:

- Statistical Methods Applied:
  - Mean: To determine the average behavior of each traffic metric over the course of each simulation.
  - Minimum and Maximum Values: To identify the range of variability for each metric.
  - Standard Deviation: To quantify the consistency or variability of traffic flow for each test run.
- Tools and Software:
  - Python was used for data processing and visualization, specifically using the libraries Pandas (for data summarization) and Matplotlib (for plotting charts).
  - Additional basic verification and organization were performed using Excel for quick visual cross-checks.
- Key Metrics (KPIs) Evaluated:
  - Average Queue Length: Represents congestion buildup at intersections.

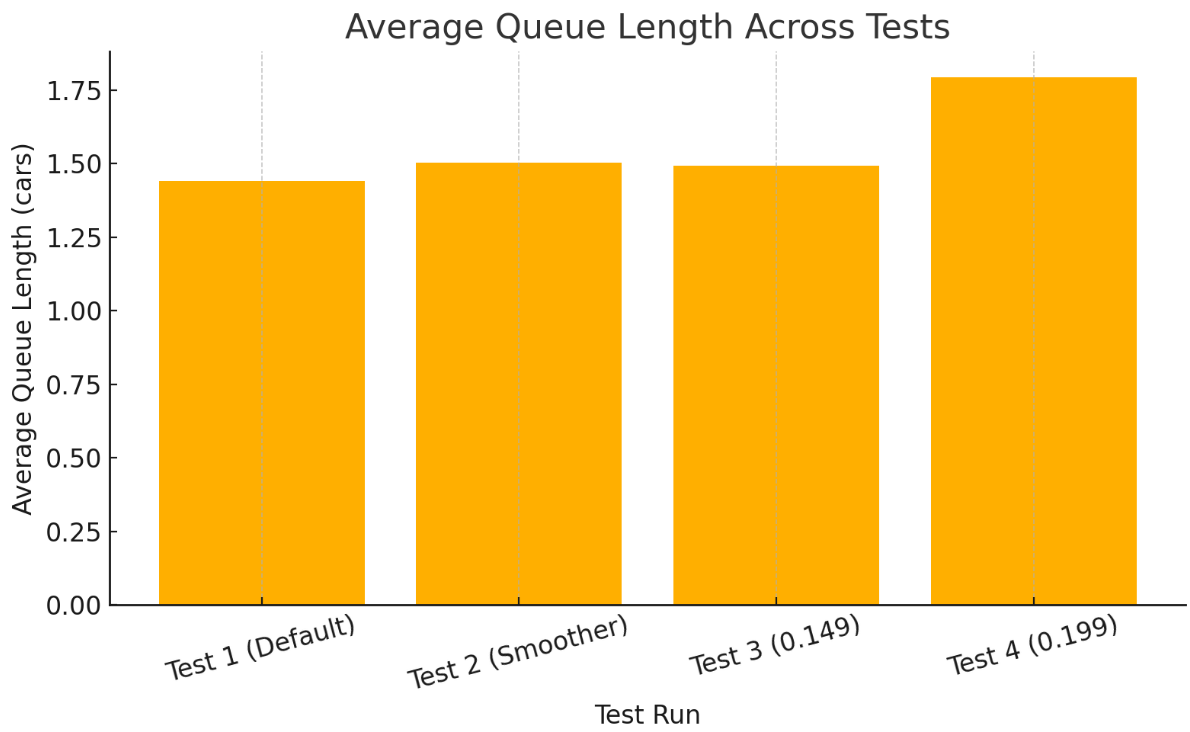

Figure 3: Average Queue Length Across Test Bar Graph

  - Average Speed: Indicates overall traffic flow and efficiency.
  - Average Wait Time: Measures how long cars were delayed.
  - Number of Stopped Cars: Tracks the level of traffic interruptions at each moment.

- Preprocessing Steps:
  - Removal of redundant placeholder rows from CSV exports.
  - Alignment of time axes across datasets to ensure direct comparability.
  - Re-verification of run labels to ensure data matched corresponding test conditions (after correcting early mislabeling issues).

5.2 Results

Key findings were visualized through bar graphs to clearly show the effects of different acceleration settings on traffic dynamics:

Visualizations:

5.3 Discussion

- Key Insights from Data:
  - Default acceleration delivered the best balance between low queue lengths, moderate speeds, and minimal stoppages.

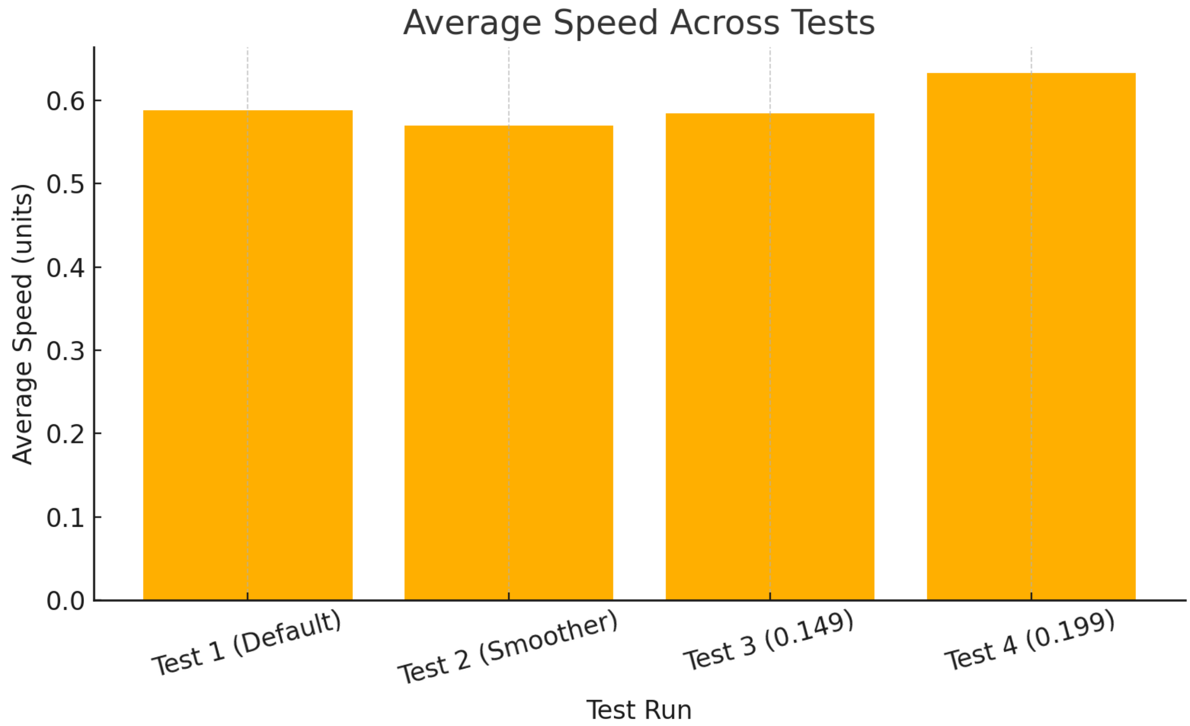

Figure 4: Average Speed Across Test Bar Graph

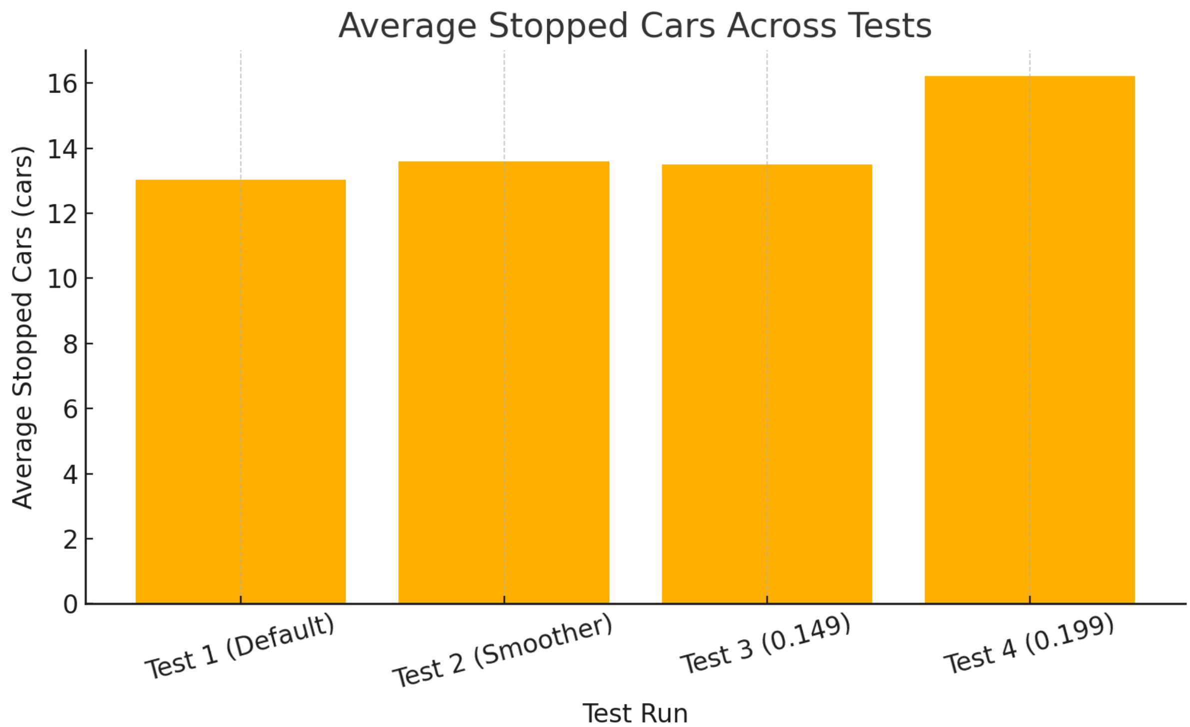

Figure 5: Average Stopped Cars Across Test Bar Graph

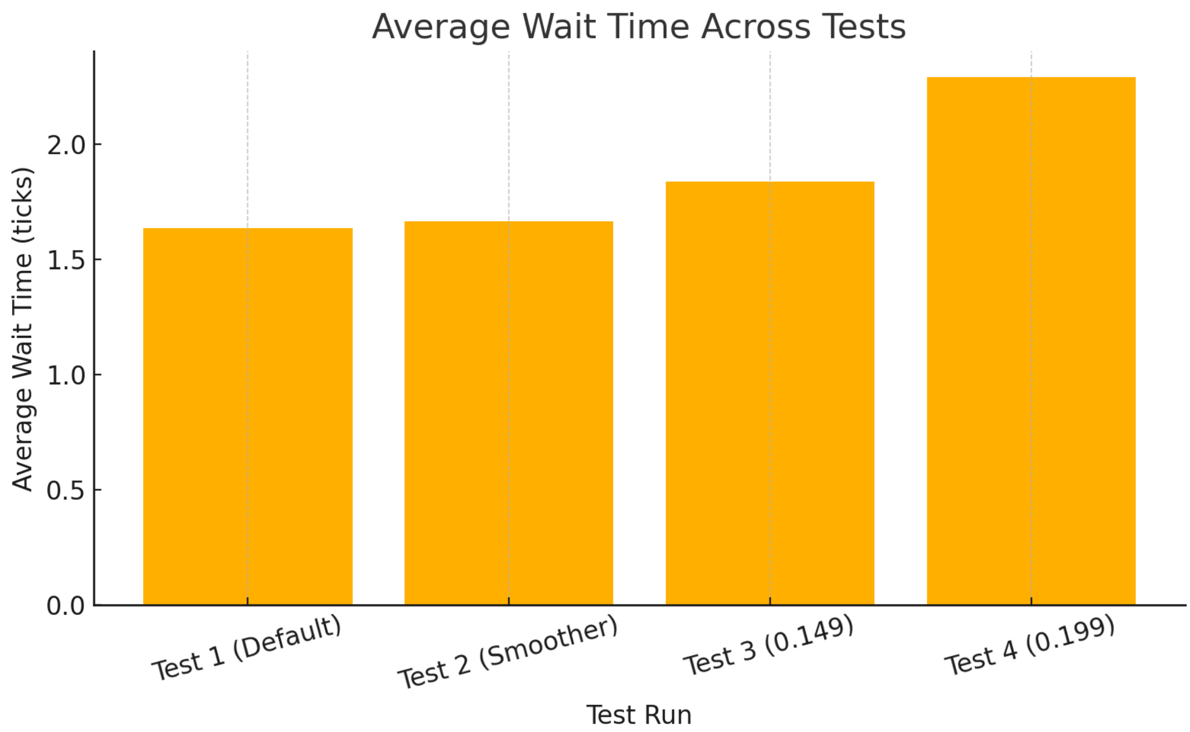

Figure 6: Average Wait Time Across Test Bar Graph

- Smoother acceleration slowed traffic slightly and increased congestion marginally, as expected.
- Moderate acceleration (0.149) did not improve traffic efficiency meaningfully and slightly worsened congestion.
- Very aggressive acceleration (0.199) significantly increased average speed but at the cost of higher congestion, longer waits, and more stoppages, which overall worsened traffic efficiency.

- Unexpected Findings:
  - It was unexpected that very aggressive acceleration would increase congestion and waiting times so significantly.
  - Rather than speeding up the system overall, it created chaotic traffic flows, where agents frequently stopped and blocked intersections.

- Contribution to Project Goals:
  - These results strongly support the project's aim of understanding how individual driver behaviors (agent acceleration settings) aggregate into emergent traffic patterns.
  - The simulation revealed that small changes in agent-level acceleration policies can create major systemic effects on traffic congestion and flow quality.

- Validation of Simulation Model:
  - The observed results, including nonlinear relationships between aggressiveness and traffic efficiency, are consistent with real-world traffic dynamics.

  - This supports the validity of the simulation model and confirms that agent-based modeling is a powerful tool for studying urban transportation systems.

6 Conclusion

Summary of Findings

The simulation runs clearly demonstrated that agent-level acceleration behavior significantly impacts traffic dynamics in an agent-based traffic model. Key findings include:

- Default acceleration produced the most balanced outcomes, maintaining moderate queue lengths, speeds, and wait times.
- Smoother acceleration slowed average speed slightly and led to minor increases in queue lengths and stoppages, somewhat counter to the expectation of smoother flow.
- Moderate aggressive acceleration (0.149) did not significantly improve traffic conditions and slightly worsened wait times and congestion.
- Very aggressive acceleration (0.199) resulted in the worst traffic outcomes, with higher average queue lengths, wait times, and numbers of stopped cars despite slightly higher speeds.

A major takeaway is that increasing driver aggressiveness worsens traffic congestion, even if it improves local speed temporarily.

Unexpectedly, extreme acceleration behaviors led to chaotic congestion patterns, challenging the assumption that faster individual behaviors would yield better system performance.

Implications for the Project

These results are critical in validating the purpose and goals of the project, which aim to understand how small-scale agent decisions influence large-scale traffic system behavior.

- The findings reinforce that agent behavior must be carefully modeled when designing or optimizing traffic flow systems.
- The model successfully captured nonlinear system dynamics, supporting the use of agent-based modeling for transportation studies.
- The results suggest that policies promoting moderate, controlled driving behaviors may be more effective in minimizing congestion than encouraging faster movement.

Additionally, the clear worsening of performance under high-aggressiveness settings suggests that future refinements to the simulation could focus on modeling collision avoidance, adaptive route selection, or penalties for erratic movement to further improve realism.

Next Steps

Building on this milestone, the next phases of the project will include:

- Exploring additional scenarios, such as mixed-agent populations with varying acceleration behaviors rather than uniform agent settings.
- Collecting additional data, including metrics like fuel efficiency, number of collisions (if implemented), and trip completion times.
- Extending the simulation model to support more complex traffic control features, such as dynamic traffic lights or priority lanes.
- Running larger grid simulations or varying grid densities to study the scalability of agent behavior effects on bigger transportation networks.
- Refining analysis techniques by applying deeper statistical tests (e.g., hypothesis testing) to formally validate observed trends.

These steps will strengthen the project's findings and provide a richer understanding of agent-based traffic modeling dynamics.

7 References

In the GitHub repository.

Run ID	Goal	Config	Expected Outcome	Actual Outcome	Notes
1	Study baseline agent behavior (default acceleration)	Acceleration = 0.099	Moderate traffic flow expected; acts as baseline for comparison.	As expected, default acceleration resulted in moderate traffic flow with relatively low queue lengths, wait times, and stoppages, serving as a solid baseline.	Default system behavior without acceleration tweaks.
2	Study smooth acceleration impact on traffic flow	Acceleration = 0.049	Gradual acceleration expected to reduce sudden stops and improve traffic smoothness.	Smoother acceleration slightly increased average queue length, wait time, and number of stopped cars compared to expectations, indicating smoother starts created slight congestion rather than reducing it.	Notes.
3	Study semi-aggressive acceleration behavior	Acceleration = 0.149	Semi-aggressive acceleration likely to cause more sudden stops and localized congestion.	Moderate aggressiveness behaved similarly to default settings, with slightly higher wait times and congestion than expected, showing minimal performance improvement despite higher acceleration.	Notes.
4	Study aggressive acceleration behavior	Acceleration = 0.199	Aggressive acceleration likely to cause the most sudden stops and localized congestion.	Very aggressive acceleration worsened traffic metrics significantly, increasing queue lengths, stoppages, and wait times rather than improving overall traffic	Notes.

Metric	Test 1	Test 2	Test 3	Test 4
Average Queue Length (cars)	1.44	1.50	1.49	1.79
Average Speed (units)	0.588	0.569	0.584	0.632
Average Wait Time (ticks)	1.64	1.67	1.84	2.29
Average Stopped Cars (cars)	13.03	13.60	13.50	16.21

Table 2: Results