Kai Matheson
May 11, 2018

Modeling, Simulation and Analysis
CS 250 - Spring 2018
Final Project: Angry Drivers

Introduction

Many individuals spend copious amounts of time doing highway driving. Certain cities are known for having worse congestion problems compared with others. People shape their lives around traffic conditions, whether it is choosing what time to go to work, whether to carpool, or even in which metropolitan area to reside. This means that the efficient design of highways to serve their purpose is crucial surrounding major metropolitan areas. Traffic simulations can provide insights on how best to design a road, given the level of expected congestion. Hypothesis testing of traffic simulation can provide key answers to questions of highway design. It can also provide insights about whether the rules by which people drive are optimal. In this work, I begin to explore the realm of possibilities by delving into particular questions related to cars' relationship with the road and one another, specifically by implementing an agent-based model that considers how drivers may respond to increasing frustration levels and honking.

Background

There are many models in the field of traffic simulation, but what I would like to emphasize is the difference between microscopic and macroscopic models. Macroscopic models deal with aggregated transportation behavior. In past work, I explored the impact of bus transportation on the spread of disease. This was a macroscopic model of transportation infrastructure, because it utilized a homogenous mixing scheme in which people were assumed to spend proportional amounts of time in each place. Macroscopic models rinse out some of the finer details to report on average dynamics of a system. Microscopic models, on the other hand, pay finer attention to how individual vehicles move at any given moment. Some microscopic models are built using cellular automata, while others use agent-based simulation methods. Car-following models are microscopic models based on differential equations. In my own car-following model of angry drivers, I take elements found in one particular car-following model called Microscopic Traffic Simulator (MITSIM) in order to calculate acceleration when not affected by frustration levels of the driver. I implement mechanisms for the cars to change lanes, accelerate and decelerate, and honk, all in connection with the frustration level of the driver at any given moment.

Model Description

A car is defined as a MATLAB struct which has the following fields:

index: one integer index of a car
desiredSpeed: one positive number, speed at which the vehicle would travel in the absence of other cars
frustration: array of positive numbers, representing frustration level of vehicle at times corresponding to the time array
acceleration: one number, current acceleration rate of vehicle
position: array of positive numbers corresponding to the time array
speed: array of numbers corresponding to the time array
time: array of time steps within which the car is on the road
lane: array of integers representing which lane the car is in corresponding to the time array
honk: binary array representing honking status corresponding to the time array

Cars are placed on the road when there is less than the desired number of cars currently on the road, and there the beginning of every lane is clear by *minFollowingDistance* kilometers. When a car is placed at the beginning of the road, it is initialized to be in a random lane, with a desired speed uniformly distributed between 30 km/s and 50 km/s, a frustration level uniformly distributed between 1 and 1.5, an initial speed uniformly distributed between 10 km/s and 50 km/s, and a lane also uniformly distributed between the number of lanes.

The *currentPositions* matrix holds crucial information about the current state of the world in each time step: it tracks every car's current position, speed, acceleration, and lane in a given moment. In each time step, each car's speed is calculated using Euler's method and the car's acceleration from the previous time step. Then, each car's position is calculated using the speed that has just been calculated. We then check whether the car is going above its desired speed. If it is, then frustration is set equal to base frustration.

The model is built up in the following sequence, and thus will be explained in the same sequence:

1. Floor 1: Cars on a one-lane road, no lane-changing, honking or frustration. Acceleration determined based on following distance and desired speed.
2. Floor 2: Cars on a one-lane road, with honking and frustration implemented. Acceleration determined based on following distance, desired speed, and frustration level.
3. Final model: Cars on a three-lane road, with lane-changing, honking and frustration.

Floor 1

The MITSIM model is a car-following model that uses three types of following behavior: free driving, car-following, and emergency deceleration. It uses time headway as a threshold to transition to different modes of car-following. My model, on the other hand, uses MITSIM's equations for following behavior, but instead of time headway, actual kilometer headway, as a threshold to transition to different modes of car-following.

In free driving, the car wants to go its desired speed. The car is at least *maxFollowingDistance* kilometers away from the leading car, and so it can accelerate as it pleases. Its acceleration is defined as

$$a_n = \begin{cases} a_n^+ & v_n < v_n^{desired} \\ 0 & v_n = v_n^{desired} \\ a_n^- & v_n > v_n^{desired} \end{cases}$$

In car-following, the car is between *maxFollowingDistance* and *minFollowingDistance* kilometers away from the leading car. Acceleration is defined as

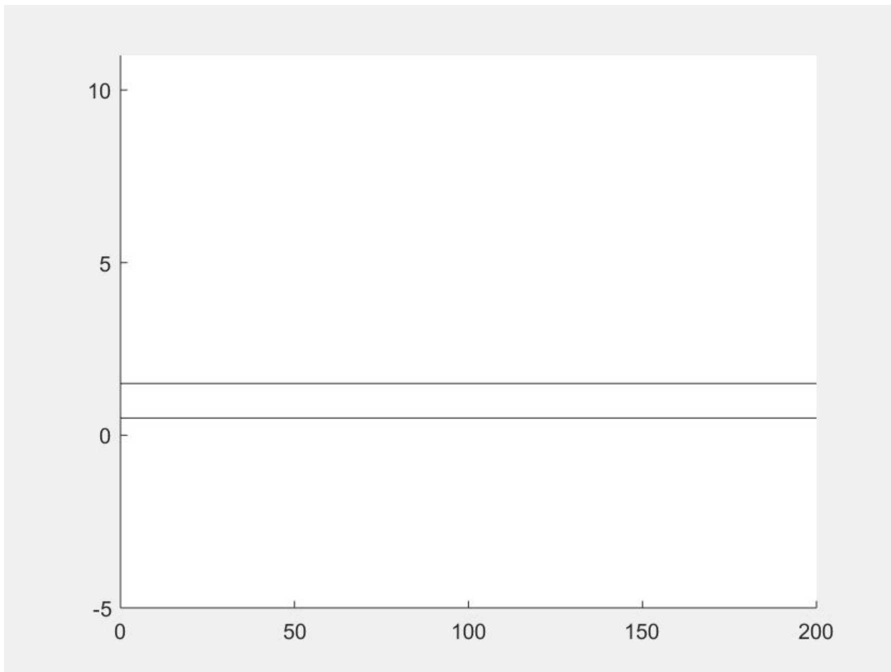
$$a_n = \alpha^{\pm} \frac{v_n^{\beta^{\pm}}}{(x_{n-1} - L_{n-1} - x_n)^{\gamma^{\pm}}} (v_{n-1} - v_n)$$

where α^+, β^+, and γ^+ are used when $v_n \leq v_{n-1}$, and α^-, β^-, and γ^- are used when $v_n > v_{n-1}$.

In emergency deceleration, the car is under *minFollowingDistance* kilometers away from the leading car. Acceleration is defined as

$$a_n = \begin{cases} min(a_n^-, a_{n-1} - 0.5(v_n - v_{n-1})^2/(x_{n-1} - L_{n-1} - x_n)) & v_n > v_{n-1} \\ min(a_n^-, a_{n-1} + 0.25a_n^- & v_n \leq v_{n-1} \end{cases}$$

I use parameter values from the MITSIM model in my own model. These can be found in the work cited by Olstam and Tapani (2004).



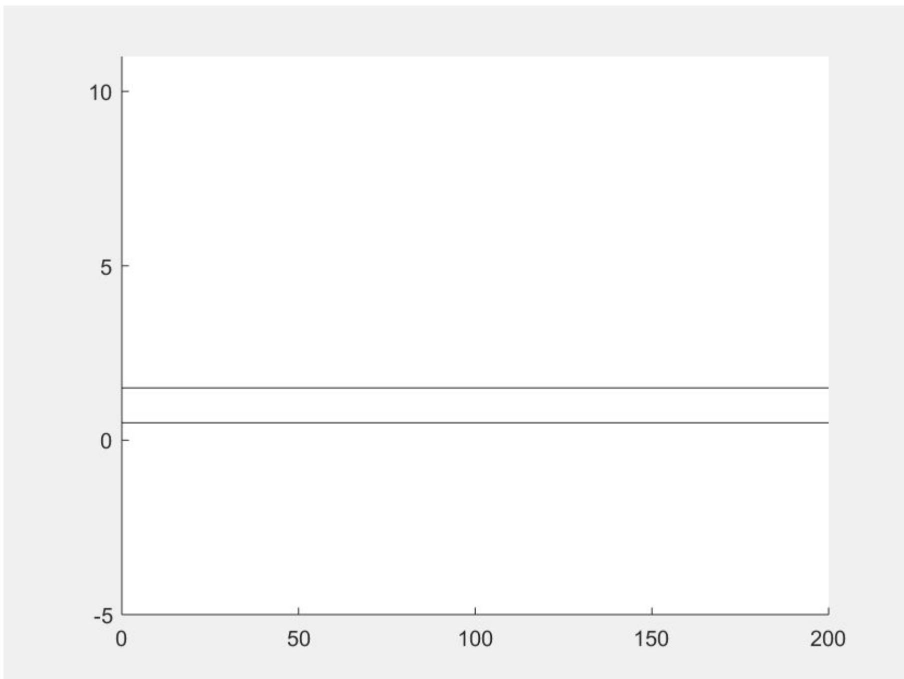
This is an example of one run of the model at Floor 1. One can see how there is a traffic jam accumulating towards the start of the road.

Floor 2

In the transition from Floor 1 to Floor 2, honking and frustration are put into effect in calculating acceleration. While emergency deceleration does not change from Floor 1 to Floor 2, the equations for the other two types of car-following do. Acceleration during free driving becomes $a_n^+ * Frustration$ if $v_n < v_n^{desired}$. Acceleration during car-following becomes $a_n * (Frustration)^{1/2}$ if $v_n < v_n^{desired}$.

Frustration is initially a value in the interval [1, 1.5]. It is then updated based on whether a car is going its desired speed. If it is going at or above its desired speed, then frustration is reset to the base level, which is 1. If it is going below its desired speed, frustration is kept at its current level. If a car experiences another car honking, its frustration goes up incrementally, which, in the final model, is a 0.05 increase.

If a car has reached its frustration threshold, currently set to 1.5, then it honk at the other drivers. When it honks, it releases frustration, so frustration resets to the base level of 1.

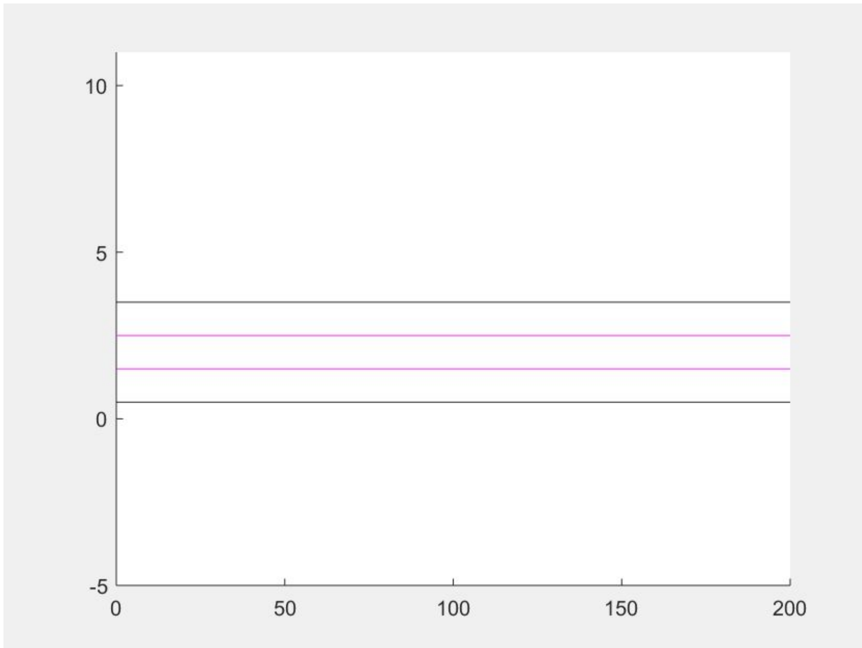


This is an example run of the model at Floor 2. Honking is represented in the visualization as a red dot just before a vehicle.

Floor 3: Final Model

In the transition from Floor 2 to Floor 3, the final model, I expand the road to multiple lanes, in this case, 3 lanes, and implement lane-changing. Now, if a car reaches its frustration threshold, then it will check whether it can change lanes. If it can change lanes, it will, but if it cannot change lanes, then it will honk. Honking functions as it did before. Cars also change lanes randomly with a very small probability, currently set to 1/100. Now, we can see the full dynamics of the system: cars are speeding up and slowing down based on leading cars, their desired speeds, and their frustration levels.

Cars honk at one another and spread frustration. We witness cars passing one another as well as the waves that are characteristic of traffic flow.



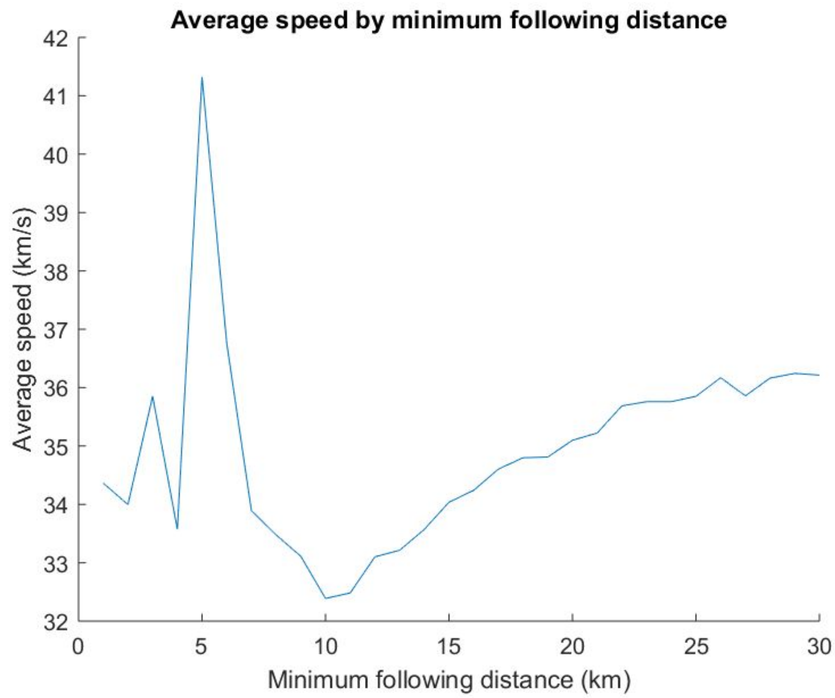
This is an example run of the final model. Honking is represented in the visualization as a red dot just before a vehicle. Changing the number of lanes in the actual simulation is easy, but the visualization code must be edited to include additional lanes.

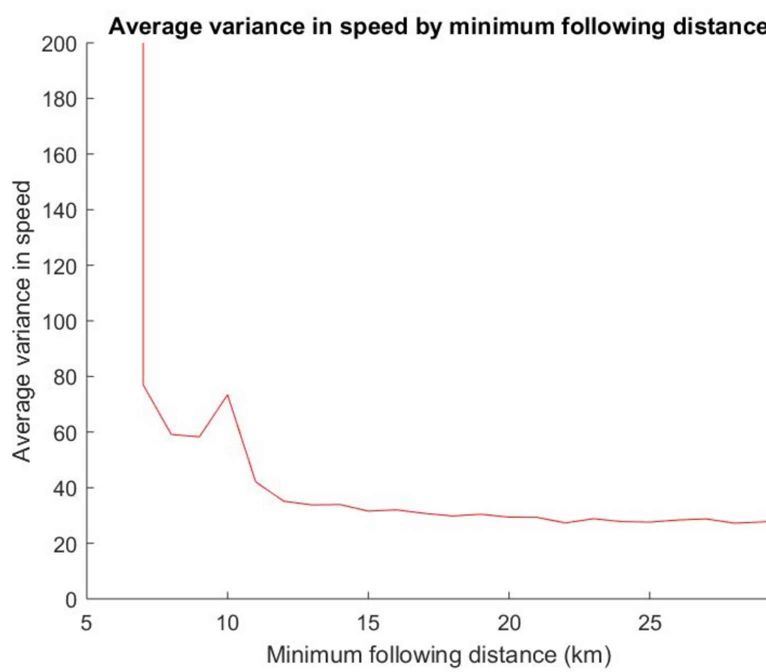
Hypothesis Testing and Results

<u>Hypothesis I:</u> Average speed of cars does not change with changes in *minFollowingDistance*, while average individual variation in speed does.

In my simulation, *minFollowingDistance* is the space between a following car and its leading car at which the following car will transition from car-following behavior to emergency deceleration. *MaxFollowingDistance* is the space between a following car and its leading car at which the following car will transition from free driving to car-following behavior. In particular, recommendations of what constitutes a safe following distance frequently arise. When *minFollowingDistance* is decreased, cars have to decelerate faster in order to avoid an accident. When *minFollowingDistance* is larger, cars may be able to travel at a more constant speed.

I ran the simulation 50 times for each value of *minFollowingDistance*, where *minFollowingDistance* is an integer between 1 km and *maxFollowingDistance*, which in this case was 30 km.





One can see in the results that for values under 10 km, there is a greater amount of variability in average speed by minimum following distance, obvious in both the graph of average speed and the graph of average variance. The relationship between average speed and minimum following distance is not as straightforward as we expected. It seems as though there is a critical point at around *minFollowingDistance* = 10 km, where before that, there is instability in the average speed of vehicles, while after that, there seems to be a more steady relationship that showcases how as minimum following distance increases, average speed actually increases as well. With variance, we can see that at lower values of *minFollowingDistance*, there is much more variability in average speed, while after around 10 km, the average variance in speed stabilizes at around 30 to 40.

These results show that with following distances above 10 km, as following distance increases, average speed increases while variance in speed decreases slightly. With following distances below 10 km, averages do not tell us as much because there is so much variation in speed. Thus, the hypothesis is incorrect: these results show that average speed increases with an increased minimum following distance, while variation in speed decreases with an increased minimum following distance. Lower variation and increased speed is more fuel-efficient, thus these results have important implications in how individuals should aim to drive.

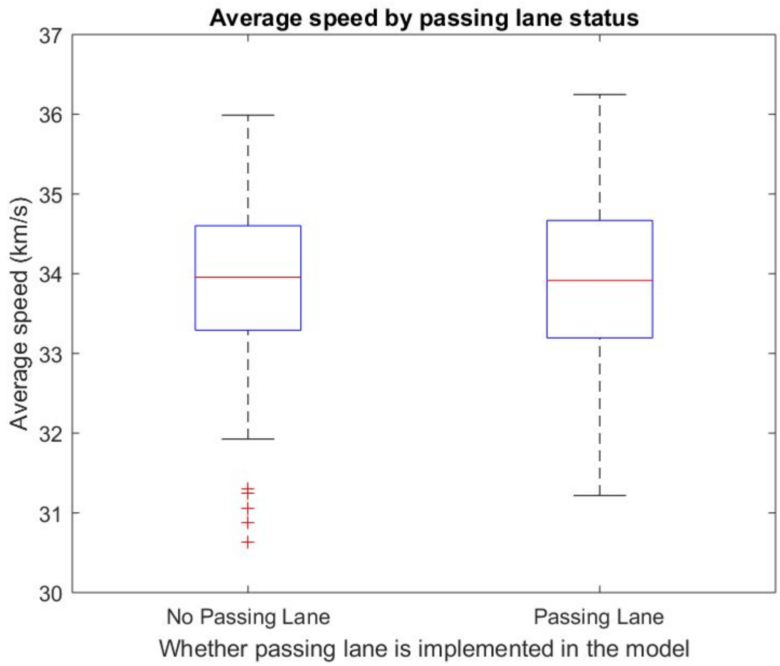
<u>Hypothesis II:</u> All else equal, implementation of a “passing lane” leads to a higher average speed.

Many highways have the left lane be the fast lane, that a car will change lanes to in order to pass a slower leading car. The belief is that this leads to higher average speeds because those who would like to travel quicker are more able to. This lane is also one that is not supposed to be driven in for longer periods of time.

I implement a passing lane by the addition of the following simple rules:

1. The left-most lane is the passing lane.
2. If you are in the passing lane after five time steps, you must change lanes as soon as the lane next to you is clear.
3. If you want to change lanes, you must change lanes to the left unless you are in the left-most lane.

The boxplot below showcases the result of 200 iterations each of running the simulation with and without a passing lane.

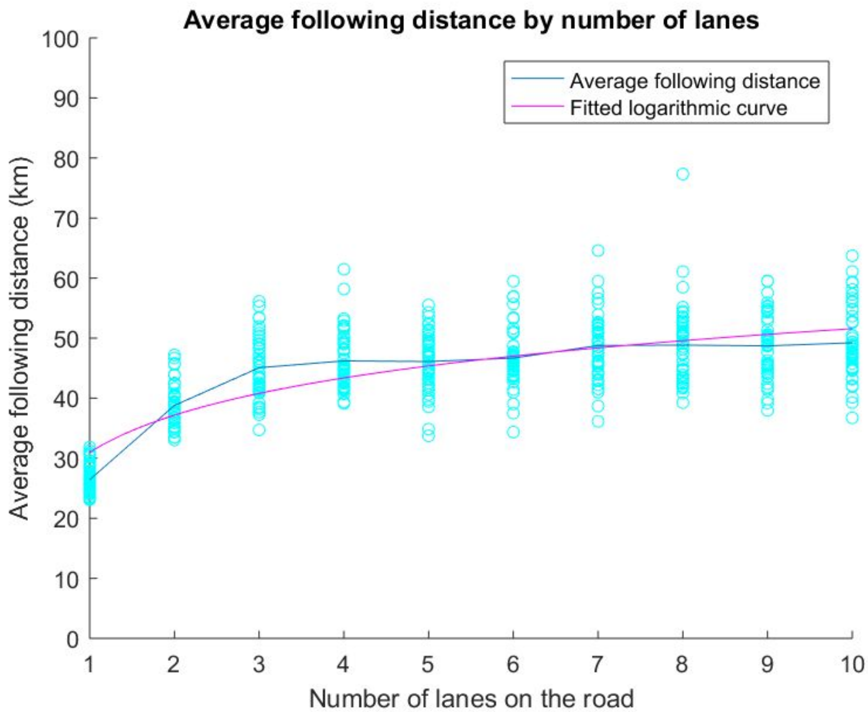


We can see that average speed is roughly the same for both of them. Average speed of vehicles without a passing lane is 34.0348 km/s, while average speed of vehicles with a passing lane is 33.8903 km/s. This is not a considerable difference, however, as one can see in the boxplots. The distribution of average speeds very much overlap, however the distribution of average speeds in the absence of a passing lane does not vary as greatly as the distribution of average speeds in the presence of a passing lane. Thus, we can reject the hypothesis that passing lanes lead to higher average speeds, although we can now make a new conclusion about the difference between highway systems with and without passing lanes. Those highways with passing lanes are more likely to have more varied values of average speeds.

<u>Hypothesis III:</u> The relationship between number of lanes and average following distance is logarithmic in shape.

To take advantage of the ease with which one can add more lanes to model, we can explore the effects of increased numbers of lanes. It is well-known that increases in the number of lanes can decrease the level of congestion. In the presence of the same number of cars on the road at any given time, how do different numbers of lanes affect average following distance between cars?

I run the simulation for 1 through 10 lanes, each 50 times, to get my results. I find the following relationship.

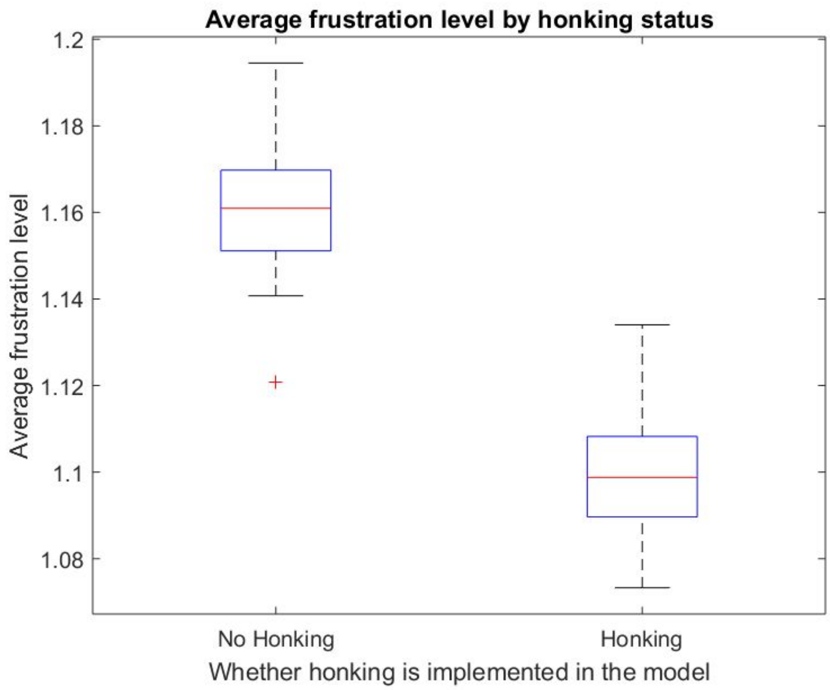


As we can see in the graph above, we do witness an initial steep rise in average following distance with an increased number of lanes However, by the time the road is 4 lanes wide or wider, average following distance does not respond much to the number of lanes on the road, all else equal. I fit a logarithmic curve to the data, and we can see that it doesn't quite capture the relationship exhibited here, although the idea is correct, that there is an initial steep incline and then it becomes flatter. It seems as though natural logarithm does not quite capture the extent to which it flattens off, though. Thus, we can reject the hypothesis that the relationship between number of lanes and average following distance is logarithmic in shape.

Hypothesis IV: The addition of honking increases average frustration levels in comparison with the no-honking model.

In my model, one car honking affects the frustration levels of the cars surrounding it. This can be seen as a kind of contagion effect. On the other hand, honking relieves frustration of the individual driver. How does this spread of honking balance out? On average, are frustration levels higher or lower in the presence or absence of honking as a mechanism to relieve and spread frustration?

I ran the simulation 50 times each with honking and without honking. I found that the mean frustration level with honking under normal conditions is 1.0992, while the mean frustration level without honking under normal conditions is 1.1604. The mean frustration level of the model without honking is higher than that with honking. This is opposite to what is expected. We can see in the boxplot below that this difference seems to be significant. The two distributions of average frustration levels barely even overlap one another at all.



As we must reject the hypothesis that honking increases frustration levels in comparison with the no honking model, it is good to think about possible reasons why. The most telling reason may be because in the model, once a car honks, their frustration level is reset to the base frustration level, while when a car witnesses a honk, their frustration level only increases by a margin that may be significantly smaller than the jump from current frustration to base frustration level. These dynamics represent a reality in which the act of honking releases frustration completely, while the witnessing of honking only marginally affects the driver's own frustration level. Once a driver has honked or changed lanes, whatever original frustration level they had is now gone and the frustration level of the driver most likely remains relatively close to the base level after that. More work must be put in to determine whether these are representative of a reality and, if not, what relationships between honking, changing lanes, and frustration are best to use.

Code Validation and Limitations

I tested my functions by creating fake data for the *currentPositions* matrix, determining what the output should be by hand, and then seeing if the output of the function matches the expected output. In particular, this was the process I utilized in testing the *calcDistance* and *canChangeLanes* functions. The *initializeCar* function was tested by calling it on different car indices, time steps, and numbers of lanes. The *calcAcceleration* function required an enormous number of inputs, so I was not able to test it using a fake *currentPositions* matrix. Rather, I thought that testing it by viewing the simulation visualization was sufficient. This was shortsighted, however, because the simulation visualization did not help me to understand why the more illogical phenomena were taking place. I was not able to figure out why I had sometimes infinite or imaginary values for speed, or why sometimes the cars just vanish off the grid. I only realized the values for speed when I began to do hypothesis testing on the model--before then, I had not checked any of the values for speed.

Possible flaws with my model stem from issues surrounding the dynamics of the system. We see in the results that oftentimes leading cars are honking into the void -- because they are leading, we would think that they have lower levels of frustration as they can drive at their desired speed. However, they are seen to be honking, so they must actually have higher levels of frustration. This is opposite to what is expected. There may be a flaw in the relationship between honking and frustration, and again in the relationship between frustration and speed, in how quickly it increases. There may also be problems relating to the order in which different states are calculated in the simulation loop. First, the car's speed and position are calculated using the last time step's acceleration. Then, frustration is calculated while taking lane-changing into account. After that, acceleration is calculated for the current time step. This is a possible source of error as sometimes we are using the current time step values and at others we are using the previous time step. Because different states are in relationship with one another, application of Euler's method to them is becoming more muddled.

In calculating average speed, I also noticed that every so often, speed of a car would reach infinity. I am unsure as to what causes this phenomenon. I also have imaginary numbers for speed. My thought was that this may result from taking the square root of frustration in calculating acceleration. Frustration is supposed to be a number between 1 and 2, but it is possible that somehow it is going below 0. However, I checked, and it does not go below 0. I do not have any other square roots in the simulation so I am unsure as to what is causing this. In order to deal with this in absence of a solution, I excluded infinite or imaginary numbers from the analyses in calculating the results. An odd occurrence that I think is happening as a result is that once cars slow to a dead stop, sometimes they disappear off the road.

Another flaw with my model is the way in which cars are generated to begin driving on the road. While what would be most natural is if cars came onto the road between *minFollowingDistance* and *maxFollowingDistance* kilometers after the last car in that lane, that is not what is currently being modeled. My model has that if there are less cars currently on the road than is equal to the *numberOfCars* parameter, then if all lanes are clear at least *minFollowingDistance* kilometers, a car will enter the road on a random lane. This creates a delay in cars entering the road, compared to the previously mentioned method.

Conclusion

The modeling of traffic is an important endeavor as it can lead us to more informed policy decisions without having to go through the trouble of building infrastructure that does not serve us best. If the simulation is accurate to reality, it can tell us a lot.

Findings of this model suggest that increasing one’s following distance not only leads to increased average speeds, but also decreased variation in speed, so less pumping on the brakes. These are appealing results to fuel-conscious drivers to encourage not only more efficient but safer driving. When it comes to the implementation of a passing lane, it is unclear whether passing lanes actually help increase overall speed of traffic. The results of the model suggest that passing lanes do not help increase overall speed of traffic, but rather create more variation in average speeds. With increasing congestion, many cities are thinking of expanding their highways to incorporate a larger number of lanes. The results of the model suggest that, given a constant amount of traffic, the benefits of one additional lane really flatten out starting at a four lane road. Lastly, the results of the model suggest that enabling cars to honk in non-emergency situations decreases overall frustration of drivers. This goes against common thought. More research is needed in order to make conclusions about these results.

In simulation, there are always limitations to the model, but this model in particular has aspects to it that may be skewing results. Future work could include studying the relationship between frustration and honking empirically so that models can have a basis for the nature of the relationship between the two. With that, models can ensure higher levels of accuracy and make more accurate policy predictions.

Works Cited

Luo, Y., & Bölöni, L. (2012, June). Modeling the conscious behavior of drivers for multi-lane highway driving. In *7th International Workshop on Agents in Traffic and Transportation (ATT-2012)*(pp. 95-103).

Olstam, J. J., & Tapani, A. (2004). *Comparison of Car-following models* (No. VTI report 960A).

Wilensky, U. & Payette, N. (1998). NetLogo Traffic 2 Lanes model. http://ccl.northwestern.edu/netlogo/models/Traffic2Lanes. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.