**Research Question**

Traffic jam is a common problem as well as a hot research topic that every driver has to face at least once in their lifetime. Multiple factors are involved in the cause of this issue e.g. driver’s behavior, quantity of vehicles, weather…etc. To avoid such inefficient and time-wasting circumstance, the driver normally prefers to stay informed about the traffic situation before they depart. However, the information is not always up-to-date due to the fact that traffic jam is not absolute predictable and is usually not timely reported. If the newest information can be delivered to every driver/vehicle in real time, the ongoing traffic can then be diverted to the other route, instead of all arriving at the same jam. Therefore, how to construct an ideal updating and communicating system which helps the vehicles with avoiding the traffic jam in advance is the main goal of this project.

**State of the Art**

* **Traffic Simulation**

The modeling of traffic flow has been quite popular since the later half of the 20th century. Existing models, like the Macroscopic ones (e.g. the Lighthill-Whitham-Richards model) where the entire traffic flow is represented by mathematical equations, mostly continuous, ignoring the individual drivers can often produce realistic outputs but lacks the complexity to model the complex driver behaviors. [1]. More popular models include treating the vehicles as individual agents with a set of given rules to follow. Models as such like the cellular automata model designed by Nagel and Schreckenberg are simple in construction but are able to exhibit the complex phenomena associated with real traffic. [1][2] Also in Microscopic models, classed as discrete models, which model the individual entities separately at a high level of detail, the individual entities (vehicles) are tracked separately as they interact with other vehicles and the environment. The interactions are largely governed by car-following and lane-changing logic.[3][4][5] . One well-known model, among various vehicle following models, which attempt to describe vehicle following based on anti-collision concept is the “Gipps model”, developed in 1981 [5], [6]. In this model a vehicle always aims to be able to stop safely if the vehicle it is following performs an emergency stop [6]. Another class of models known as the “Psycho-physical” vehicle following models attempt to capture both the physical and human components of vehicle control. They do this by maintaining a vehicle state, where the current state is determined through the differences in speed and distance to the leading vehicle [4].

* **Agent Based Simulation**

A relatively new concept, Agent based systems have gained popularity over the years. Agents defined as the computing entities, which receive input, form their environment and react in a manner that influences their environment. Agents have objectives, and can aim to achieve these objectives through plans or actions that they decide to perform [7] . In agent based traffic simulation, the environment is the road network and the agents are the vehicles who can influence the agents in its sphere of influence [5] .

* **Cloud based communication**

Tesla has not only made pure electric cars possible on the market, but also introduce the swarm ­intelligence-learning network among their Tesla vehicles. This implantation, for example, collects the traffic information (construction, accident, speeding camera…) from each car while they are being driven around the globe, and then uploads it to a global server so that every other car can access to it.

* V2X communication
* Sensors for collision avoidance

**Methods**

Since the design and development process for multi-agent simulations is different to traditional systems, literature suggests that the modeling is essentially a 7-stage process: brainstorming, theory, hypothesis, flowchart, code, analyze and publish [5]. Although complex design and development methodologies [8] like MESSAGE, PASSI, Tropos, Prometheus, MaSE etc. exist, for the purpose of this project, existing methodologies such as the Agent UML will be incorporated also keeping in mind the 7-stage process. The idea is to write the Multi-agent traffic system in an OO language like C++ or Java. This is mainly because agents can be considered as an extension to objects.

A multilane traffic scenario with different vehicles (agents) will be implemented in ROS (Robot Operating System). The integrated Stage simulator is chosen to be the testing environment because of its readily available models such as laser scanner and ultrasonic sensors, which are going to significantly reduce the time we invest in building up basic functions and in turn focusing more on designing agent’s behavior.

The simulation is split into main three parts, such as modeling of the world, the vehicle, and traffic scenarios. According to different scenarios, one or more lanes may be blocked by an obstacle that either slightly or heavily impact the traffic flow. While the former arriving group of vehicles might be stuck in the traffic jam, the later coming traffic will need to coordinate with each other in order to bypass the obstacle or choose another route in a most efficient manner. In addition, the vehicle model will consist of the dynamics of real vehicle aptly scaled for simulation so that the results shall be realistic.

**References**

[1] P. Ball, *Critical Mass: How One Thing Leads to Another*. London: Arrow, 2005.

[2] K. Nagel and M. Schreckenberg, “A cellular automaton model for freeway traffic,” *J. Phys. I*, vol. 2, no. 12, pp. 2221–2229, Dec. 1992.

[3] G. Lind *et al.*, “Best Pratice Manual,” *Deliv. D8 SMARTEST Proj. Disponível Em Httpwww Its Leeds Ac Ukprojectssmartestdelivd8 Html Acessado Cm 05 Maio 2002*, 1999.

[4] T. Schulze and T. Fliess, “Urban Traffic Simulation with Psycho-physical Vehicle-following Models,” in *Proceedings of the 29th Conference on Winter Simulation*, Washington, DC, USA, 1997, pp. 1222–1229.

[5] “Traffic Simulation using Agent-Based Modelling | Andrew Lansdowne.” .

[6] G.-L. Chang and T. Junchaya, “Simulating Network Traffic Flows with a Massively Parallel Computing Architecture,” in *Proceedings of the 25th Conference on Winter Simulation*, New York, NY, USA, 1993, pp. 762–770.

[7] J. Ferber, *Multi-agent systems: An introduction to distributed artificial intelligence*, 1 edition. Harlow: Addison-Wesley Professional, 1999.

[8] M. Kolp, B. Henderson-Sellers, H. Mouratidis, A. Garcia, A. Ghose, and P. Bresciani, *Agent-Oriented Information Systems IV: 8th International Bi-Conference Workshop, AOIS 2006, Hakodate, Japan, May 9, 2006 and Luxembourg, Luxembourg, June 6, 2006, Revised Selected Papers*. Springer, 2008.

**Timeline**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | April | | | May | | | | | June | | |
| 4/10 | 4/17 | 4/24 | 5/1 | 5/8 | 5/15 | 5/22 | 5/29 | 6/5 | 6/12 | 6/19 |
| Expose |  |  |  |  |  |  |  |  |  |  |  |
| - Research questions |  |  |  |  |  |  |  |  |  |  |  |
| - Project timeline |  |  |  |  |  |  |  |  |  |  |  |
| - Research State of the art |  |  |  |  |  |  |  |  |  |  |  |
| - Project structure |  |  |  |  |  |  |  |  |  |  |  |
| Model the system |  |  |  |  |  |  |  |  |  |  |  |
| - UML-diagrams |  |  |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |  |  |  |  |  |
| - Set up platform |  |  |  |  |  |  |  |  |  |  |  |
| - Implement the world |  |  |  |  |  |  |  |  |  |  |  |
| - Implement a single car |  |  |  |  |  |  |  |  |  |  |  |
| - Test run |  |  |  |  |  |  |  |  |  |  |  |
| - Expand the population |  |  |  |  |  |  |  |  |  |  |  |
| Project Report |  |  |  |  |  |  |  |  |  |  |  |
| Project Presentation |  |  |  |  |  |  |  |  |  |  |  |