5-1-2020

Argumentation among self-driving vehicles

Tharun Reddy Katukuri
Southern Illinois University Carbondale, tharunreddy1996@gmail.com

ARGUMENTATION AMONG SELF-DRIVING VEHICLES

by

Tharun Reddy Katukuri

B.Tech., SR Engineering College, 2018

A Thesis
Submitted in Partial Fulfillment of the Requirements for the
Master of Science Degree

School of Computing
in the Graduate School
Southern Illinois University Carbondale
May 2020

THESIS APPROVAL

ARGUMENTATION AMONG SELF-DRIVING VEHICLES

by

Tharun Reddy Katukuri

A Thesis Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Master of Science

in the field of Computer Science

Approved by:

Dr. Henry Hexmoor, Chair

Dr. Koushik Sinha

Dr. Dunren Che

Graduate School
Southern Illinois University Carbondale
April 9, 2020

AN ABSTRACT OF THE THESIS OF

Tharun Reddy Katukuri, for the Master of Science degree in Computer Science, presented on April 8, 2020, at Southern Illinois University Carbondale.

TITLE: ARGUMENTATION AMONG SELF-DRIVING VEHICLES

MAJOR PROFESSOR: Dr. Henry Hexmoor

In this day and age where the number of vehicles that are being used on highways and roads has been increasing considerably, there is a need for a global driving technique, or a driving phenomenon, where the vehicles can communicate with each other and maintain efficient driving positions by automating the process without the help of a human driver. With the available technology, self-driving cars are already under the spotlight, but these vehicles only offer limited support to the driver and they require human input in the process of driving.

Argumentation techniques can be used to develop an efficient algorithm to resolve the conflicts between Agents i.e vehicles to allow safer travel, reduced emissions and better traffic distribution over road networks. Considering the importance of cooperative driving. platoon transition that has been overlooked in the existing research, our implementation tests the use of an Argumentation technique, on top of the platoons, providing an edge over the existing work related to self-driving vehicles.

Utilizing the Argumentation allowed an effective way in resolving the conflicts among platoon leaders allowing a smoother transition of platoon groups. The conducted experiment compared the traffic flow of vehicles between two scenarios namely cooperative driving and non-cooperative driving, deriving the results that showcase the advantages of cooperative driving and also the role of argumentation in conflict resolution among vehicle agents.

ACKNOWLEDGMENTS

I would like to thank:

Dr. Henry Hexmoor for making me understand the topics and for Guiding me all the way through during my time of work.

Dr. Koushik Sinha and Dr. Dunren Che for their valuable suggestions and insightful thoughts on the topic.

TABLE OF CONTENTS

CHAPTER PAGE

ABSTRACT i

ACKNOWLEDGMENTS ii

LIST OF TABLES iv

LIST OF FIGURES v

CHAPTER 1 – INTRODUCTION 1

CHAPTER 2 – LITERATURE REVIEW 2

2.1 Concept of Platooning 2

2.2 Platooning Protocol focused on communication among vehicles. 4

2.3 Usage of priority value for conflict Resolution 6

2.4 Autonomous driving with no cooperation among agents 8

2.5 Cooperative Driving and Argumentation. 8

2.6 Utilization of abstract social argumentation 12

CHAPTER 3 – PROBLEM DESCRIPTION 13

CHAPTER 4 – METHODOLOGY 16

CHAPTER 5 – RESULTS/CONCLUSION 24

REFERENCES 31

VITA 34

LIST OF TABLES

TABLE PAGE

Table 4 1 – Request stack maintained by platoon leader. 19

Table 4 2 – Request stack after the ordering of requests. 20

Table 4 3 – Decision list that is shared by platoon leaders. 23

LIST OF FIGURES

FIGURE PAGE

Figure 2 1 – A Platoon Implementation by SPARTE [3] 3

Figure 2 2 – State transition diagram of vehicles in a platoon state [5].. 5

Figure 2 3 – Priority-based Framework, a) implementation. B) major components. 7

Figure 2 4 – Communication among vehicles[12].. 9

Figure 2 5 – Functional architecture for cooperative automated maneuvering.[13].................... 10

Figure 2 6 – The life cycle of argument formation, conflict resolution[16] 11

Figure 3 1 – Traffic Deaths, 2009-2018……………………………………………………14

Figure 5 1 – Testbed used for simulation………………………………………………...24

Figure 5 2 – Platoons formed during the test run.. 25

Figure 5 3 – Arguments generated and their outcomes. 26

Figure 5 4 – Average Travel time comparison. 27

Figure 5 5 – Average Speed comparison. 28

Figure 5 6 – Slowdown count. 28

Figure 5 7 – Lane change count 29

CHAPTER 1

INTRODUCTION

Self-Driving Vehicles, or driverless vehicles, are still a possibility for the near future. The road to get there is already being paved with available driver-assist features such as lane assist, blind-spot monitoring and various driver aid software features.

Different cars are capable of different levels of self-driving and are often described by researchers on a scale of 0-5 called the *Layers of Autonomy.*[1]

- **Level 0**: All major systems are controlled by humans
- **Level 1**: Certain systems, such as cruise control or automatic braking, may be controlled by the car, one at a time
- **Level 2**: The car offers at least two simultaneous automated functions, like acceleration and steering, but requires humans for safe operation
- **Level 3**: The car can manage all safety-critical functions under certain conditions, but the driver is expected to take over when alerted
- **Level 4**: The car is fully-autonomous in some driving scenarios, though not all
- **Level 5**: The car is completely capable of self-driving in every situation

We can safely say that the vehicles that are available nowadays have reached level – 4 of autonomy. Automotive companies, such as Tesla, are pioneering the driver assist technology in such a way that has been never thought of before.

The proliferation of low-cost wireless connectivity, combined with the growth of distributed peer-to-peer cooperative systems, is transforming next-generation vehicular networks. Drivers and passengers inside moving vehicles will be able to obtain and share their interested data, such as MP3 music, news, and video clips[4].

CHAPTER 2

LITERATURE REVIEW

This chapter examines the literature on self-driving car mechanisms, lane shifting techniques, and Argumentation and its various applications that have been used. Following a detailed account of the research plan of action, the prior research linked to Autonomous driving vehicles and the concept of Argumentation and its usage (including its architecture, definitions, methodologies, policies, procedures, algorithms, relationships and characteristics) will be discussed. The literature related to Autonomous driving vehicles is reviewed by summing up a detailed discussion on vehicular platoons, smart vehicles equipped with microcontrollers, algorithms implementing lane shifting and conflict resolution among vehicles by using Argumentation.

I have used several research databases containing publications, articles and peer reviews on the topic. The work also combines the research of other dissertations, proposals and prospects related to the topic to ensure originality and to determine what has already been examined within the field about the technology and advancements in the field of Autonomous Driving.

Every review of a topic is the combination of the results and research until then, so knowing is covered most of the things.

2.1 Concept of Platooning

The platooning concept can be defined as a collection of vehicles that travel together, actively coordinated in a formation. Some expected advantages of platooning include increased fuel, traffic efficiency, safety and driver comfort. There are many variations within the implementation of the concept such as the goals of platooning, how it is implemented, mix of vehicles, the requirements on infrastructure, what is automated (longitudinal and lateral control) and to what level it's being automated. [2]

Vehicle-to-Vehicle(V2V) communication allows the sharing of local vehicle signals, such as speed and sensor data, among vehicles in the platoon. These shared signals are used in the control algorithms of the platoon. The platoon forms a cooperative system where sensing, control algorithm and actuation are distributed throughout the platoon and data is communicated between vehicles. Automatic control over an individual following vehicle is partly external from the lead vehicle and partly internal from the systems and sensors in the following vehicle itself. The following vehicles automatically strive to maintain the specified gap to the vehicle in front and the path and trajectory as specified by the lead vehicle[3], Figure 2 1 describes the platoon implementation. Furthermore discussion is available in[6].

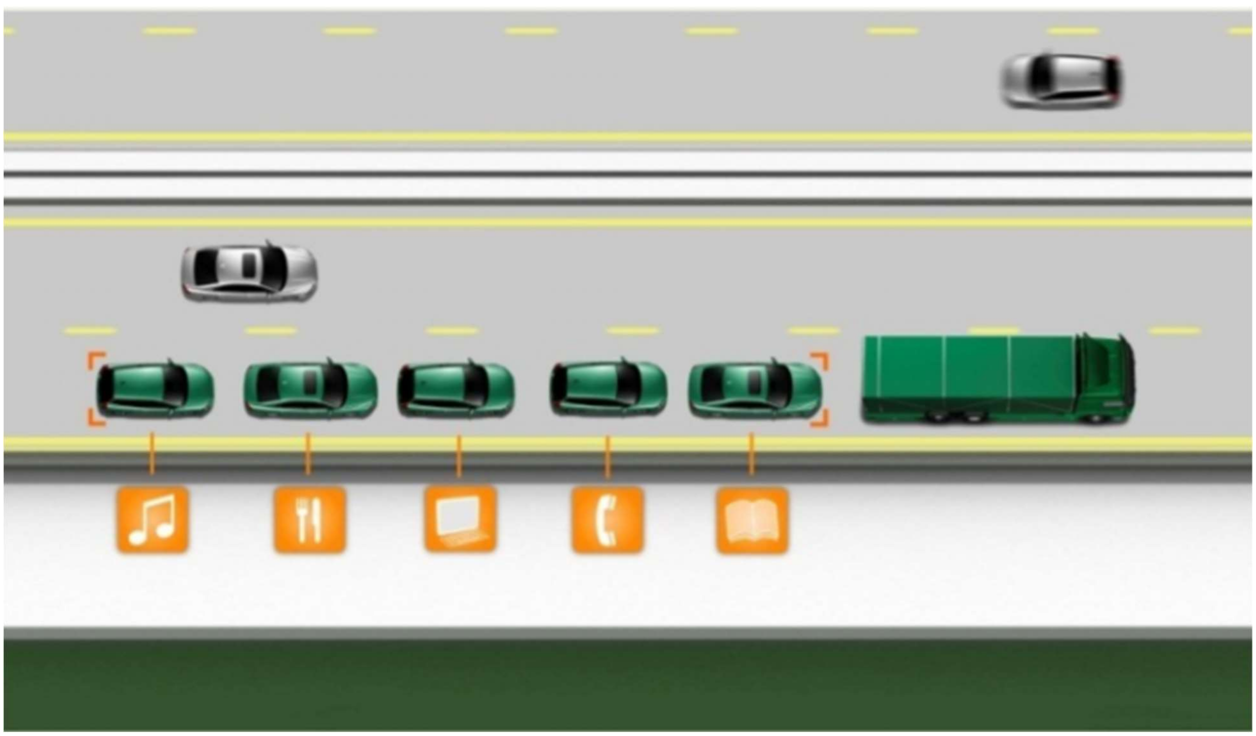

Figure 2 1 – A Platoon Implementation by SPARTE [3]

2.2 Platooning Protocol focused on communication among vehicles

In this section, the platoon protocol, in general, is discussed. At any given time, each vehicle stays at one of the following states[5]: 1) *Initial*; 2) *Join*; 3) *Quasi-Split;* 4) *Split*. When a vehicle enters the network, it is at the *Initial* state. Later, when it meets other vehicles in the same direction, it may join them as a platoon member. After one vehicle is detected to join the platoon, it enters the *Join* state and sends out a *platoon-join* message to all platoon members to announce that a new member has joined the platoon. The message contains the information of *vehicle ID*, its *interest list*, data *list*, and *buffer size*. As the platoon leader receives this message, it will use the information to determine the best data replication arrangement for the next replication cycle. When one vehicle detects mobility anomaly, it switches its state to *Quasi-Split*, where the anomaly will be further analyzed. If the anomaly comes from the change of road layout (e.g., the platoon is passing a curving road), the anomaly is resolved, and the vehicle returns to the *Join* state. Otherwise, if the vehicle is detected to split from the platoon, it enters the *Split* state. It sends out a *platoon-split* message to inform other platoon members that it is going to leave the platoon. At the same time, it starts to prefetch its interested data and transfer its buffered data to nearby platoon members. there is a possibility for a message being lost for a multitude of reasons such as channel interference or collisions. However, this will not affect the performance of the platooning protocol too much because both join and split actions can always be detected by neighbouring vehicles through its beacons.

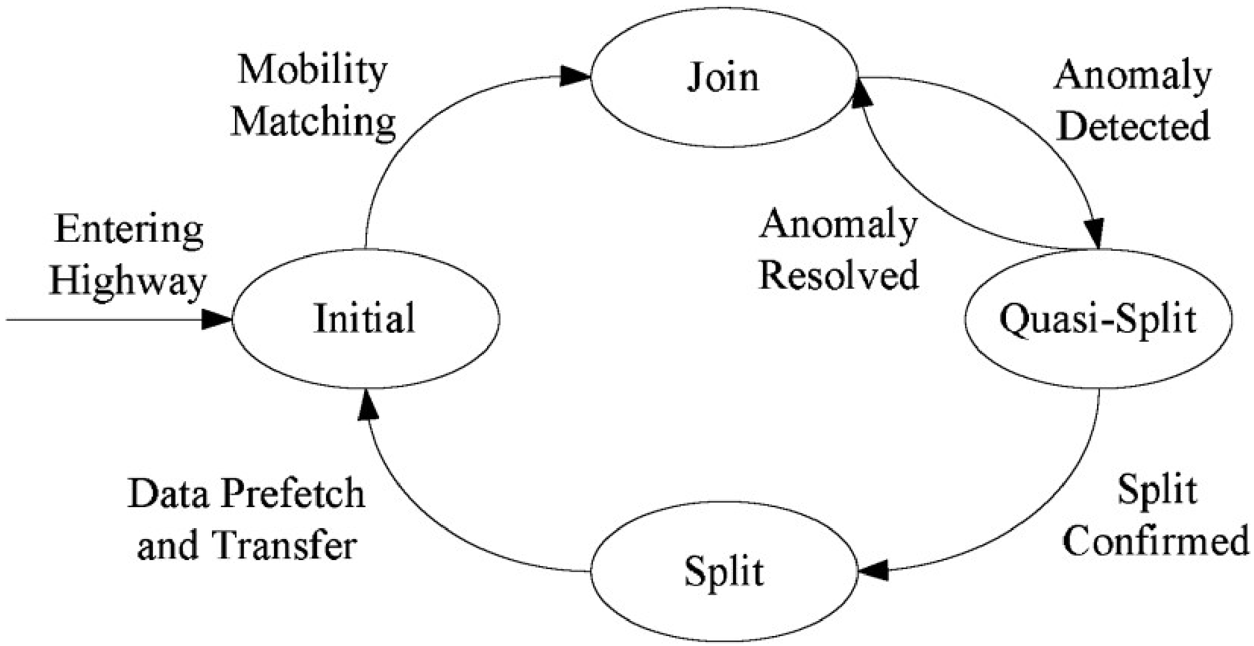

Figure 2 2 – State transition diagram of vehicles in a platoon state [5]

Furthermore, existing reliable and efficient broadcasting techniques [7] can be used to provide reliable message delivery.

vehicular platooning has been presented as a special case of crowd-sensing framework where sharing sensory information among a crowd is used for their collective benefit in[15], the formation of platoons, platoon management has been discussed with the results. The limitations of the presented model include but are not limited to is the use of an efficient algorithm that allows vehicles to either disband or join a platoon.

Consider the scenario in which, at a given point of time, multiple vehicles from different platoons disband their respective groups; this results in a conflict of interest among individual agents. The disintegration of individual vehicles from their respective platoon, if not properly controlled, may result in a high density of vehicles entering a new lane thereby causing Slowdown of vehicles and possible crashes.

One effective solution to fight this issue is to utilize an algorithm that can effectively allow the agents to disband from their respective platoon group in an orderly fashion, thus eliminating the issue of overpopulating vehicles in a limited space.

2.3 Usage of priority value for conflict Resolution

The following implementation[8] describes an Algorithm that provides conflict resolution among vehicles arriving at an intersection presenting right of way to each vehicle based on priority value that will be assigned by a controller at an intersection. The representation is shown in Figure 2 2, detailing the structure also the components that are being used.

- An autonomous vehicle sends a request to the intersection controller once it enters the cooperative area.
- The intersection controller works in discrete time. At the beginning of each time step, the intersection controller collects and processes the requests according to the *priority assignment policy* in an arbitrary order. In this simple system, the policy assigns priorities sequentially. In each step, the right-of-way is assigned to the vehicle which can pass the intersection with maximal throttle command and assigned the lowest priority. The goal of such design is to maximize the vehicle speed and reduce the vehicle sojourn time within the intersection. All requests that are not admissible in this step will be left to the next step for processing.
- The intersection controller notifies admitted vehicles to progress under given priorities. Non-admitted vehicles are required to stop in front of the stop lines.

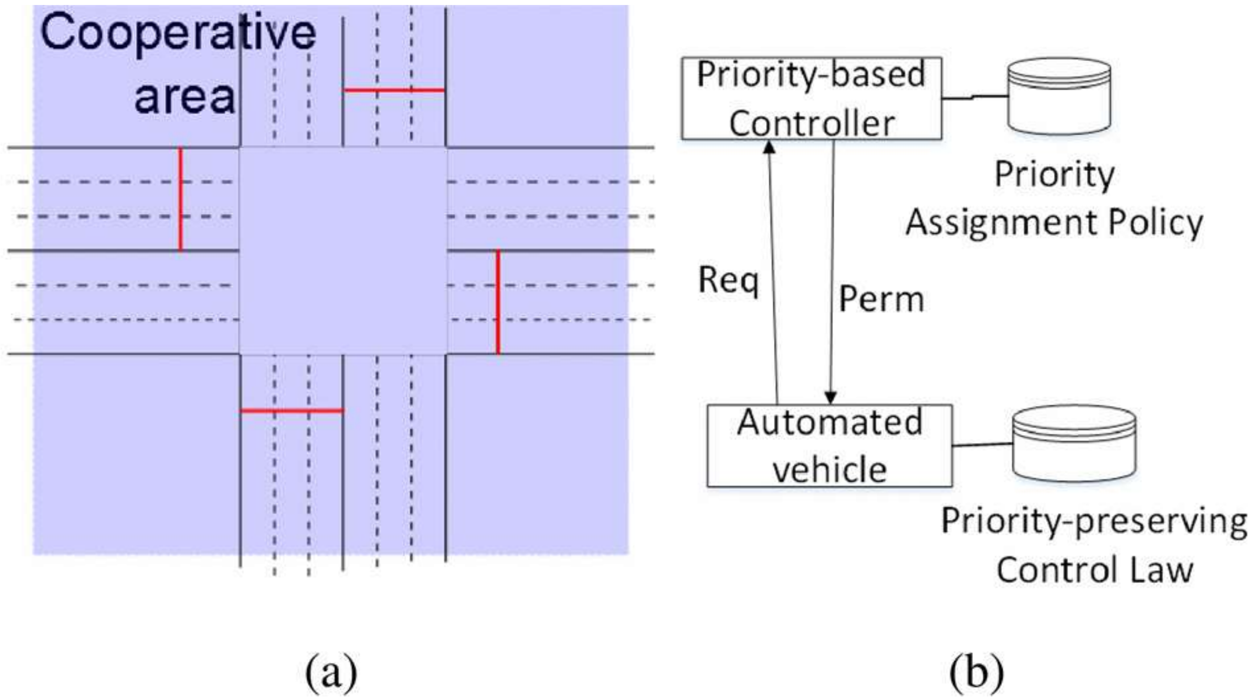

Figure 2 3 – Priority-based Framework, a) implementation. B) major components.

In a similar implementation, an effective solution to resolve the conflict of interest among vehicles at an intersection, based on a Control Framework, using User-selected priority values has been described in[9].

The discussed technique[8,9] utilizes Priority values to resolve conflicts and provides a Right of way to vehicles. A similar kind of attribute known as priority value will be implemented in our algorithm. This is to provide a value of strength to all the Arguments that will be generated by the agents. At the same time, priority value can be used as a variable to judge the importance of an agent's argument. The implementation takes place on a straight road in contrast to an intersection. Furthermore, Argumentation technique will be used in the resolution of conflicts.

This can be a viable solution in reducing the density of vehicles and also in avoiding crashes. Provided the following scenario, multiple vehicles disband a platoon by changing the lane(discussed in section 2.2).

2.4 Autonomous driving with no cooperation among agents

Previous research on this topic, most of it and implementation related to autonomous driving in the real world by automotive companies, is limited to a vehicle, or an Agent, having no cooperative or Argumentative structure to help resolve the conflicts during driving. Instead, there is always a huge factor of an agent sensing the environment and adjusting its speed and path to cope with the surrounding traffic or obstacle. This is the case even in driver-assist vehicles that are available today.

An example of a single agent processing its environment has been presented in[10]. An MPC strategy for CAS is developed to avoid or mitigate collisions in multi-lane roads by utilizing brake, throttle and steering control inputs in the presence of stationary or moving obstacles in all lanes. Road boundary constraints and moving obstacle collision avoidance constraints are integrated to generate convex safety constraints.

Furthermore, a detailed evaluation of different research approaches centered on autonomous driving based on single agents using various methods and techniques to process and adjust to the surrounding traffic has been presented in[11].

With the advent of communication capabilities among agents, a cooperative driving technique combined with above-mentioned capabilities related to a single agent sensing and adjusting to the surroundings could be a viable solution for self-driving vehicles

2.5 Cooperative Driving and Argumentation.

A cooperative driving technique using MPC and establishing a communication channel between the vehicles has been described in[12]. The following figure represents the vehicles being allocated on a two-lane road and the communication structure among the vehicles.

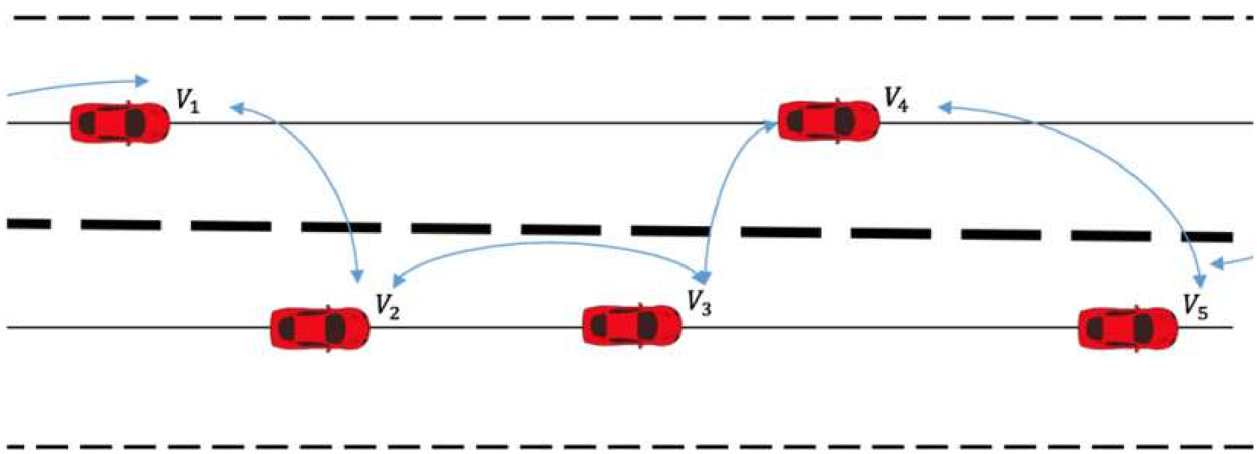

Figure 2 4 – Communication among vehicles[12]

An application of cooperative driving has been elaborately described in[13] as a means of communication among agents using protocols and also a controller equipped with the agents to process the information exchanged and using it to cooperate with the other agents on road.

The implementation has been described in the following Architecture that has been broken up into three levels as described below.

- The *Strategic Layer* is responsible for the high-level decision making regarding, e.g., routing, optimization of fuel consumption, travel times, and, in case of platooning, the scheduling of platoons based on vehicle compatibility, destination, and impact on highway traffic flow and infrastructure. To this end, the Strategic Layer may utilize cloud-based services.
- Driven by the Strategic Layer, the *Tactical Layer* coordinates cooperative manoeuvres, such as platoon forming, merging, intersection crossing and also speed synchronization between neighbouring vehicles, to support lane changes in heavy traffic. As such, this layer runs the interaction protocol. Depending on the type of application, the Tactical Layer can be implemented in a distributed or a centralized manner.
- The *Operational Layer* involves the actual real-time vehicle control to execute the

required maneuvers, amongst which platooning and merging.

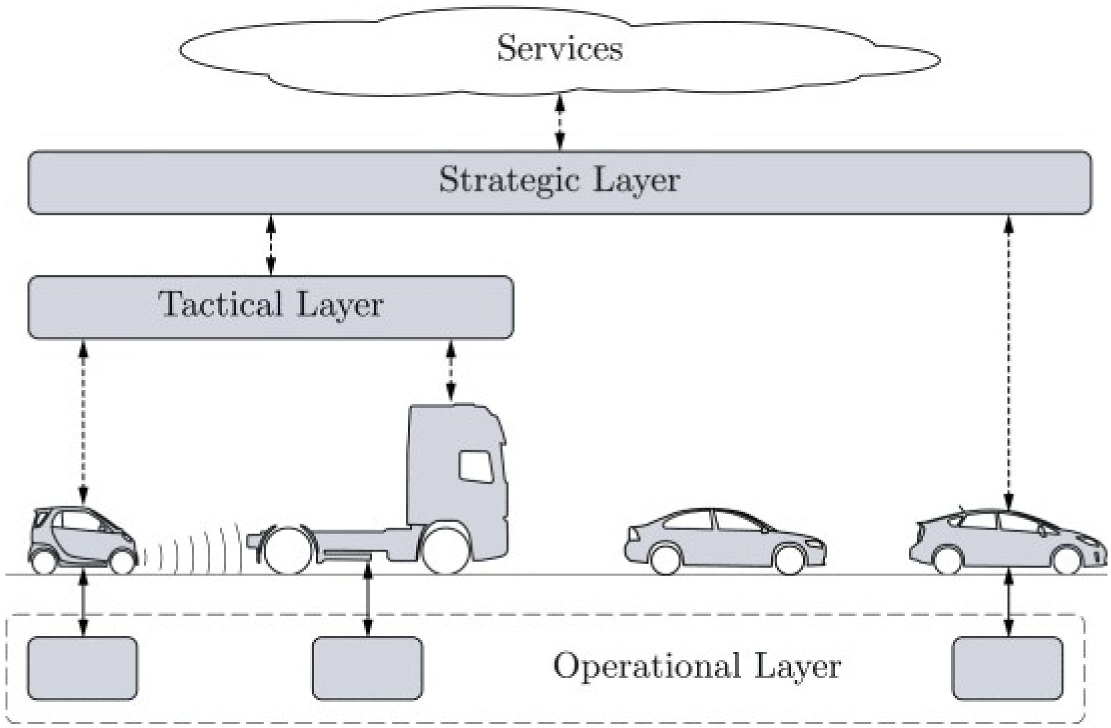

Figure 2 5 – Functional architecture for cooperative automated maneuvering.[13]

The research work in this report mainly focuses on the *Tactical layer* of Implementation. An algorithm to facilitate Cooperative driving will be elaborately discussed. The Strategic layer and the operational layer will be discussed elsewhere.

Furthermore, the implementation of a cooperative resolution of conflicts among agents is shown in[13]. Two scenarios have been discussed namely a protocol for an urban intersection and a lane reduction scenario for Highway. In the case of a Highway lane reduction, an effective method for vehicles in two different platoons trying to merge into one larger platoon has been described with a four-stage process involving communication between the agents.

The drawback for the mentioned Technique is, multiple agents are communicating and cooperating at a given point of time. The scenario for agents disbanding the platoon and newer vehicles merging into the platoon effectively has not been considered, these drawbacks are taken into consideration and to tackle the issues an effective algorithm has been implemented with the use of argumentation.

Furthermore, Argumentation[16] has been implemented on IOT Enabled Vehicles for an effective lane selection. A similar kind of implementation has been utilized in our algorithm where instead of a lane selection that is done by the agents based on social argumentation in which the outcome of an agents argument(win or loss) is dependent on social voting from other agents, Similar kind of social argumentation has been utilized to facilitate vehicle maneuver across different platoon groups and lane shifting. the lifecycle of argument is described in the following figure.

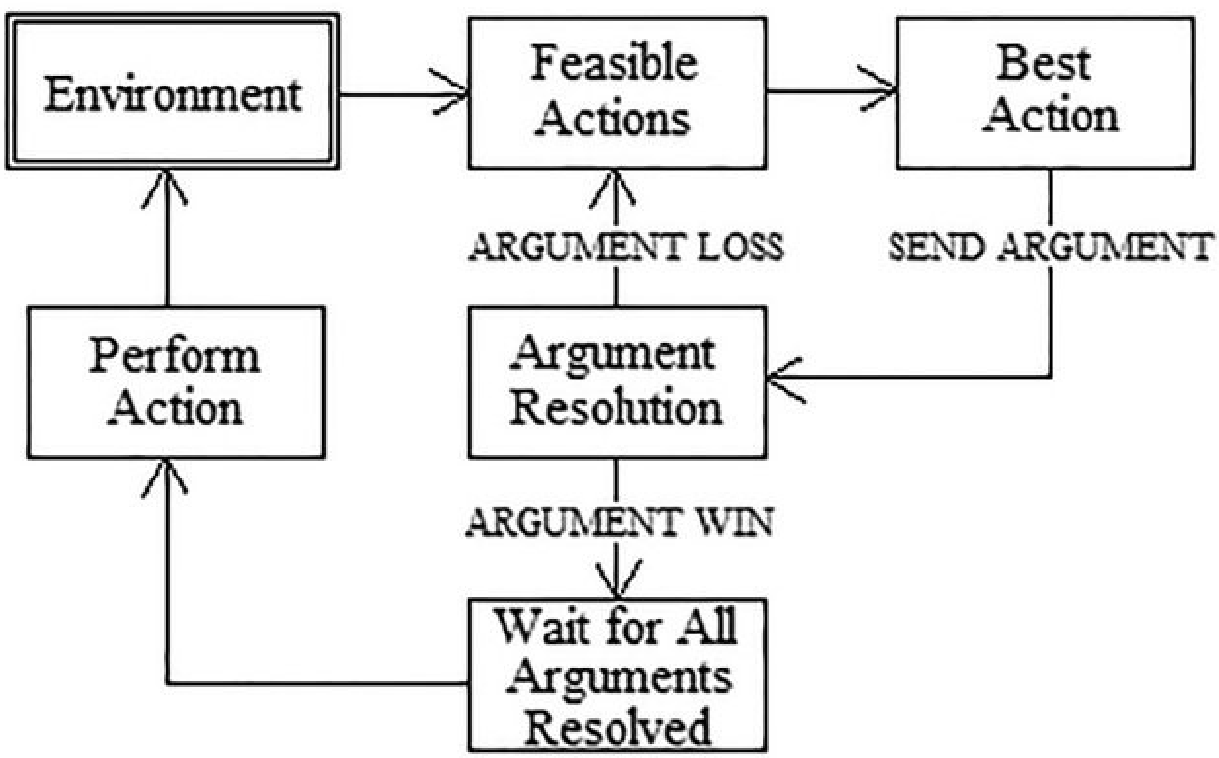

Figure 2 6 – The life cycle of argument formation, conflict resolution[16]

2.6 Utilization of abstract social argumentation

We are going to utilize the concept of Argumentation to resolve the conflicts. As described in[22], all the attacks have a value that weighs in on the strength of the argument. agents are entitled to raise an attack based on any of the arguments. Ultimately, social support by votes decides the winner out of all the attacks.

Considering the vehicles as individual agents and an agent can raise one attack at a given point of time, the attack will be based on an argument and that argument will have a value to define its strength. In this case, that value is the priority value(PL).

Social voting will be taking place among the agents ultimately deciding one agent as a winner. All the vehicles act as individual agents. Platoon leaders are considered as social agents and the voting takes place among the platoon leaders. An agent's vote is based on the argument and the priority value that backs up the argument's strength. Ultimately, the argument with the highest priority value will be the winner.

Individual agents which are not part of a group are also entitled to a vote. During a lane change, any agent which receives a request from another agent that is trying to occupy the lane will be subjected to argumentation. However, in this case, it will not be a social argumentation. Instead, it is a simple form of argumentation among agents.

CHAPTER 3

PROBLEM DESCRIPTION

The vehicles nowadays are equipped with level 4 Autonomy features such as lane assist, adaptive cruise control and emergency braking. Some manufacturers are making strides towards fully autonomous vehicles. Some of the vehicles, which are very limited, offer a level of fully autonomous driving.

Driver Assist technology is there to provide safety and, at the same time, reduce the stress on the human driver. During the advent of self-driving technology in recent years, the mentioned criteria have been fulfilled to some part.

The cooperative driving technique is still unused in vehicles. Instead, the Driver Assist features are a result of the vehicle, or the Agent, sensing the surroundings and making the adjustment to itself with no interaction with other agents on the road.

Based on the prior research on cooperative driving and also Argumentation techniques that help in resolving conflicts between two agents, there is a possibility for developing an Architecture considering vehicles as Agents and a dependable algorithm that helps to maintain fully autonomous driving among vehicles.

Human riders are prone to make errors while driving, this has been proven time and time again by the accidents that are happening on Roadways. Since a computer is less prone in making errors, Autonomous driving where an Agent, in this case, a vehicle that is controlled by a microcontroller can be much safer to ride also can avoid accidents.

The following table shows the variance in the number of fatalities occurred on roads each year, the data has been collected by NHTSA.

Year	Fatalities	Annual percent change	Fatality rate per 100 million vehicle miles traveled	Fatality rate per 100,000 registered vehicles
2009	33,883	-9.5%	1.15	13.08
2010	32,999	-2.6	1.11	12.82
2011	32,479	-1.6	1.10	12.25
2012	33,782	4.0	1.14	12.72
2013	32,893	-2.6	1.10	12.21
2014	32,744	-0.5	1.08	11.92
2015	35,484	8.4	1.15	12.61
2016	37,806	6.5	1.19	13.13
2017	37,473	-0.9	1.17	12.79
2018	36,560	-2.4	1.13	NA

Figure 3 1 – Traffic Deaths, 2009-2018.

Source: U.S. Department of Transportation, National Highway Traffic Safety Administration.

Considering the advent of Driver Assist features in Motor Vehicles, the Fatalities during driving Pretty much remain unchanged each year. This is because there is still a human error that results in accidents.

The advantages of cooperative driving compared to traditional driver-assist features are as follows:

Maintaining a communication channel between vehicles. This helps in data transfer and sharing information between two agents. An agent can let its neighbouring agent know about its future maneuver. At the same time, it can also receive information from the neighboring agent about its intention or future maneuver.

The primary cause of accidents on road is due to unexpected scenarios occurring without prior knowledge. This results in the driver making decisions that are unsafe and leads to an accident. Having a cooperative communication channel can eliminate the chance of accident by a substantial margin.

Having an organized manner of driving among the vehicles can result in efficient driving and reduce emissions and fuel consumption.

CHAPTER 4

METHODOLOGY

Assuming the set of vehicles on the road are v's = {v1, v2, v3, v4, …. Vn} and the platoon set p's = {p1,p2,p3, …..pn} and p's is a subset of v's.

Every vehicle on the road is entitled to its own **priority value('P')** that is calculated by two factors: 'the type of the vehicle' and 'the purpose of travel'. The range of the value P has been set from 1 to 10 for the sake of this experiment. The value '1' being the least prioritized, or weighted, and '10' being the highest weighted value in an argument between multiple agents. In general, the higher the priority value the more tendency that the vehicle's argument is won.

The leader of the platoon has information about the vehicles in its group. These include the vehicle id, average speed, type of vehicle and the priority value of the vehicles. The priority values are considered in resolving the arguments raised by individual vehicles.

The speed of a platoon is maintained in such a way that the vehicle with the lowest top speed or engine power will be able to catch up and maintain the speed that is equivalent to other vehicles in the platoon.

The following steps describe the formation of platoon also the communication channel that exists between the platoon leaders and the resolution of any arguments using argumentation.

Formation of platoon:

Consider the vehicles are not a part of a platoon. The responsibility of being a platoon leader is randomly initiated by any vehicle. This is a random phase where any vehicle can send a broadcast signal to its neighbouring vehicles mentioning its destination.

The vehicles which are on the receiving end of the spectrum send a request back to the Agent requesting to join it's platoon group. This happens if the destination is the same or at least some part of the destination is the same.

Considering the vehicle that takes initiative to send the broadcast signal as a platoon leader, the following pseudo-code explains the process.

An individual Agent can send the request not only at the time of receiving an invitation from broadcast, but it can Request a platoon leader that is travelling along for Permission to merge into the platoon.

The platoon leader can either Accept or Deny the Request based on the Size of its platoon. i.e, size = number of vehicles in the platoon group. Max size is dependent on the number of constraints, the platoon leaders communication range. Hardware limitations on the vehicles and network latency etc. The max size can be limited to a certain extent based on the effective communication that can be maintained across a platoon group.

```
Platoon_Leader_Broadcast(Invitation, Destination)
{
    Sends a Broadcast signal to neighbouring agents with an invitation to join the platoon
    along with information about the travel path or destination.
}
Neighbouring Agent(Request)
{
    If travel path is a subset of platoon leader's travel path then
    Send Merge Request to Platoon Leader and wait for Acceptance.
}
Platoon Leader()
{
```

```
        For Any Request received
        If
        {
                Size of the platoon < max size
                Accept the Merge Request
        }
        Else
                Reject the Merge Request.
}
```

Arguments Raised by Agents:

An individual agent can raise an Argument or, in this case, a request to the platoon leader at any point of the time during the travel. In this case, the request could be to leave the platoon or split the lane etc.

Since a platoon can have multiple vehicles, the decision to Approve or Reject the Request is managed by the Platoon Leader. This way there wouldn't be any conflicts between the vehicles in the platoon.

The platoon leader maintains a Stack Buffer of all the requests that are received from the platoon vehicles.

There is a possibility that there may be multiple requests at the same point of time. If the requests are not handled accordingly, there is a chance of an unresolved conflict among the two agents. To prevent this, the platoon leader will order the requests and prioritize them based on their priority value 'P'.

The requests are handled in such a way that the agent with the highest priority value has it's argument pushed towards top of the list. Likewise, all the remaining agents and its arguments will be ordered based on their priority value.

The following example demonstrates the stack ordering. Assuming the platoon 'p' has a total of 4 vehicles, the platoon leader is assumed to be 'pl', assuming the vehicles have generated arguments based on the individual agent's interest. The set of arguments raised by individual agents are {arg1, arg2. ..arg4}. Since an agent can only raise one argument at a given point of time and assuming the 'P' value to be constant, the arguments raised by individual agents are stacked in a 'decision stack' by the platoon leader and the leader executes the arguments or the requests based on the 'decision stack'

Below is the figure representing the Request stack that is stored by platoon leader before ordering the requests.

Table 4 1 – Request stack maintained by platoon leader.

Vehicle id	argument	Priority value
V3	Arg1	3.0
V1	Arg2	4.0
V2	Arg3	8.0
V4	Arg4	2.0

The below figure represents the arguments that are ordered based on priority value and stores it in the decision stack.

Table 4 2 – Request stack after the ordering of requests.

Vehicle id	argument	Priority value
V2	Arg3	8.0
V1	Arg2	4.0
V3	Arg1	3.0
V4	Arg4	2.0

Process that takes place on a vehicle that's inside a platoon:

```
V_id(priority value, request )

{

        If(change in direction or unsatisfactory platoon speed)

                {

                        Send request to platoon leader:

                                Send Arg, vehicle_id, priority value('P')

                }

        Else()

                {

                        Follow the platoon leader

                        No request sent.

                }

}
```

Resolving the Arguments by platoon leader:

The platoon leader also maintains a communication channel with other platoon leaders. Since it is important to maintain a steady flow of traffic, the leader agents will allow the agent to disband from their platoon when there is enough room for the vehicle to pass by.

The platoon leader broadcasts the information to other platoon leaders regarding the Request from the vehicle inside its platoon and this vehicle is the one which is on top of the list from its request stack with highest priority value out of all.

Since all the platoon leaders will be broadcasting regarding one of the vehicles from their platoon, there will be another stack buffer maintained by platoon leaders and this stack contains the information regarding the agents from different platoons. They will be ordered based on the priority value.

In this case, the agent which provides a request with the highest 'Priority value' will be able to let the agent with the argument pass first. After the vehicle has left the platoon, the leader notifies all the other agents regarding the status of vehicle.

This process takes place with platoon leader('PL') during intra-platoon communication:

```
PL(requests, vehicle_ids, Priority values )
{
	Accept requests from Agents()
	{
		Request are stored in a stack along with priority values and vehicle id.
		Initial_stack (vehicle_id, Argument, Priority value)
	}
	Sorts the requests based on priority value – argument sort()
	{
		Decision_stack(vehicle_id, Argument, Priority value)
	}
```

```
        Execute the Arguments()
        {
                Broadcast top most argument with other platoon leaders.
                Bcast(argument, Priority value)
                Read data from other platoon leaders(argument, priority value)

                If( local Agents ‘p’ value > all other received ‘P’ values)
                {
                        Wait for acknowledgement of approval from all the Agents;(social voting)
                }

                Elseif( local Agents‘P’ value is < any one of received ‘p’ values )
                {
                        Find agent with highest ‘p’ value
                        Send the acknowledgment of approval;
                }
        }
}
```

The arguments requested by the agents within the platoon are processed by the platoon leader and stored in a decision stack. Requests will be implemented one-by-one.

The figure represents Argument buffer between platoon leaders.

Table 4 3 – Decision list that is shared by platoon leaders.

Agents (Platoon leaders)	'P' value of their requests or arguments.
A2	3.5
A3	4.2
A4	2.4
A5	1.7
A6	8.0
A7	5.6

From the table above we can observe that Agent-6(platoon leader – 6) has the highest priority value for its argument. So, the Agents(Platoon Leaders) collectively will let the agent 6 make its move or let one of the vehicles in its group shift the lane and pass up. The agent which has won the argument sends an update(acknowledgement) to other Agents once the vehicle has exited its platoon.

After the acknowledgement is received, the agents will try to resolve the arguments again based on the priority value and the process thus follows.

CHAPTER 5

RESULTS/CONCLUSION

The simulation is conducted using Matlab. The following figure shows the Road map with the source and Destination on which the test has been conducted.

The map has 3 distinct sources for vehicle origins and two Destinations. In the initial run without implementing the cooperative driving, the vehicles travel across the Map utilizing Lane shift technique and occasional slowdowns.

With the co-operative driving, the vehicles can form platoons, communication has existed between the platoon leaders. Vehicles merge and leave the platoon accordingly also an aspect of Argumentation exists between vehicles when there is a conflict during Lane Change.

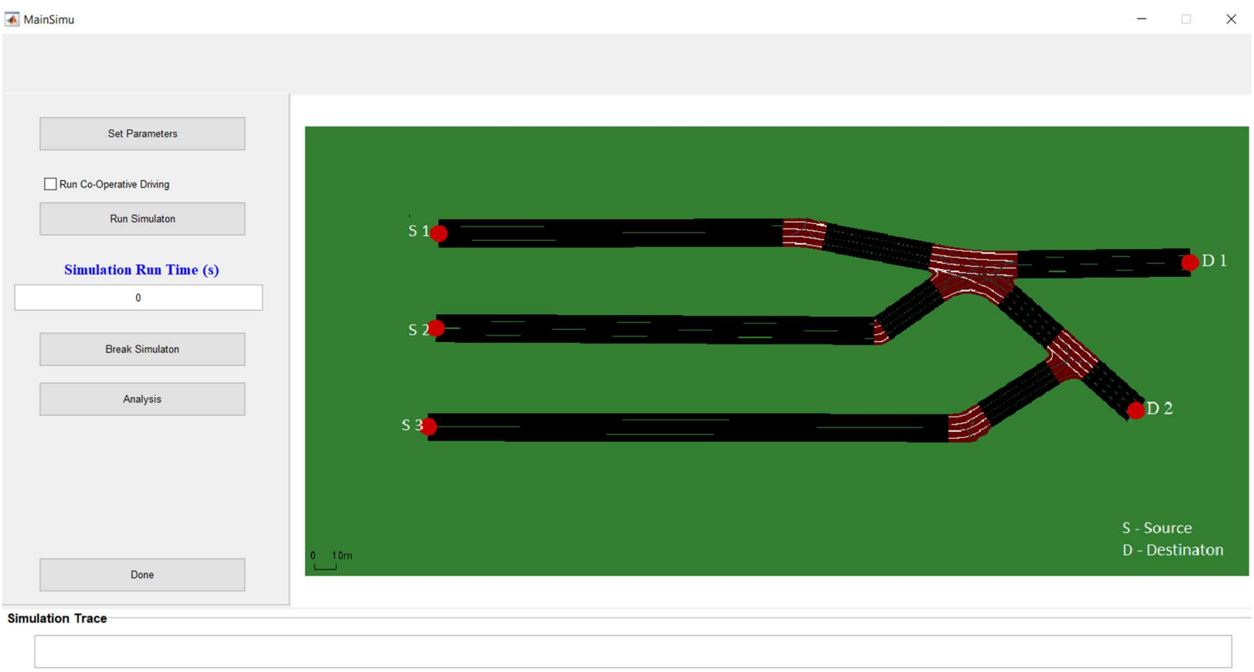

Figure 5 1 – Testbed used for simulation

The simulation conducted has two different run time options, a non-cooperative run and a cooperative algorithm. The following (Figure 5 2) represents the observed results from the cooperative algorithm testing.

Represented in the graph are number of vehicles on the Y-axis as count and the simulation time in seconds on x-axis. The platoon groups that are formed during the runtime, along with the total number of platoon merge requests and the accepted requests, are represented.

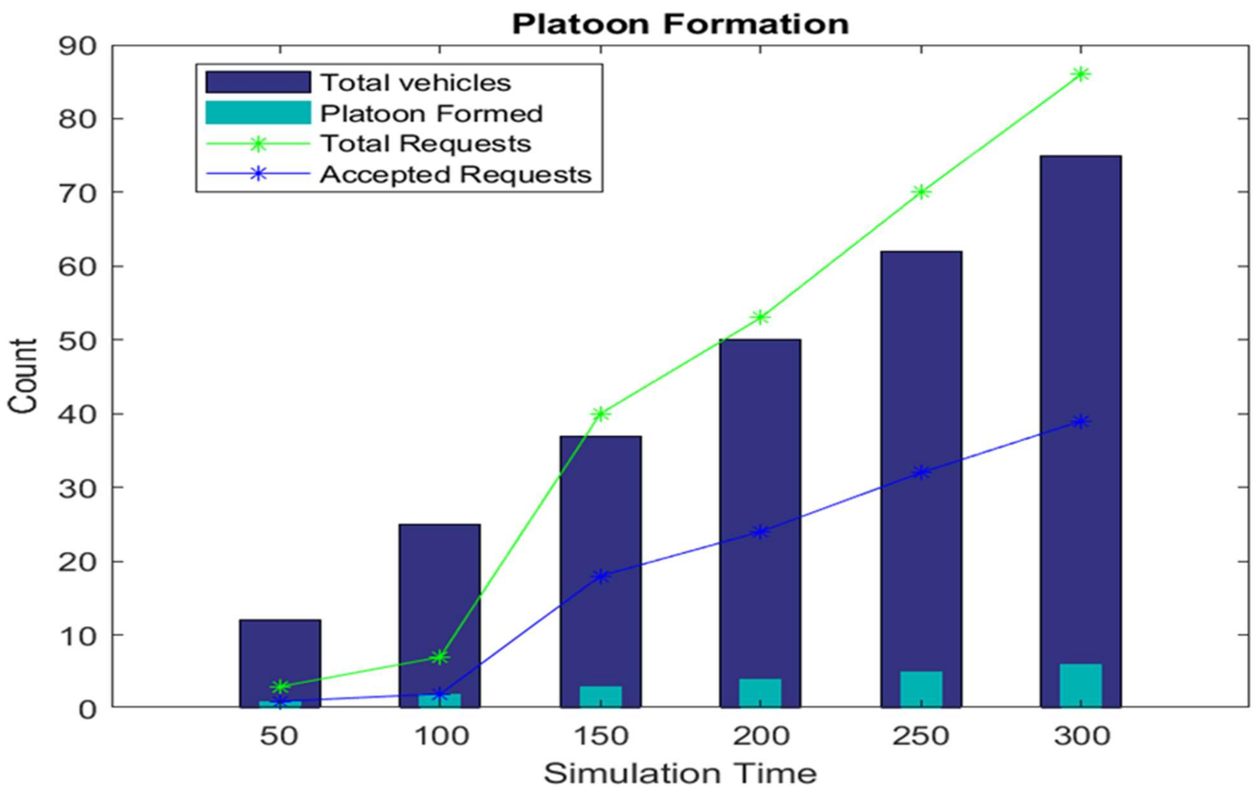

Figure 5 2 – Platoons formed during the test run

The number of Arguments that are generated by individual agents and the outcomes based on the argument are represented in (Figure 5 3). The X-axis represents the time in seconds and the y-axis represents the Argument generation count. The total arguments generated, number of rejected arguments, and the Arguments resulting in approval of lane change or a request to slowdown are presented.

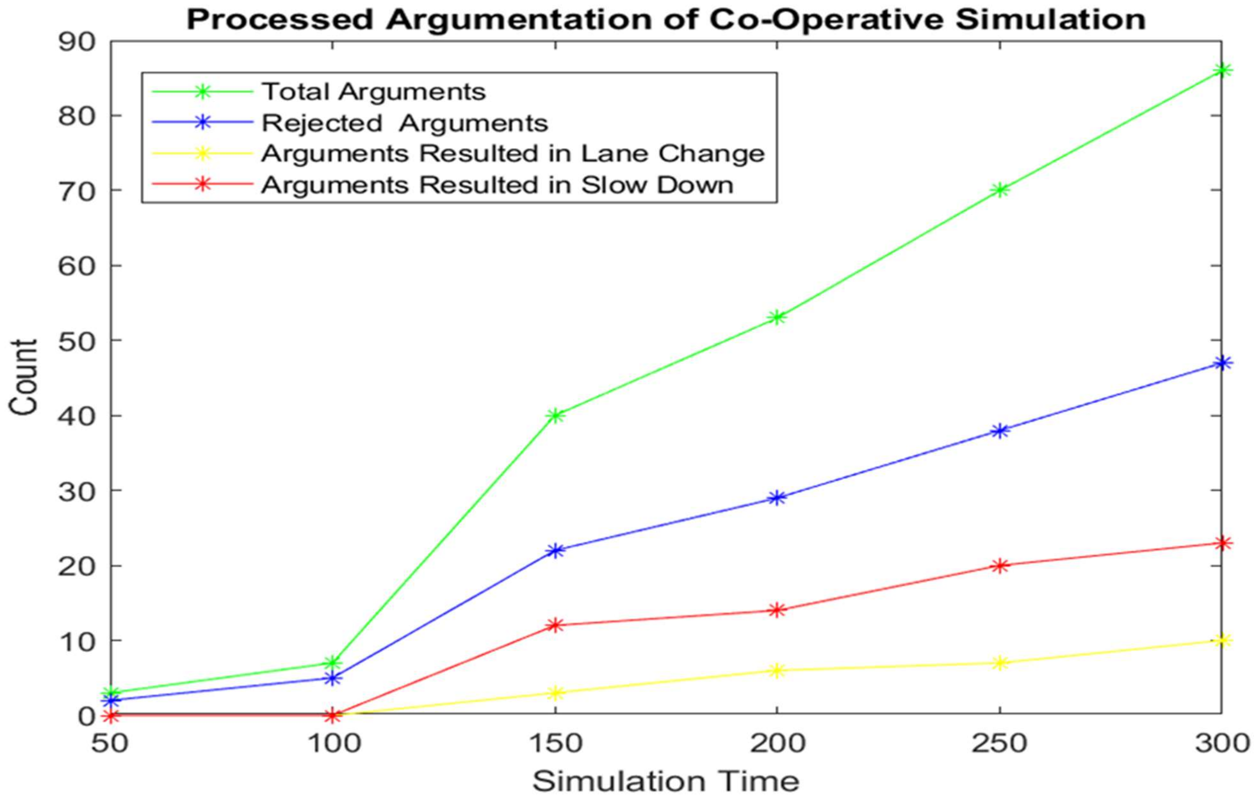

Figure 5 3 – Arguments generated and their outcomes.

The following graph (Figure 5 4) compares the Average travel time on both scenarios i.e non-Cooperative driving and Cooperative Driving. The X – axis represents the time in seconds and the Y- axis represents the Travel time in seconds. As the number of vehicles increase, there is a steady increase in the travel time due to the increase of the number of vehicles that are injected onto the testbed.

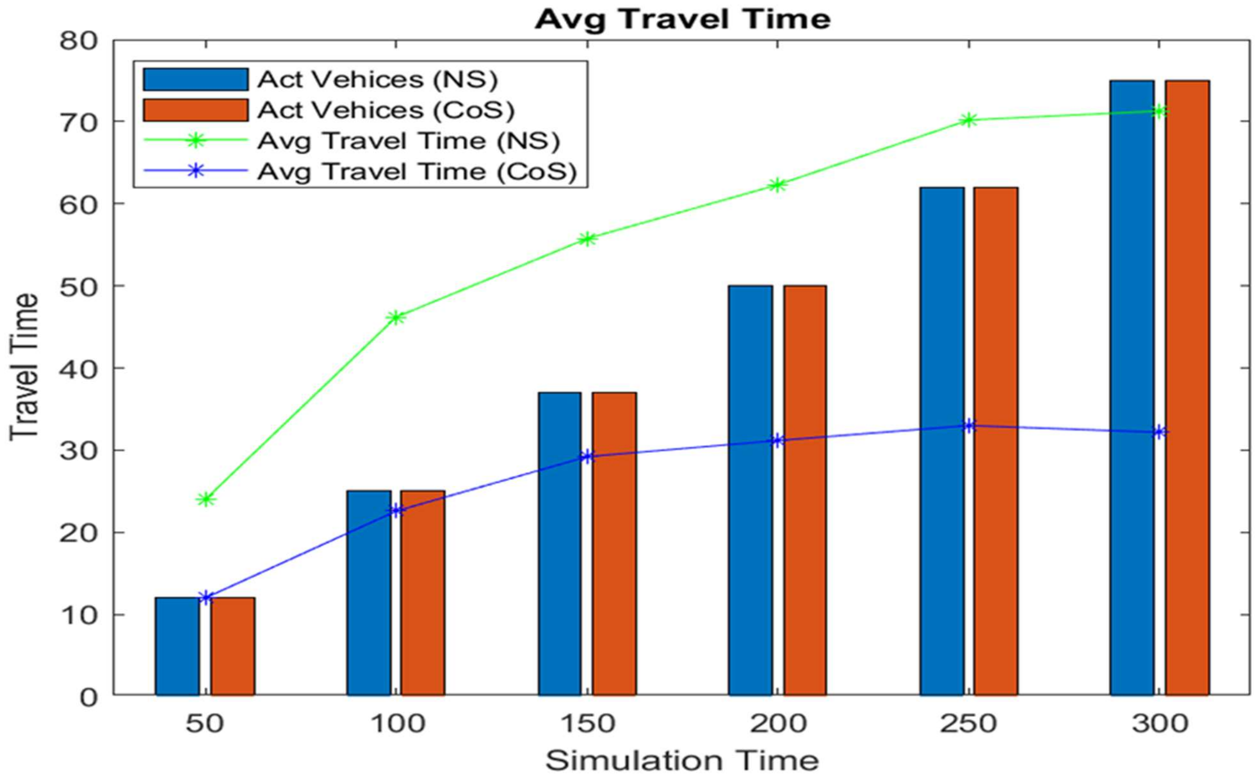

Figure 5 4 – Average Travel time comparison.

The following graph (Figure 5 5) compares the Average speed of vehicles travelling on both scenarios i.e non-Cooperative driving and Cooperative Driving. The X-axis represents the time in seconds and the Y-axis represents the speed in Mph.

The graphs (Figure 5 6 and Figure 5 7) represent the number of slowdowns and lane changes that occurred to the vehicles during the total runtime respectively. These values are compared in both Cooperative Driving and non-Cooperative Driving scenario. The X-axis representing the simulation time and the Y-axis representing the count.

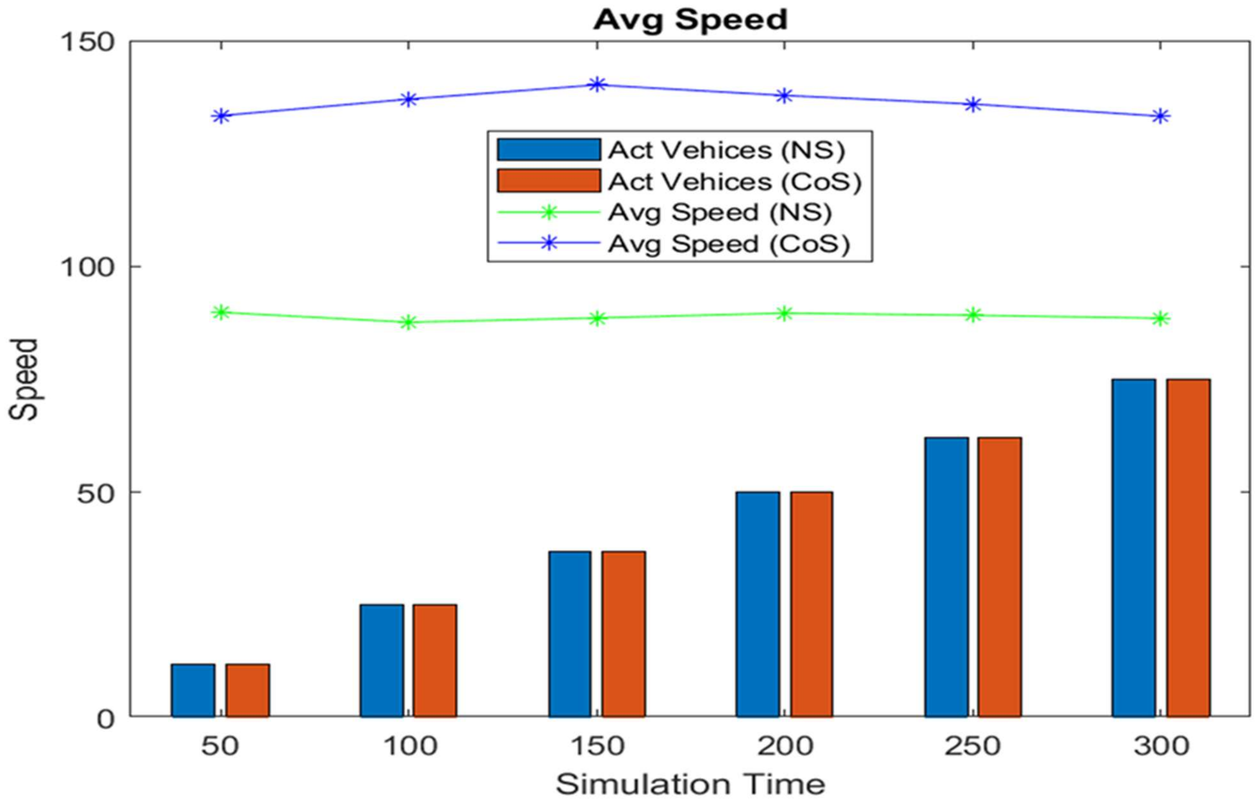

Figure 5 5 – Average Speed comparison.

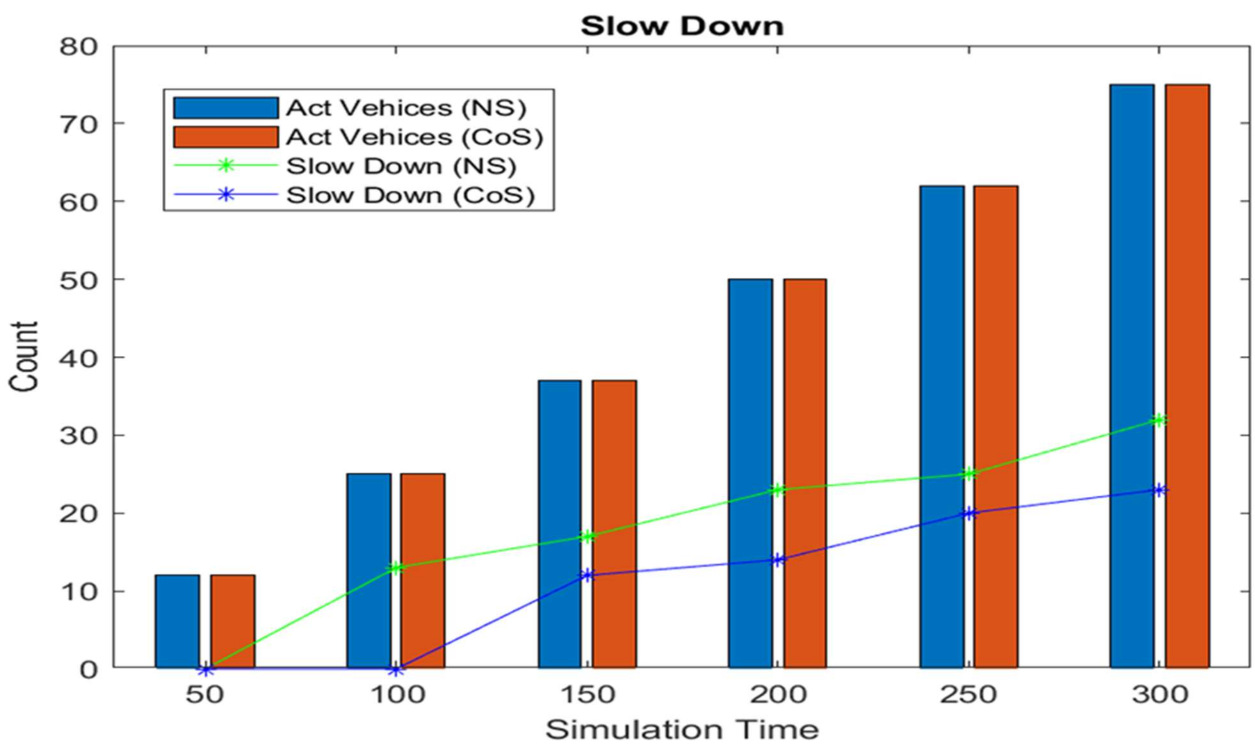

Figure 5 6 – Slowdown count.

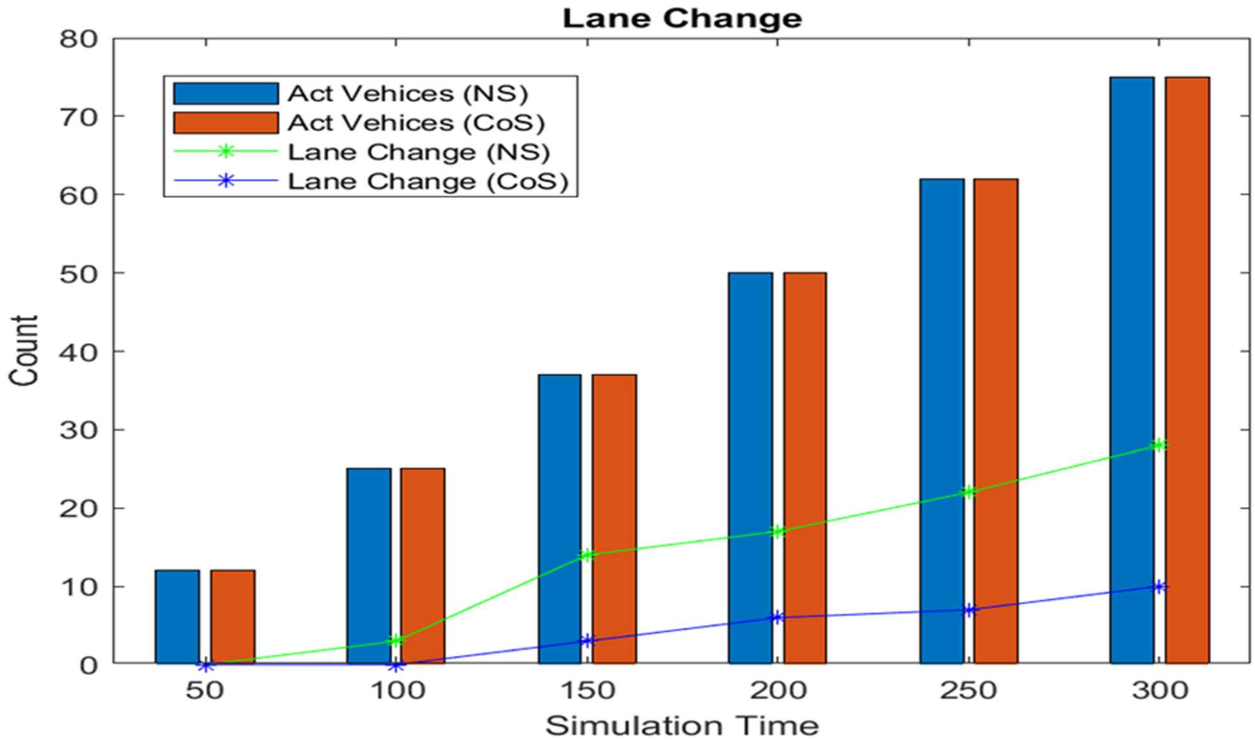

Figure 5 7 – Lane change count

The simulation run has been conducted for both scenarios cooperative driving and non-cooperative driving for 300 seconds each. The number of vehicles that injected onto the testbed is 75. The priority value range is set between 1 and 10, The top speed range is randomly distributed among vehicles between 60 and 120 mph.

In conclusion, cooperative driving algorithm held a higher average speed and lesser average travel time compared to a non-cooperative driving scenario throughout the experiment.

The average travel time was close by in the initially but as the number of vehicles and runtime kept increasing the gap was slightly increased, in the end, the cooperative simulation had less travel time. As the number of vehicles kept on increasing the travel time kept getting better. It started with a 50% less travel time and the change was slightly increased to a 60% less travel time for the cooperative driving.

The average speed remained close to constant across the whole runtime in both scenarios, with cooperative driving having a 50% higher average speed than the mo cooperative driving scenario. This trend continued from start to end of the simulation.

The average speed and average travel time proved the cooperative driving scenario to be more resourceful to accommodate the traffic and clear out the traffic in lesser time than the non-cooperative scenario.

The cooperative driving scenario enabled the vehicles in the formation of platoons and argumentation to resolve the conflicts. These were the observations in terms of platoons formed, argumentation occurred and its output.

Out of the total arguments that have been generated regarding the lane shift during the runtime. 50% of the Arguments were rejected by an agent. 30% of the Arguments resulted in the agent to slow down because of the surrounding traffic and 20% of the Arguments were successfully won and the agent was able to make the lane change.

The platoon formation also improved as time progressed. 50% of the total requests have been rejected by an agent(platoon leader). There was an increase in the number of platoon groups by the end of the runtime. There was a gradual increase in the number of groups formed as the time progressed and newer vehicles were being injected onto the testbed.

Cooperative driving proved to be more accommodating towards the vehicles and the trend seemed to be improved as the number of vehicles on the testbed are increased. This shows a promising future for cooperative driving techniques and different concepts that can be tested on this platform. Argumentation has proven to be useful in resolving the conflicts among agents, this allowed in a more uniform distribution of vehicle traffic across the road.

REFERENCES

[1] AutomotiveUI '14: Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular ApplicationsSeptember 2014 Pages 1–8https://doi.org/10.1145/2667317.2667330

[2] Bergenheim, C. ; Shladover, S. ; Coelingh, E. (2012) "Overview of platooning systems". Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)

[3] Bergenhem, Q. Huang, A. Benmimoun, T. Robinson, "Challenges of Platooning on Public Motorways," 17th World Congress on Intelligent Transport Systems, October 25-29, 2010, Busan, Korea.

[4] Y. Zhang, J. Zhao, G. Cao, "Roadcast: A popularity aware content sharing scheme in VANETs", *Proc. IEEE ICDCS*, pp. 223-230, 2009.

[5][R1]Y. Zhang and G. Cao, "V-PADA: Vehicle-Platoon-Aware Data Access in VANETs," in *IEEE Transactions on Vehicular Technology*, vol. 60, no. 5, pp. 2326-2339, Jun 2011. doi: 10.1109/TVT.2011.2148202

[6] R. Hall, C. Chin, *Vehicle sorting for platoon formation: Impacts on highway entry and throughput*, 2002.

[7] J. Zhao, Y. Zhang, G. Cao, "Data pouring and buffering on the road: a new data dissemination paradigm for vehicular ad hoc networks", *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3266-3277, Nov. 2007.

[8] X. Qian, J. Gregoire, F. Moutarde and A. De La Fortelle, "Priority-based coordination of autonomous and legacy vehicles at intersection," 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, 2014, pp. 1166-1171.

[9]Mladenovic, Milos & Abbas, Montasir. (2014). Priority-based Intersection Control Framework for Self-Driving Vehicles: Agent-based Model Development and Evaluation. 10.1109/ICCVE.2014.7297575.

[10]M. Jalalmaab, B. Fidan, S. Jeon and P. Falcone, "Model predictive path planning with time-varying safety constraints for highway autonomous driving," *2015 International Conference on Advanced Robotics (ICAR)*, Istanbul, 2015, pp. 213-217.

[11] *Chenyi Chen, Ari Seff, Alain Kornhauser, Jianxiong Xiao*; The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2722-2730

[12] M. Jalalmaab, M. Pirani, B. Fidan and S. Jeon, "Cooperative road condition estimation for an adaptive model predictive collision avoidance control strategy," *2016 IEEE Intelligent Vehicles Symposium (IV)*, Gothenburg, 2016, pp. 1072-1077.

[13] J. Ploeg *et al*., "Cooperative Automated Maneuvering at the 2016 Grand Cooperative Driving Challenge," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1213-1226, April 2018.

[14] Krishna, Madhava & Hexmoor, Henry & Chellappa, Srinivas. (2005). Reactive navigation of multiple moving agents by collaborative resolution of conflicts. J. Field Robotics. 22. 249-269. 10.1002/rob.20063.

[15] Henry Hexmoor, Kailash Yelasani, Economized Sensor Data Processing with Vehicle Platooning, World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:12, No:6, 2018

[16] Lovellette, Ellie & Hexmoor, Henry & Rodriguez, Kane. (2018). Automated Argumentation for Collaboration among Cyber-Physical System Actors at the Edge of the Internet of Things. Internet of Things. 5. 10.1016/j.iot.2018.12.002.

[17] G. Leen and D. Heffernan, "Expanding automotive electronic systems," Computer, vol. 35, pp. 88-93, 2002.

[18] J. A. Misener, S. Dickey, J. VanderWerf, and R. Sengupta, "Vehicle-Infrastructure Cooperation," in Vehicular networks: from theory to practice. Vol. 20, S. Olariu and M. A. C. Weigle, Eds., ed: Chapman & Hall/CRC, 2009

[19] H. Hartenstein, K. Laberteaux, and I. Ebrary, VANET: vehicular applications and inter-networking technologies: Wiley Online Library, 2010.

[20] Ü. Özgüner, T. Acarman, and K. Redmill, Autonomous Ground Vehicle. Norwood, MA: Artech House, 2011.

[21] "Preliminary Statement of Policy Concerning Automated Vehicles," National Highway Traffic Safety Administration2013.

[22] Eğilmez S., Martins J., Leite J. (2014) Extending Social Abstract Argumentation with Votes on Attacks. In: Black E., Modgil S., Oren N. (eds) Theory and Applications of Formal Argumentation. TAFA 2013. Lecture Notes in Computer Science, vol 8306. Springer, Berlin, Heidelberg

VITA

Graduate School
Southern Illinois University

Tharun Reddy Katukuri

tharunreddy1996@gmail.com

SR Engineering College, May 2018
Bachelor of Technology, Computer Science

Thesis Paper Title:

ARGUMENTATION AMONG SELF-DRIVING VEHICLES

Major Professor: Dr. Henry Hexmoor