Modelling and development of an Autonomous Truck Platooning System

Anas I. Abulehia
Department of Computer Science
Fachhochschule Dortmund
Dortmund, Germany
anas.abulehia001@stud.fh-dortmund.de

Mariam Jamal
Department of Computer Science
Fachhochschule Dortmund
Dortmund, Germany
mariam.jamal001@stud.fh-dortmund.de

Sergey Rogachevsky
Department of Computer Science
Fachhochschule Dortmund
Dortmund, Germany
sergey.rogachevsky001@stud.fh-dortmund.de

Mohammadreza Bagherifar
Department of Computer Science
Fachhochschule Dortmund
Dortmund, Germany
mobag002@stud.fh-dortmund.de

***Abstract*—[Anas] In this project, we have applied Mechatonics System Engineering Approach for Truck Platoon System. It includes application of CONSENS method, Modeling via SysML, behavior implementation using time automaton on UPPAAL, Cognitive Operator and CARLA simple scenario implementation. This paper demonstrates our work on Mechatronics System Engineering Course Project.**

***Index Terms*—Truck Platoon, Mechatronics System, Formal Verification, Artificial Intelligence , MUML, CONSENS**

I. Introduction

[Anas] Truck platoon is a solution for truck transportation in highways. It is a very simple idea of making trucks drive together in a section of highway lane with a predefined distance between trucks to maintain safe clearance. The trucks drive with the same velocity and coordinates between each others. This coordination contains exchanging messages and data between vehicles. This technology has great benefits such, first it reduces the air drag so better fuel efficiency. Second is reduction of congestion in highway and better utilization of roads resources. Lastly, less traffic collisions.

Source: The identical-path truck platooning problem

Fig. 1. Truck Platoon System

Several automotive companies have already implemented and offered truck platoon system like Volvo, Scania, MAN, Volkswagen and others.
Trucks platoon system is a complicated system. It contains of several independent systems; each vehicle by itself is an independent system. Collectively, they formulate a truck platoon System. Truck platoon is consider as System of Systems. System of systems create a very optimized behavior and advantages, however it imposes challenges in design integration, security and scalability.
In this paper, we demonstrate our design and approach for design. It contains CONSENS. The paper contains many parts of our works as a guide, however it does not shows every part of our work. Further documentation is provided via the GitHub repository.

II. Application Scenarios

[Mohammadreza] Application scenarios helps us to verify the correctness of our system. The essence of creating application scenarios is focusing on the most important issues that may occur in the platoon system. In the following paragraphs, we will outline a few application scenarios along with the solution for each application scenario.

Application Scenario 1 : At an intersection with a traffic light

Description : The connected vehicles reach to the traffic light, there is the possibility that some participants vehicles have already passed the traffic light and that would cause the participant vehicle to disconnect from the platoon system.

Provided Solution : We can consider two separate platoon systems(1-participant vehicles behind the traffic light, 2- participant vehicles which have passed the traffic light). After the traffic light turns Red, the first participant vehicles in the line behind the traffic light notify the leader about interruption and take the control of the second platoon system and after the traffic light turns green, the second platoon system can proceed with higher speed in order to get to the first platoon system and when they get close enough to each other, they can merge and the second platoon system can give up the control.

Application Scenario 2 : Establishing vehicle to vehicle and vehicle to infrastructure communication

Description : It is possible that the leading vehicle has an accident or loses control due to technical problems, or it may fail in communicating with the platoon participants.This may cause the whole platoon system unstable and unsafe.

Provided Solution : We should have a control room which have a control over the platoon system, leader, participant vehicles, weather and other parts of the system so in case of any kind of problems between participant vehicles or in the road, the responsible employee for the operation have to act immediately and fix the issue.

[Mariam] Application Scenario 3: Transfer of control for Leader Vehicle

Description : In case of situations where the trucks are bound to different destinations, and the leader vehicle has to leave the platoon, it shall inform and transfer the very next follower in line to take over control as the leader of the platoon.

Provided Solution : The leader vehicle shall pass the message via V2V communication to the very next follower vehicle to take over control of the platoon well in advance before it leaves the platoon. Once the acknowledgement has been received, it should hand over the control and inform all vehicles about the transfer. All other vehicles then request to join the new leader platoon. Only after completion of transfer, it should proceed to disengage from the platoon by increasing the distance.

Application Scenario 4: Follower vehicle leaves platoon

Description : In the situation of splitting of one of the trucks, it should be informed about it well in advance for the driver to take over the control.

Provided Solution : When a driver comes into the vicinity of a split point - a place where he is about to leave the platoon - the driver is informed of this upcoming event via the onboard HMI. When leaving a platoon, the CACC automatically increases the distance to the leading vehicle and the driver can leave the platoon at a highway exit or highway split. It is shown to the other drivers that the truck is about to leave the platoon and after splitting from the platoon the truck follows its individual route. The immediate next vehicle then reduces the distance after the split and CACC is again active.

[Rogachevsky]Application Scenario 5: Start of the route

Description : Each platoon before start of the route should execute a starting procedure. The procedure should define heading truck, following trucks, arrange the positions of the following trucks, configure communication channels between members of the platoon, check platoon's specific attributes, such as distance between trucks, the highest speed, minimum turn radius, etc.

Any single vehicle has the ability to be a platoon leading vehicle, another vehicle should be able to join the lead vehicle and together construct a platoon, which will consist of 2 vehicles: a leader and a follower. It can take place either before start of moving of the platoon or already while driving.

Provided Solution : When a truck tries to find a platoon, it start a special procedure "Establish connection". The trucks should be equipped with special equipment to communicate between each other. If there is no response from any platoons, the truck will set the role "Leading" to itself and depending on the urgency either will immediately move to the destination point or wait for other trucks. In case, some platoons will respond, the truck will assess the most suitable platoon and, communicating with the leading truck, try to engage it. In case, it won't be accepted by the platoon in defined time limit the truck will put the platoon in the temporary black list and invoke again the procedure "Establish connection".

III. Requirements

[Mariam] In system engineering, one of the most crucial step in modelling the overall system is to identify and define requirements in a precise manner. This task is not only driven by keeping the end-user in mind but rather all the stakeholders involved. The requirements diagram helps to concretize the goals of the system by describing in detail what features and characteristics a system must possess in order to satisfy the stakeholders. They serve as the basis for validation and verification of the system in further development phases. Some of the requirements can be seen in the figure below. It can be seen they are further classified as Demand(D) or Wish(W). Demand type requirements are mandatory to be fulfilled for the system to function correctly while Wish type demands are optional.

Requriement list			
No	Requriement description		Type (D/W)
1	Geometry		
1.1	The gap between following trucks should be not more than 3 meters.		D
1.2	Road clearance of each truck should be not less than 30cm.		W
1.3	Turn radius of each truck should be less than 15m		D
1.4	The minimum load should be more than 10 tonns		W
1.5	Each fully loaded truck should be able to make a trip on a distance at least 1500km.		W
1.6	Size of trucks:		
1.6.1		lenght not more than 20m	W
1.6.2		width not more than 2.5 meters	D
1.6.3		height not more than 3.5 m	D
1.7	The length of a platoon should be limited to 7 vehicles		D
2	Communications		
2.1	All trucks should be able to broadcast and recieve platooning information through V2V		D
2.2	The system in the ego truck shall broadcast its actual and intended acceleration via V2V to enable following vehicles to detect emergency braking events.		D
2.3	The ego truck shall be informed in case of emergency braking events of the preceding trucks in the platoon. Therefore at least the requested and actual acceleration value of the preceding platoon truck must be received and to be compared with a defined acceleration threshold value.		D

Fig. 2. Requirements - Platoon System

In our project, we have divided the requirements into different subcategories namely:

- Geometry: Here, physical details such as dimensions of the trucks, load capacity, minimum gap between trucks, size of the platoon are defined.

- Communications: All communication related requirements including the protocols to be used are defined here.
- Safety: In modelling autonomous systems, utmost importance must be given to design a safe and secure system as its malfunction can cause danger to both life and environment. Hence, these requirements cannot be compromised in any case and must be prioritized.
- Driver Satisfaction: Since, truck platoons are bound to long distance destinations, the major stakeholder is the end user; the driver in this case. Therefore, their satisfaction and requirements are taken into consideration while designing our project.
- System Interaction: The human machine interface must be easy to use and understand, simple to learn and adapt, secure and should have a fast response.

IV. Functions

[Mariam] Based on the requirements list, the functionality of the system is identified and described using the functional diagram. This diagram comprises of a hierarchical subdivision of the expected functionality of the system. A function is a relationship between the input and outputs, with a aim to fulfill a task. To begin with, the overall function is to provide mobility for trucks carrying goods. This task is then subdivided into 4 functionalities, namely

- Autonomous movement: It describes the task of the system to move autonomously and without human intervention. This is achieved by different ADAS subsystems such as ACC, AEB, LMS, GPS etc. which comprise of different sensors and actuators for its functionality.
- Platooning: Functions pertaining to searching, engaging, disengaging, formation, breaking, driving are mentioned as seen in the figure 3.

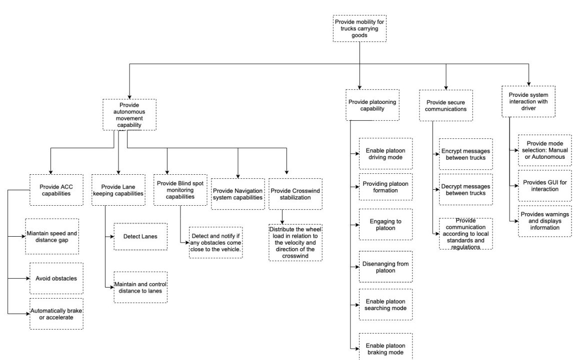


Fig. 3. Functions - Platoon System

- Secure communication: All communications for this safety critical system should be secure, for this task to be achieved, the system should be able to encrypt and decrypt the messages before sending and receiving them. Apart from this, it should also follow the standard regulations and protocols for communication. Here, the communication is ensured through wireless V2V communication protocol.
- System Interaction: This consists of functionalities related to the interaction with the system. Truck platooning system should provide with functionalities related to how the user; the driver receives and sends the information such as warnings or notifications. This can be done using a GUI for better usability. It should be noted, that our system has two modes of operation: Autonomous and Manual mode. The manual mode is meant for emergency cases where the ADAS system fails.

V. Behavior Diagram

[Mohammadreza] In order to model the behavior of the system, we have used the use case diagram and activity diagram.

A. Use Case Diagram

[Mohammadreza] In the following use case diagram, the platoon system has been described as simple as possible. As it is obvious from the diagram, we have a leader vehicle which is able to create, join, leave and disband the platoon, and also it can communicate with the operation base. And also we have the follower vehicle which can also create platoon in case as the leader vehicle leaves the platoon. The following vehicle can join and leave the platoon too. On the other hand, we have the operator who is in the control room and checks and controls all these operations.

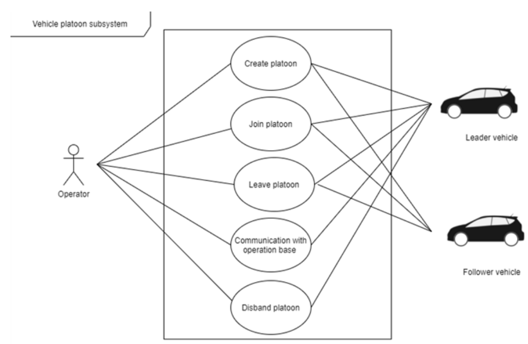


Fig. 4. Use Case Diagram

B. Activity diagram

[Mohammadreza] In the activity diagram, we can see the control flow from a start point to a finish point, showing the various decision paths that exist while the activity is being executed. leading vehicle creates the platoon and then the operator creates a safe communication line with the leader and sends the needed information such as road condition, the platoon members, current location etc. to the leading vehicle. leading vehicle at the same time adjusts its speed according to the regulation, and then the platoon system is activated. After the platoon activation, other vehicle ask for joining the platoon and if the leader vehicle accept the request, then following vehicle can collect the leader vehicle's information such as speed, distance and relative position and then adjust its gap with the nearest vehicle in 3 seconds and connect the platoon.

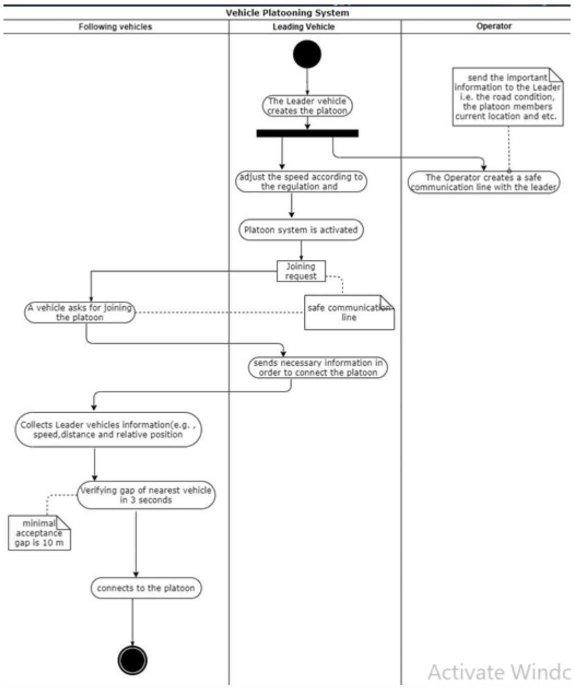


Fig. 5. Activity diagram

VI. ACTIVE STRUCTURE

[Rogachevsky] Active structure describes interaction between internal and external modules based on 3 types of flows: measurement information, information flow, energy flow. On the active structure it is shown a part of the platoon system. Each of the following trucks should be equipped at least with adaptive cruise control module and 2 subsystems -Break module and communication module. Adaptive cruise control module (ACCM) adjusts speed of the following trucks based on the information from the leading truck, GPS communication and break modules. Break module consists of following elements: brake control system, hydraulic actuator, brakes, wheels. Information flow from adaptive cruise control module transmits controlling signals to the brake control system, where based on urgency the element calculates the value of necessary pressure and sends it to the hydraulic actuator. Hydraulic actuator presses brakes to the wheels, while wheels will interact with layer of precipitations on the road. ACCM also communicates with GPS communication module to verify speed. Connected directly to the on-board computer of the GPS communication module it can select predefined satellite schema, receive current position and speed. On-board computer interacts with GPS receiver, which measures signals from GPS antenna.

figure 6.

VII. SHAPE

[Rogachevsky] One of the obligatory parts of the CONSENS methodology is to module a shape of the truck. As it can be seen from the image, a standard shape of auto was chosen. it has 4 wheels, middle height, dynamic outlines for decreasing

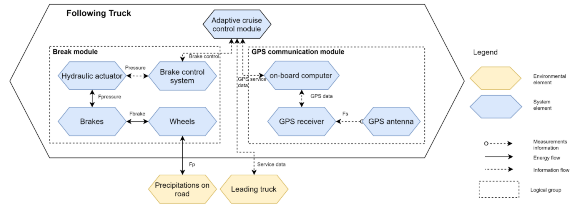


Fig. 6. Active structure

effect of air resistance, that will reduce the fuel consumption, enhance the stability, and other performance factors.

Fig. 7. Shape

VIII. SYSTEM OF OBJECTIVES

[Rogachevsky] System of objectives is developed based on the requirements to the system, it describes external, inherent and internal system objectives of the system and their interrelationships. On the figure 8, it can be seen an excerpt of modelled system of objectives. External objectives are defined by another system or by the users, inherent objectives are specified by the design purposes of the system. Internal system is constructed based on the previous two and shows the objectives at a given moment during the system's operation.

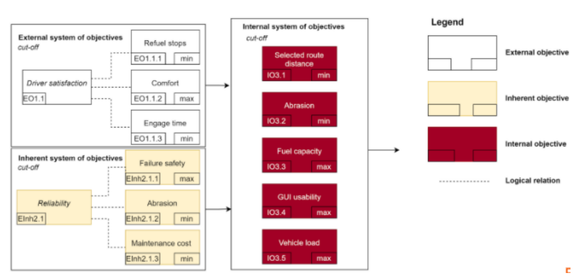


Fig. 8. System of objectives

However some internal objectives can contradict to each other. In such situations optimization of the system have to take place. The relative impact of internal objectives on each other is modelled in a table of objective correlation.

	IO3.1	IO3.2	IO3.3	IO3.4	IO3.5	IO3.6	IO3.7
IO3.1							
IO3.2	+						
IO3.3	-	-					
IO3.4	0	0	0				
IO3.5	0	-	-	0			
IO3.6	0	-	-	0	-		
IO3.7	0	0	0	-	0	0	

Fig. 9. Correlation of objectives

IX. Macro Architecture

[Mohammadreza] Here we have Relatively large-scale architecture for the platoon system in which the follower vehicle and leader vehicle are inherited from the vehicle and also a composition association relationship connects a sensor and actuator with the vehicle. And also our vehicle have a communication with roadside unit and uses the communication technology and gets service from cloud.

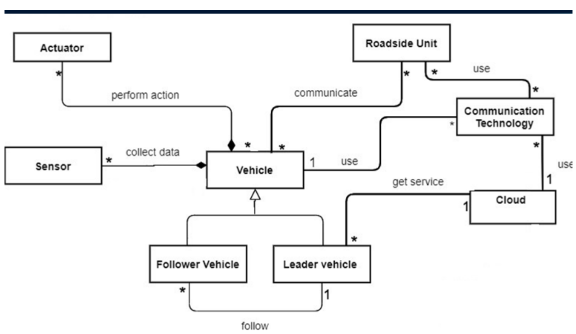


Fig. 10. Macro Architecture

X. Micro Architecture

[Mohammadreza] For the micro architecture, we chose the sensors and V2V communication to explain in detail

A. Sensor

[Mohammadreza] Sensor fusion is the ability to bring together inputs from multiple radars, Lidars and cameras to form a single model or image of the environment around a vehicle, and it can be seen in the Fig.11, it consists of in-Vehicle sensors, laser sensors, SSR/LRR Radars, cameras and V2V communication and eHorizon.

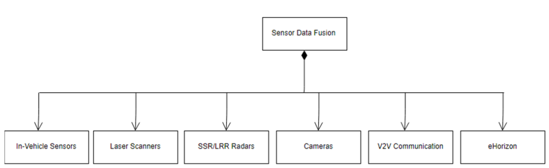


Fig. 11. Sensor

B. V2V Communication

[Mohammadreza] Fig. 12 represents the Block diagram of V2V communication-based AEB system. Vehicle-to-vehicle (V2V) communication enables vehicles to wirelessly exchange information about their speed, location, and heading. AEB systems use Lidar, radar, camera or a combination of all three to detect an impending crash in the road ahead. A 'Forward Collision Warning' alerts the driver and, if they fail to react, the system applies the brakes automatically to reduce the impact speed or avoid the crash altogether. The V2V communication-based AEB system described here is operated based on the collision detection system. During operation, the system received information about the nearby vehicles and the user vehicle through V2V communication and employed it for operation. the system receives the vehicle's positioning and direction from the positioning system and the velocity from the In-Vehicle sensor through the wireless communication module and after implementing the Kalman filter, sends the estimated position to the collision detection system and also The Collision detection system receives the lane information directly from camera and accordingly display warning vehicle and perform the collision avoidance act.

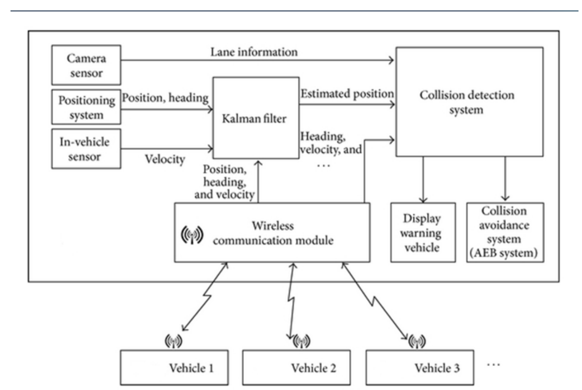


Fig. 12. V2V Communication

XI. Component Diagram

[Mohammadreza] Fig. 13 represents the component diagram for the platoon system, which breaks down the actual system under development into various high levels of functionality. The leader vehicle creates, join and communicate with the control room through the control interface and at the same time passing this information to the V2V Communication interface and the following vehicles receive the information through the V2V communication.In the other side, when the leader vehicle increase speed or brakes, sends the action to the sensor interface which can be received later by the following vehicles.

XII. Time Automaton and Verification

[Anas] Timed automata is a theory for modeling and verification of real time systems. Examples of other formalisms with the same purpose, are timed Petri Nets, timed process algebras, and real time logics. [1]. We have used timed process

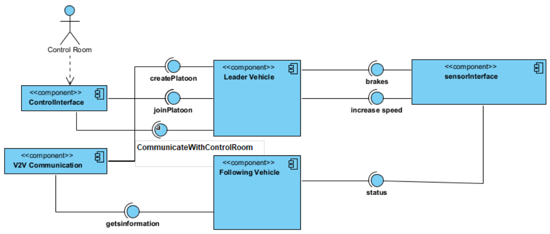
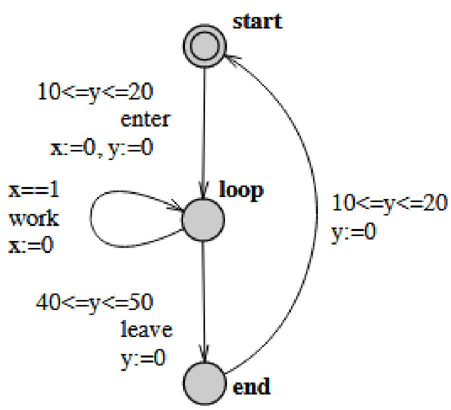


Fig. 13. Component Diagram



Source: [1]

Fig. 14. An example of a timed automata

to model a small part of our project and verify it.
Figure 14 shows a simple timed automata, it has 3 states, 4 transitions and controlled by 2 clocks x and y. Model checking techniques are used to verify these timed automaton. For that, we used UPPAAL tool for both modeling and verification. UPPAAL tool helps in both modeling and verification. In the first draft model, there is a small inconsistency that UPPAAL discovered and trace it, so that we have corrected the model. In verification parts, it shows how and why certain properties are not satisfied leading to add changes to the model. Our model simply contains 2 time automata a leading vehicle and a follower vehicle.

A. Leading Vehicle

Leading Vehicles automata has 4 states. Start and Manual driving and their transaction realize the turning on and off the vehicle. When the vehicle is running, it is in the manual driving model. At this state, a vehicle can move to the Platoon Mode. In the platoon mode state, the leading vehicle receives requests to join the platoon, the driver accepts or rejects, if it accepts the request, it adds the number of vehicles in the platoon. All members of platoon will receive data from the leading vehicle every 2 time units. There is also a possibility for the leading vehicle to 'kill the platoon' and return to manual driving mode. All of these transitions and states are show in figure 15.

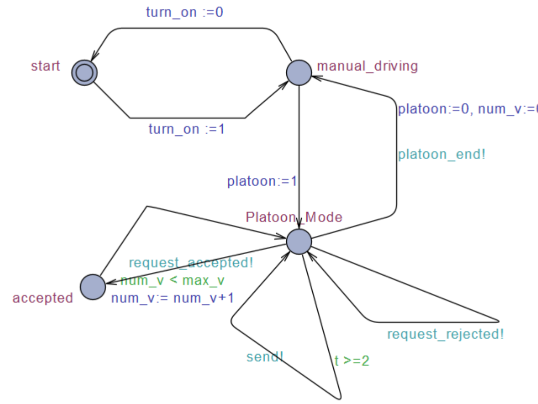


Fig. 15. Leading Vehicle Automata

In our platoon implementation, there is only one leading vehicle; there is no possibility to create multiple instances of leading vehicle object.

B. Following Vehicle

The following vehicle Automata has 4 states too, but different logic in transitions and states. The start and manual driving states represent switch on and off the vehicle. In manual driving, an transaction is enabled only if there is already a platoon; Leading vehicle is responsible for altering this variable. If there is a platoon, the state of the system is going to be in temporary state. In this state it waits for approval of the joining. In case of rejection it goes to manual driving mode where as in case of accepting, it goes to listen state, where it receives data from the leading vehicle.

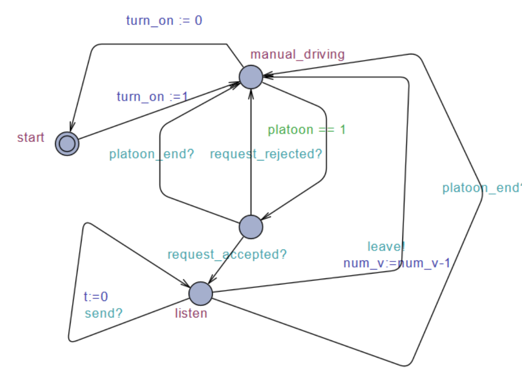


Fig. 16. Following Vehicle Automata

We have created multiple instances of following platoon object.

C. Verification

Verification is the process of make sure that we have implemented the system properly. 'Doing things right' compared with validation 'Doing the right thing'. UPPAAL software tool gives the capability to check some properties. Some verified properties are :

- it is possible for a vehicle to be in a platoon; listen state and receive data form leading vehicle. The syntax is $E <> v1.listen$
- The number of vehicles in the platoon less than maximum possible number as specified in requirements. The UPPAAL synatx is $A[]numv <= maxv$.
- A very important property is when a follower vehicle in the platoon the leading vehicle is always not in manual or start mode. This property is safety requirement. The UPPAAL synatx is $E\ <>$ v1.listen and not vleading.manualdriving and not vleading.start
- A property we seek in any system is deadlock free. There is no deadlock. The UPPAAL synatx $A[]$ not deadlock.

All of previously mentioned properties are shown in Figure

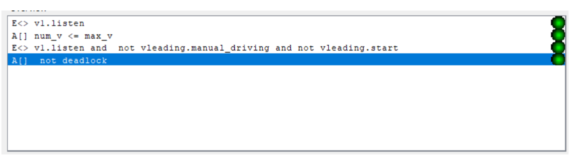


Fig. 17. UPPAAL Verification Window Shows Checked Properties

[Mariam] In order to make sure that the system is robust in case of the loss of communication and the required distance is maintained, the above mentioned UPPAAL models were extended as follows:

D. Leading Vehicle

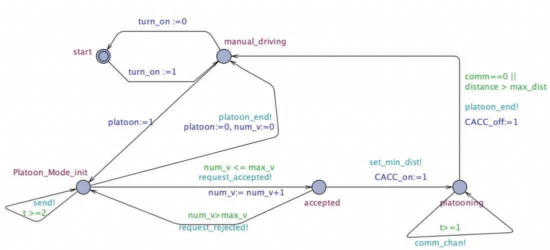


Fig. 18. Leading Vehicle

For the leading vehicle, after entering the platoon mode, the system will be initialized for receiving requests. The requests can either be accepted or rejected based on the already existing number of trucks in the platoon. If it is less than the maximum number of platoon(7) then, the request is accepted and it moves onto the accept state, where the minimum gap distance to be maintained and the speed for CACC is communicated as to the following vehicles. After the setup, it enters platooning mode, where each second a boolean is sent and received back to establish that the connection is not lost. If there is no connection, the communication boolean is set to 0. This acts as a trigger, and the system enter the manual driving mode by turning off the CACC and updating the existing number of trucks in the platoon.

E. Following Vehicle

In the follower vehicle automata, after setting the minimum gap distance and entering the platooning state, it remains in this state unless the communication is lost or if the distance between the vehicles is more than the maximum distance set. In such case, the platoon mode ends and it enters into the manual driving mode.

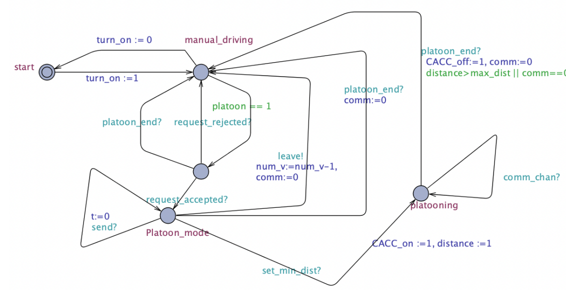


Fig. 19. Following Vehicle

XIII. Cognitive operator

[Rogachevsky] Cognitive operator is a part of operator controller module. The purpose of this module is to adjust algorithms of the reflective operator. Cognitive operator commonly consists of such modules as:

- Knowledge base
- Situation analysis
- System of objectives
- Adaptive system's behaviour

The element "Knowledge base" is used for storing information, which is necessary for functioning of the system. The data from this element can be classified or requested by the block "situation analysis", which may also get information from the reflective operator. The purpose to use the element "System of objectives" is to project and optimize system performance, to weight objectives and to simulate models. The module "Adaptive system's behaviour" is used to transmit configurations or defined parameters, and to evaluate information. The processing of information through these cognitive operator's elements is called "cognitive loop". For implementation of this module we chose supervised machine learning algorithm "Random forest". Among important advantages of this algorithm for the truck platooning system are the following:

- it can handle missing values
- it doesn't overfit the model
- it can handle large datasets with high dimensionality

As it can be seen from the figure below, for using the algorithm it is required to have a well prepared dataset to train

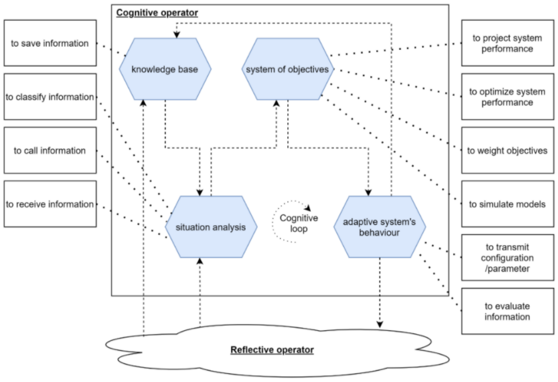


Fig. 20. Cognitive operator

the machine learning model. The correctness and consistancy of the dataset greately impacts on the prediction accuracy. After getting a machine learning model, which meets the requirements of the system it can be used later to predict outcomes based on current situation conditions.

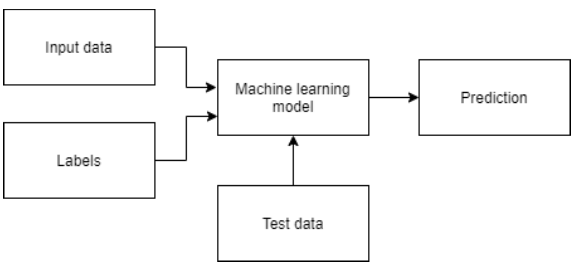


Fig. 21. Machine learning algorithm

For definition of the possible role of a truck in platoons such attributes were chosen:

- Assigned_Role_Id - role id of the truck. It may have such values as: searching, leading, following
- Route_Destination_Id - id of the truck's destination city
- Distance - distance to the destination city from current position
- Platoon_Destination_Id - id of the destination of already existing platoon
- Platoon_Number - the identification number of platoon
- Max_Platoon_Size - the maximum size of the defined platoon
- Platoon_Size - current amount of trucks in the defined platoon
- Distance_to_platoon - distance in km from the current position of the truck to the position of platoon
- Weather_condition - assessment of the weather according to the information from platoon
- Truck_load - current load of truck
- Truck_load_limit - truck load limit for platoon
- Timeout - reflects exceeded the predefined time limit

Although the random forest algorithm has low impact of configuration parameters on its data processing, it showed high performance for our dataset, which consisted of 50 records. The overall accuracy reached 88%, what can be assessed as good enough for predicting the possible role in a platoon for the truck.

```
[[ 1  0  0]
 [ 1  1  0]
 [ 1  0 14]]
              precision    recall  f1-score   support

           0       0.33      1.00      0.50         1
           1       1.00      0.50      0.67         2
           2       1.00      0.93      0.97        15

    accuracy                           0.89        18
   macro avg       0.78      0.81      0.71        18
weighted avg       0.96      0.89      0.91        18

0.8888888888888888
```

Fig. 22. Machine learning algorithm outcome

XIV. Carla Scenario Implementation

[Anas] Carla is an open-source simulator for autonomous driving research. CARLA has been developed from the ground up to support development, training, and validation of autonomous urban driving systems [2]. For controlling Carla objects, several Python scripts run on command window. We have used Carla for the implementation of a simple case, this case is driving 2 vehicles in a straight line by sharing the control signals between vehicles. All vehicles are Tesla model truck, the environment is simplified as well; no other vehicles in the road.
There were 2 scripts. The first one is environment setup. It makes a communication with the sever 'unreal engine 5' then it loads a certain map, lastly it removes all objects form the map like vehicles, pedestrians and etc. The second script is platoon script, it load 3 points and create 3 Tesla trucks with 10 meter distance gap. Then all vehicles receive control signals profile for brake and throttle signals. Figure23 is a snippet of a recorded video of our Carla Model.

XV. Conclusion

[Mohammadreza] An automated vehicle platoon and its configuration, and control systems, overall system behavior, and the possibility of platoon implementation have been introduced. The leading vehicle can lead up to 7 vehicles, and the various sensors has been implemented to reduce the errors which may occur in the road. Finally, the automated platoon of 3 vehicles drove on a test track on CARLA has been implemented to check the viability of the system. The system seems stable for the 7 follower vehicles, and we hope that we can improve it in future and platoon unlimited vehicles on the highway.

References

[1] Timed Automata: Semantics, Algorithms and Tools Johan Bengtsson and Wang Y
[2] CARLA: An Open Urban Driving Simulator Alexey Dosovitskiy etl

Fig. 23. Carla Implementation Snippet

[3] Cho, Hong and Kim, Gyoung-Eun and Kim, Byeong-Woo. (2014). Usability Analysis of Collision Avoidance System in Vehicle-to-Vehicle Communication Environment. Journal of Applied Mathematics. 2014. 1-10. 10.1155/2014/951214.

[4] Coda, A., Konstantinopoulou, L, 2018. Functional specification for white-label truck. D2.4 of H2020 project ENSEMBLE(https://platooningensemble.eu/library).

[5] Vissers, J., 2018. Platooning use-cases, scenario definition and Platooning Levels. D2.2 of H2020 project ENSEMBLE (https://platooningensemble.eu/library).

[6] Willemsen, D., 2018. Requirements Review from EU projects. D2.1 of H2020 project ENSEMBLE (https://platooningensemble.eu/library).

[7] Nyßen, A., Mühlbrandt, M., 2013. Entwicklerdokumentation für das CONSENS SysML 1.3 Profil.