Hatab, Mehdi

Maneuver management of a platoon in highway

Bachelor's Thesis

to achieve the university degree of

Bachleor of Science

Bachelor's degree programme: Electrical engineering

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. BSc. Renzler, Tobias

Institute of Automation and Control
Head: Univ.-Prof. Dipl-Ing. Dr.techn. Horn, Martin

Graz, January 2021

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online:
https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present bachlor's thesis.

Date

Signature

Abstract

Vehicles require to communicate and share their information such as velocity and location to form a platoon. In the joining maneuver problem, a joiner must know where will be joined in a platoon. Will a vehicle joins from the tail or the platoon's side or maybe become the new leader. In the leaving assignment, the followers require to create a space, which is enough to allow a vehicle to change its lane. This thesis concerts on implementing an algorithm of joining/leaving maneuver and information send to the leader. The validity of the approach is proven using a simulation showing either that the maneuver can successfully be performed or not.

Contents

List of Figures

1. Introduction

In a platoon system, several vehicles follow one leader vehicle in a vehicle platoon system and run in a line to maintain a constant velocity. Also, each vehicle maintains a safe distance from its preceding vehicle. The inter-vehicle distance in one lane can provide higher roadway throughput and better traffic flow control. It can also help to reduce energy consumption by avoiding unnecessary changes in acceleration. Despite these advantages, the shorter inter-vehicle distance brings about a safety issue, which requires the vehicles in a platoon to always maintain the safety inter-vehicle distance. This requirement needs platoon maintenance for vehicle dynamics (i.e., vehicle joins and departures). When a vehicle enters a platoon, its succeeding vehicles need to decrease their velocities to leave enough space for the entering vehicle. Similarly, when a vehicle leaves the platoon, the vehicles' velocities need to be increased to maintain the platoon. This thesis will concert on such a problem: how can a vehicle join or leave a platoon.

This thesis will concert on platoon formation, especially the joining and leaving assignment.

1.1. Motivation

To form a platoon and obtain the platoon's benefits as increased lane capacity due to shorter inter-vehicle distance and reduce fuel consumption since all vehicles dose not accelerate nor decelerate. Any vehicle should able to join a platoon. The joining maneuver must be done without affecting the other vehicles, i.e., a vehicle joins a platoon safely—the same problem in the leaving assignment. A leaver wishes to leave a platoon when it reaches its destination. Therefore, leaving a vehicle should be done without affecting the traffic and safely splitting from the platoon.

1.2. Contribution

The critical result is that any vehicle can join from anywhere in the platoon. With this, a platoon can be formed. The position of the joiner is not restricting from where it is located toward the platoon. With the help of LQR, the longitude distance will be controlled. By modifying the joiner's tracking reference and the behind one, a vehicle can merge safely when both vehicles reach the modifying or even surpass it. It can be formed a platoon since there is a vehicle that attends to join a platoon. The same way of leaving any vehicle can leave a platoon, and the approach is the same in joining maneuver by modifying the tracking reference.

1.3. Outlines

Chapter 3 will discuss the algorithm for the platooning maneuver and how the joining and leaving be done. In chapter 4 we will simulate the algorithm using MATLAB/SIMULINK and SUMO and observe the result. In the last chapter 5, we will summarize the thesis and discuss any improvement.

2. Background

2.1. Platooning

A platoon is a group reacting in a coordinated way. Establishing vehicle platoons can result in increased road capacity, better efficiency due to less air resistance, less consumption of fuel, reduced congestion, less air pollution and better comfort for passengers and driver. In a platoon of vehicles there is one vehicle that leads the platoon while all other vehicles follow it with the same velocity within a shorter inter-vehicle distance [**bergenhem2012overview**]. In addition, platooning vehicles are expected to cooperate with each other to maintain stability of vehicle string: string stability is a measure of how errors are circulated from one vehicle to another in a platoon. These errors can be distance or velocity errors. In a string stable platoon, these errors have to decrease while moving down the platoon from vehicle to vehicle [**swaroop1996string**]

2.2. Network

The platooning application requires a communication application to allow vehicles share their information. The commonly application is V2V. V2V (**V**ehicle-**2**-**V**ehicle) communication is based on WLAN-Standard IEEE 802.11p. With V2V communication vehicles can sharing their state information e.g., velocity and position among vehicles in the platoon. The shared informations are used to control the vehicles platoon. Using V2V implies that data can be sent directly from the source rather being indirectly measured locally with sensors. Detecting platoon movements via only local sensors is prone to lag and to accumulate errors. This is because local sensor measurements are only based on the adjacent vehicle. [**bergenhem2012overview**]

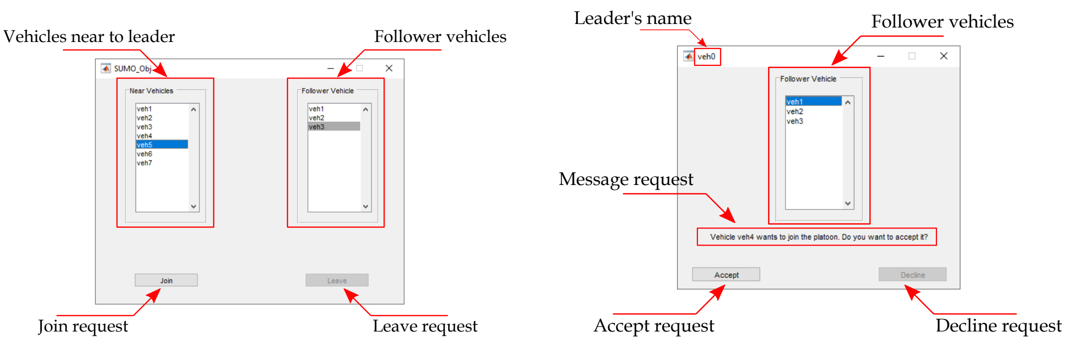
2.3. Inter-vehicle distance

To maintain the inter-vehicle distance a controller is requiring. One of those controller is CACC. **C**ooperative **A**daptive **C**ruise **C**ontrol (CACC) is an extended version of a standard **A**daptive **C**ruise **C**ontrol (ACC) that exploits wireless communication to reduce the inter-vehicle gap. The ACC, instead, relies only on a radar or a lidar. the inter-vehicle distance for an ACC is a constant time headway, i.e., the actual distance is computed as the product of the time headway and the current speed, so the distance increases proportionally with speed. On the other hand The CACC longitudinally controls the vehicle by computing a desired acceleration that should be applied to maintain a certain inter-vehicle distance. To take its decision the control system obtains data from sensors and via wireless communication. This data can include distance to the front vehicle, position, speed, or acceleration of other vehicles. [**ploeg2011design**]

2.4. Software

Programs, that being used in this thesis, are the following:

1. Microscopic traffic simulator "**S**imulation of **U**rban **MO**bility" SUMO [**SUMO1**]. SUMO is designed for generating real driving behavior on highways or urban traffic and to provide a graphical user interface (GUI) to observe the motion of vehicles. It allows to consider different road types, traffic lights, intersections, slopes, additional vehicles (e.g. cars, trucks, bicycles, trams, etc.) and pedestrians.
2. MATLAB/SIMULINK [**MATLAB**] for managing the join/leave request also for the control vehicles. The join/leave request will be managed using 2 GUIs as is shown in figure 2.1. GUI in 2.1a will be used for sending join and/or leave requests and in 2.1b will be used for the acceptance/rejection the requests.



(a) GUI to send join/leave request.

(b) GUI to accept/decline request.

Figure 2.1.: GUI that had been using to manage the request

3. Maneuver algorithm

In this chapter, we will discuss the algorithm, which has been used in this thesis. In the beginning, we will discuss joining a vehicle to a platoon, with any case, i.e., join a vehicle from the rear, side, and front of the platoon. Further, we will discuss leaving a vehicle from a platoon again from rear and side of the platoon.

3.1. Joining maneuver

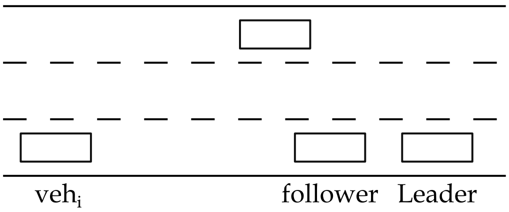
The Join maneuver is triggered when a vehicle wants to join a platoon. The simplest solution of all is to make vehicles join the tail of the platoon. The other one is to make vehicles join the side of the platoon. In this case, vehicles open a gap that allows the joining vehicle to merge to the correct lane. This solution is more challenging since it requires more coordination between several vehicles. When a vehicle sends a join request to a leader of a platoon, it will send as well its position, lane, ID, etc., in other word its state information:

$$\mathbf{Data} = \begin{cases} ID \\ Position \\ Velocity \\ Lane \\ Signal \\ Leader \\ InPlatoon \\ Angle \end{cases} .$$

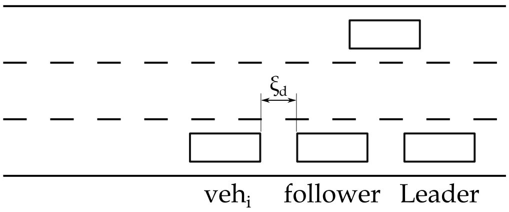
ID represents to the vehicle's identity, *Position* represents to the distance reference to the leader, *Velocity* represents to the current velocity, *Lane* represents to the current lane, *Signal* represents to a request which is sending to the leader, *Leader* represents if the vehicle is a leader or not, *InPlatoon* represents if the vehicle is a part of the platoon and *Angle* represents to the angle reference to the leader.

3.1.1. Joining from rear and side

Now, let discuss about joining, where the joiner locates behind the leader. After receiving a join request from veh_i, the leader can accept or reject the offer. For the case, which the leader denies the request, veh_i receives a negative response. On the other hand, if the joiner (veh_i) obtains a positive response from the leader, the leader starts to send information to veh_i. Therefore, it starts the merging phase.



(a)



(b)

Figure 3.1.: Joining maneuver from the rear of the platoon.

After the leader sends an acceptance to allow veh_i joins the platoon, veh_i assign the parameter *InPlatoon* equal to two. This means a vehicle is in the merging phase. Furthermore, veh_i return the parameter *Signal* from one to zero, since it receives a response from the leader.

For each iterate, the leader obtains the data from the follower and the joiner. The leader vehicle split the data, which had been received from the vehicles into two parts, depending on the parameters *Signal*. The first part contains the vehicle sending a join/leave request. In other word, the Parameter *Signal* is not equal to zero and save it in *requests*. $Signal = 1$ means the vehicle wants to join the platoon, and $Signal = 2$ means the vehicle wants to leave the platoon. On the other hand, any vehicle with the parameter $Signal = 0$ saves in *followe_date*, i.e., those vehicles already in the platoon or even in the merging phase. The leader works on the split data-parallel.

The first part of the data received from the other vehicles is used to manage the request. The main approach of managing the requests when there are too many

Data: Recieived data from the vehicles
Result: Datas split into two parts
Split the vehicle's data into two parts, depending on the parameters *Signal*;
for *All vehicles* **do**
  if *The parameter "Signal" is not equal to zero* **then**
    Put this vehicle's information into the variables *requests*;
  else
    Put this vehicle's information into the variable *follower_date*;
  end
end

Algorithm 1: Split the vehicles data.

requests simultaneously is to use semaphores, and the vehicle closest to the leader is served first. The following points explain the way:

1. If the leader receives only one request from veh_i, the leader blocks other requests until it sends a response to veh_i.
2. If the leader receives more than one request simultaneously, it will receive the nearest vehicle toward the leader and block the other request.

After the leader sends a response to veh_i, either positive or negative acknowledgment veh_i set the parameter $Signal = 0$. If veh_i wants to join the platoon, it will set the parameter $InPlatoon = 2$. That is, veh_i is in the merging phase to join the platoon. On the other hand, if veh_i wants to leave the platoon, it will set the parameter $InPlatoon = 3$. That is, veh_i is in the leaving phase to leave the platoon.

The second part of the divided data is where the leader manages the followers' velocity and lane. In the beginning, the leader sorts the data depending on the distance and saves the vehicles' sorted and unsorted index. The leader then calculates the difference in distance and velocity between each vehicle with the sorted index. Furthermore, The leader checks the parameter *InPlatoon* for each vehicle. As was discussed, $InPlatoon = 1$ means the vehicle is a follower, $InPlatoon = 2$ means the vehicle is in merging phase to the platoon, $InPlatoon = 3$ means the vehicle is in leaving phase. For the case joining from the side of the platoon, the leader checks which vehicle is behind the joiner (veh_{i+1} behind veh_i as is illustrated in figure 3.2b). The leader sends to veh_{i+1} to slow down and to create a gap, which is enough to let veh_i merges with the platoon. The method, where veh_{i+1} is slowing down, is by modifying the controller's reference. Which means, the reference tracking of veh_{i+1} modify from ξ_d to ξ_{og}, where the predecessor remains the same (veh_{i-1}). On the other hand, the leader vehicle also controls veh_i using the same predecessor of veh_{i+1}. In this case, the controller tracks the longitude distance between veh_i and veh_{i-1} until it reach ξ_{jg}.

Since the value of ξ_{og} can not be chosen randomly, due to the case the joiner's length may longer than ξ_{og}. Therefore ξ_{og} is chosen as

$$\xi_{og} = 2\xi_{jg} + L_i. \tag{3.1}$$

where L_i is the length of veh_i. When distance between veh_i and veh_{i-1} as well as between veh_{i+1} and veh_i is larger or equal than ξ_{jg}, veh_i receives a signal from the leader to change its lane. After veh_i changes its lane, the desired distance changes again from ξ_{og} to ξ_d for veh_{i+1} as well as for veh_i from ξ_{jg} to ξ_d. The predecessor of $vehi+1$ will become veh_i and the predecessor of veh_i is veh_{i-1}. Figure 3.2 illustrates the joining maneuver from the side of the vehicle platooning.

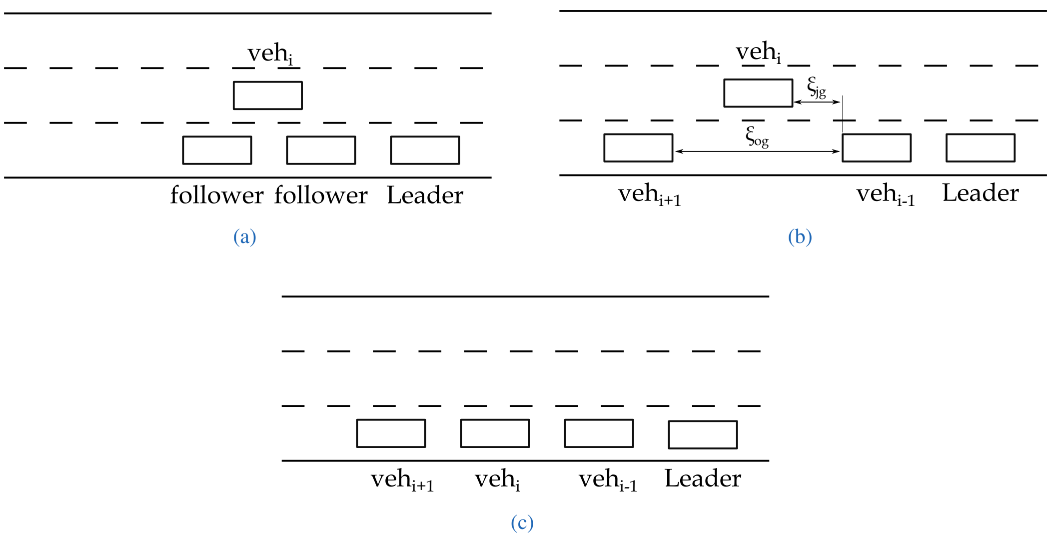


Figure 3.2.: Joining maneuver from the side of the vehicle platooning.

Data: Followers data
Result: Error calculation between vehicles.
Sort the data depending on the position of the vehicles refers to the leader;
Put the original index in m;
Put the sorted index in $\tilde{m}$;
for $i = \tilde{m}$ **do**
  if *i is equal to 1* **then**
    $e_d(\tilde{m}(i)) := d(\tilde{m}(i));$
    $e_v(\tilde{m}(i)) := v(\tilde{m}(i)) - v_{Leader};$
  else
    $e_d(\tilde{m}(i)) := d(\tilde{m}(i)) - d(\tilde{m}(i-1));$
    $e_v(\tilde{m}(i)) := v(\tilde{m}(i)) - v(\tilde{m}(i-1));$
  end
end

Algorithm 2: Error calculation.

3.1.2. Joining from front

Until now, it had been covered how a vehicle can join into a platoon from the end and the side of the platoon. The last scenario is if a vehicle (veh_i) sends a join request where is located ahead of a platoon. In this case, if the leader accepts the request, it becomes a follower and starts to sends its information to the new leader (veh_i). The followers also send information to the new leader. Figure 3.3 illustrates a joining vehicle from the front of a platoon

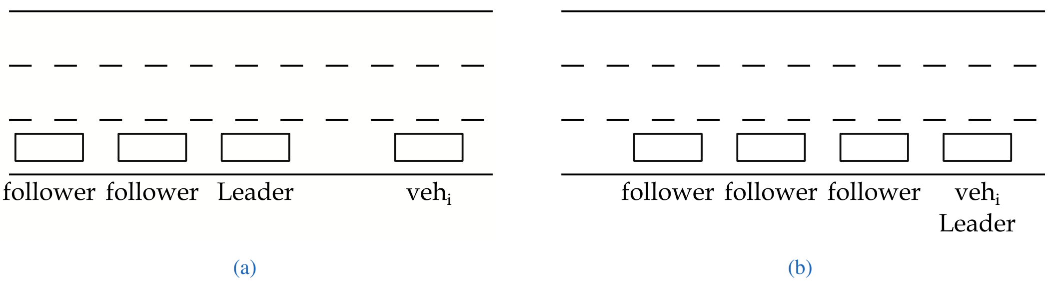


Figure 3.3.: Joining from the front of the platoon.

Data: Followers data
Result: Modifying the reference for a joining maneuver
Find a vehicle in the joining maneuver;
Put n equal to the index, where $InPlatoon$ is equal to 2;
for $i = n$ **do**
    Find the index of $\tilde{m}$ where is equal to i and put it in idx;
    if *The i-th vehicle is not in the platoon's lane* ***and*** *is the last vehicle* **then**
        if $e_d(idx) + \xi_{og} \leq 1$ **then**
            Send a message to this vehicle to change its lane;
        else
            $d_{desired}(idx) := -\xi_{og}$;
        end
    else
        if *The i-th vehicle is not in the platoon's lane* ***and*** *is not the last vehicle* **then**
            if *The i-th vehicle is between leader and first follower* **then**
                $e_d(idx) := d(idx) + length(idx)$;
                $e_v(idx) := v(idx) - v_{Leader}$;
            else
                $e_d(idx+1) := d(idx+1) - d(idx-1) + length(idx)$;
                $e_v(idx+1) := v(idx+1) - v(idx-1)$;
            end
            if $e_d(idx+1) + \xi_{og} \leq 1$ ***and*** $e_d(idx) + d_{gap} \leq 1$ **then**
                Send a message to this vehicle to change its lane;
            end
            if $|e_v(idx) - e_v(idx+1)| \leq v_{gap}$ **then**
                $d_{desired}(idx) := -\xi_{jg}$;
                $d_{desired}(idx+1) := -\xi_{og}$;
            end
        end
    end
end

Algorithm 3: Joining maneuver.

3.2. Leave maneuver

In the previously section 3.1 it was covered the joining maneuver from different location. In this sections it will cover how a vehicle can leave a platoon from the rear and from the side of a platoon as well as the leader's leaving maneuver.

The beginning of the leave maneuver is a vehicle (veh_i) send a request to the leader vehicle to leave the platoon. Which means, the parameter $Signal$ put to equal two. If the leader rejects the leave request from veh_i, the parameter $Signal$ reset to zero and veh_i remains a follower of the platoon. Otherwise, if the leader vehicle accepts the request, the parameter $Signal$ will rests but $InPlatoon$ sets to 3 rather than 1. In other words, veh_i will be in the leaving phase.

After veh_i acquires a permission to leave the platoon from the leader, it begin to decelerate. Thus, veh_i creates a gap to change lane safely. The approach of slowing down veh_i is the same approach using in joining maneuver from the side. This approach is by modifying the tracking reference of veh_i from ξ_d to ξ_{lg}. The changing of the tracking reference is also valid for veh_{i+1}, if veh_i is not the last vehicle of the platoon. When the error distance between veh_i and veh_{i-1} as well as between veh_i and veh_{i+1} reach the new reference ξ_{lg} or even surpass it, the leader vehicle notify veh_i to change its lane safely. The time that veh_i change the lane safely, the tracking reference of veh_{i+1} swap to ξ_d. Therefore, veh_{i+1} begins to accelerate to close the gap that left due to the leaving of veh_i. Figure 3.4 illustrates the behavior of a leaving a vehicle from the side of a platoon and figure 3.5 illustrates the behavior of a leaving a vehicle from the rear of a platoon.

For the case of leader's leaving maneuver the principle is the same. However, in this case the first follower is modified its reference rather than leader its speed to create gap. When the first follower converge to the modified reference ξ_{lg}, the leader change its lane safely. Moreover, after the leader change its lane safely, it sends all platoon's data to first follower and all followers send their information to the new leader. Figure 3.6 illustrates the behavior of leader's leaving and the first follower take the roll of the platoon.

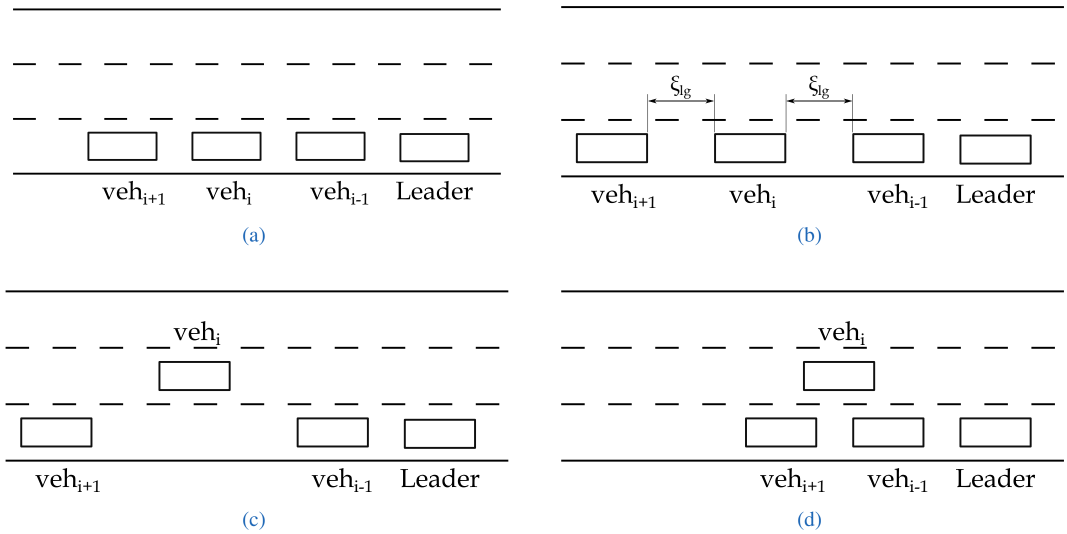


Figure 3.4.: Leaving maneuver from the side of the platoon.

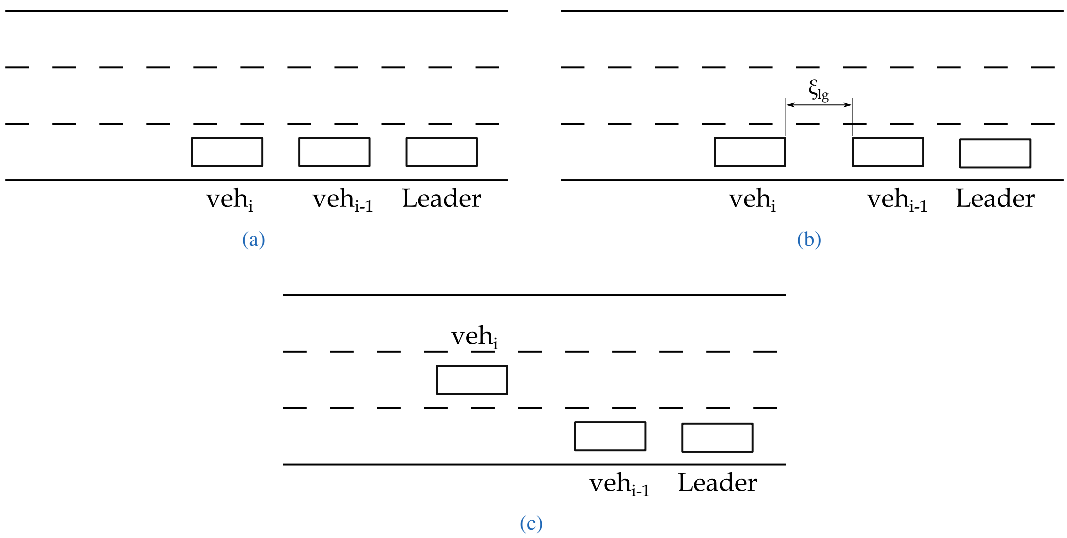
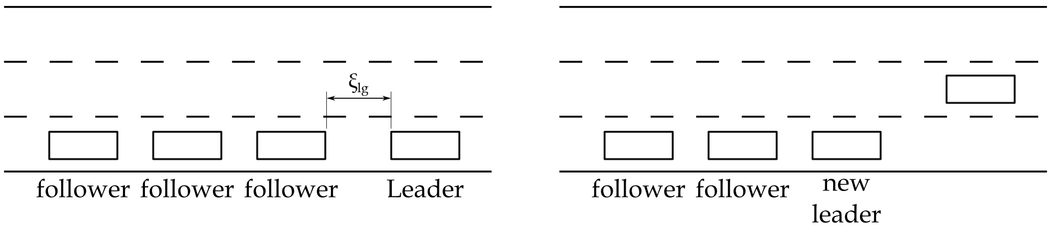


Figure 3.5.: Leaving maneuver from the rear of the platoon.



(a) (b)

Figure 3.6.: Leaving maneuver of the platoon leader.

Data: Followers data
Result: Modifying the reference for a leaving maneuver
Find a vehicle in the leaving maneuver;
Put n equal to the index, where $In\_Platoon$ is equal to 3;
for $i = n$ **do**
  Find the index of $\tilde{m}$ where is equal to i and put it in idx;
  if *The i-th vehicle is in the platoon's lane **and** is the last vehicle* **then**
    if $e_d(idx) + \xi_{lg} \leq 1$ **then**
      Send a message to this vehicle to change its lane;
    else
      $d_{desired}(idx) := -\xi_{lg}$;
    end
  else
    if *The i-th vehicle is in the platoon's lane **and** is not the last vehicle* **then**
      if $e_d(idx+1) + \xi_{lg} \leq 1$ ***and*** $e_d(idx) + \xi_{lg} \leq 1$ **then**
        Send a message to this vehicle to change its lane;
      else
        $d_{desired}(idx) := -\xi_{lg}$;
        $d_{desired}(idx+1) := -\xi_{lg}$;
      end
    end
  end
end

Algorithm 4: Leaving maneuver.

3.3. Controller

Until now it was covered an approach how can a vehicle join or leave a platoon. This approach is implement by modifying the tracking reference of the controller. In this section it will cover the implementation of the controller.

To implement the controller, the state space of ideal vehicle is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{3.2}$$

which $\mathbf{x} = [\, s \quad v \,]^T$, where s is the distance of vehicle and v is the velocity of a vehicle.

Let assume there are 2 vehicles, which start to accelerate with the same initial distance but with difference velocity and their state space is:

$$\dot{\mathbf{x}}_1 = \mathbf{A}\mathbf{x}_1 + \mathbf{b}u_1, \qquad \dot{\mathbf{x}}_2 = \mathbf{A}\mathbf{x}_2 + \mathbf{b}u_2. \tag{3.3}$$

Now, let assume, the error state space is given by $\mathbf{e} = \mathbf{x}_2 - \mathbf{x}_1$ and the linear time-invariant system is given

$$\begin{aligned} \dot{\mathbf{e}} &= \dot{\mathbf{x}}_2 - \dot{\mathbf{x}}_1 \\ &= \mathbf{A}(\mathbf{x}_2 - \mathbf{x}_1) + \mathbf{b}(u_2 - u_1) \\ &= \mathbf{A}\mathbf{e} + \mathbf{b}(u_2 - u_1). \end{aligned} \tag{3.4}$$

Since the initial distance of the 2 vehicles are the same, then this implies, that error state is actually contains from the distance between two vehicles and difference in their velocities.

$$\mathbf{e} = \begin{bmatrix} e_d \\ e_v \end{bmatrix}. \tag{3.5}$$

The state error represents the same data in algorithm 2. Thus, this state error will be used to design the controller. This implies a error state space for veh_i is

$$\begin{aligned} \dot{\mathbf{e}}_i &= \mathbf{A}\mathbf{e}_i + \mathbf{b}(u_i - u_{i-1}) \\ y &= [\, 1 \quad 0 \,]\mathbf{e}_i = \mathbf{c}^T\mathbf{e}_i. \end{aligned} \tag{3.6}$$

With this, it can be designed a full state feedback control to track the desired distance d_{des}

$$u_i = -\mathbf{k}^T\mathbf{e} + V d_{des_i} + u_{i-1} \tag{3.7}$$

which $\mathbf{k}^T$ is a gain vector to control the error state $\mathbf{e}$, d_{des_i} represents to the tracking reference of veh_i, u_{i-1} is the veh_{i-1}'s acceleration and V is a factor with a value

$$V = -\frac{1}{\mathbf{c}^T(\mathbf{A} - \mathbf{b}\mathbf{k}^T)^{-1}\mathbf{b}}. \tag{3.8}$$

4. Simulations and results

In this section, it will be covered the use of the algorithm, which had been discussed in section 3 in SUMO and MATLAB/SIMULINK. First, we will define our platoon problem, simulate, and discuss an example of join and leave maneuver.

4.1. Problem definition

This section will define our problem, which is created a platoon maximum of 10 followers and one leader. Vehicles running in the simulation can assume three roles: Non-platoon, Leader, and Follower. The leader will be defined before the simulation starts. There will be only one leader in the simulation. Furthermore, the leader will never leave the platoon only if there is a vehicle that wants to join from the front, the joiner will be the new leader, and the current leader will be a follower, as discussed in section 3.1. Non-platoon is a vehicle, which is not part of the platoon. However, these types can join the platoon. The last role is the followers, which they are part of the platoon. Moreover, they send their information to the leader to let the leader maintenance the platoon.

4.2. Simulation

Before we proceed with the simulation, we need to define some parameters for the simulation.

4.2.1. Simulation parameters

Vehicle parameters

Regarding the vehicle parameters (acceleration and deceleration values, vehicle length, etc.), it was hard to find evidence of any reliable example values on related work that was useful for this simulation. For this reason, the defined parameters were obtained from the SUMO proposal [**SUMO2**].

The parameter for the follower which had been using is as shown.

- Vehicle types: *passenger*.
- Length: 4 m.
- Acceleration $3\,\mathrm{m/s^2}$.
- Deceleration: $7\,\mathrm{m/s^2}$.
- Maximum speed: $35\,\mathrm{m/s}$.
- Sample time: $0.5\,\mathrm{s}$
- The rest parameters remains default.

Table 4.1 describe the color phase for each vehicle to let us easy to understand what is the behavior on vehicle, for example which is leader, sending a join request and etc.

State	**Color**
Leader	Green
Join request	Orange
Leave request	Pink
Follower	Blue
Join maneuver	Red
Leave maneuver	Violate
non-Follower	Yellow

Table 4.1.: Vehicle's color in SUMO.

Controller parameters

As it was discussed in section 3.3, we use a full state feedback controller, which is one of the most powerful for a linear time-invariant system. Although the main question is, where is supposed to place the pole of the close loop system. Therefore, we using **LQR** (**L**inear **Q**uadratic **R**egulator)

$$J = \int_0^\infty \mathbf{e}^T \mathbf{Q} \mathbf{e} + R u^2 \tag{4.1}$$

to find the optimal solution for the gain vector $\mathbf{k}^T$, where the weight matrices $\mathbf{Q}$ and R are using to find the minimal cost function of $\mathbf{e}$ and u.

Hence, the following parameters will be used for the simulation.

- $\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$.
- $R = 1$.
- $\xi_d = 10\,\text{m}$.
- $\xi_{jg} = \xi_{lg} = 15\,\text{m}$.

4.3. Simulation scenario

In this section it will be discussed the simulation scenario to simulate our implemented algorithm, that had been discussed in chapter 3. The scenario of the simulation is as following

1. In the beginning there are three vehicles join from the rear of the platoon, which are *veh1*, *veh2* and *veh3*.
2. *veh4* and *veh5* join from the side platoon.
3. Afterward, there are four vehicles send join request. One of those request declines (*veh8*). Also, two vehicles join from the side of the platoon *veh6* and *veh7*. The last vehicle (*veh9*) joins from the rear of the platoon.
4. *veh10* joins from the front of the platoon and takes the roll of the platoon.
5. *veh0* leaves the platoon. Before *veh0* obtains a response from the leader vehicle *veh11* sends a join request.
6. *veh1*, *veh4*, *veh2* and *veh3* leave the platoon.

7. In the end, one vehicle (*veh9*) leaves from the rear of the platoon then the rest follower leave the platoon.

4.3.1. Simulation in SUMO

At the beginning of the simulation, we will define a leader (*veh0*). Afterward, *veh1* appears near to the leader and sends a join request with its position. The leader receives a join request and checks where is the position of the joiner. Since the joiner is behind the leader and at the same time, it will be the first follower. The leader sends an accept response to the joiner to join the platoon. Furthermore, the leader saves the information of *veh1* in its memory. This vehicle in the platoon and in each time sample sends a new velocity to control the gap using a full state feedback controller. Right now *veh1* is in the joining phase (red color). Afterwards, *veh1* sends a join request (orange color) to the leader. It waits until it obtains a response from the leader. As mentioned, leader (*veh0*) accept the join request and *veh2* starts to merge with the platoon. Before, *veh1* and *veh2* merging with the platoon, *veh3* send also a join request, which it returns with an positive acknowledgment. Figure 4.1 represents the joining maneuver of *veh1*, *veh2* and *veh3* in SUMO.

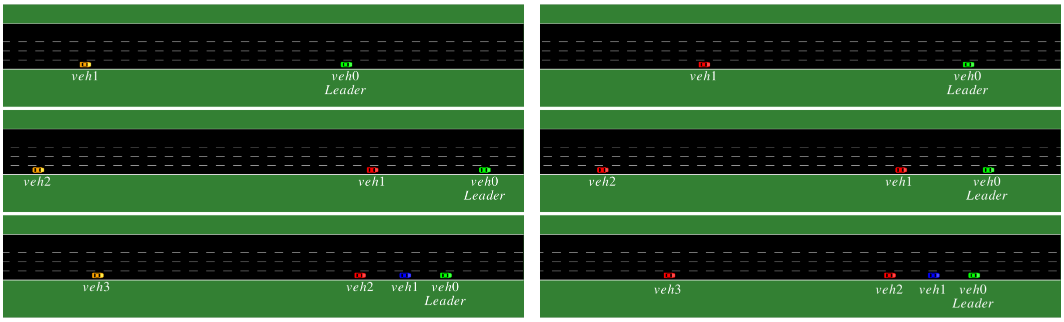


Figure 4.1.: *veh1*, *veh2* and *veh3* joining from rear of the platoon.

Right now, there is a platoon with a leader and three followers, which they had joined from the rear of the platoon. Rather than joining from the rear, let start with joining from the side of a platoon. *Veh4* sends a join request to the leader and awaits a response from the leader. The leader accepts the join request from *veh4* and saves the data of *veh4* on its memory. The leader checks in which lane is *veh4*, where is in lane 2 as is shown in figure 4.2. When the difference in the speed between *veh4* and the speed of the predecessor is less than $1\,m/s$, then the leader modify the reference of the vehicle behind *veh4*, which is *veh2* as is shown in figure 4.2, from ξ_d to ξ_{og} in reference to *veh1*. After *veh4* merge with the platoon *veh5* starts to broadcast a join request. The leader vehicle accepts the join request. Thus *veh5* starts to merge with the platoon from the side. Figure 4.2 represents the joining maneuver of *veh4* and *veh5* in SUMO.

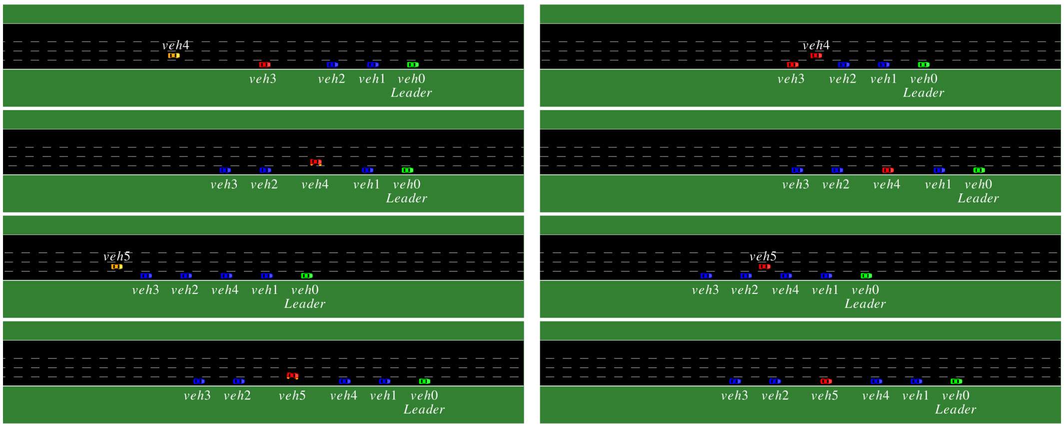


Figure 4.2.: *veh4* and *veh5* joining from side of the platoon.

The next scenario is, *veh6* sends a join request, which is returned to a positive response. *Veh6* starts to merge with the platoon, in the same approach as *veh5*. Before the merging is done, *veh7* also sends a join request, where the leader vehicle accepts the request. After some time, two vehicles transmit a join request. One of these requests will be blocked, and the other one, the leader, receives from the nearest vehicle (*veh8*). As was mention before, the leader rejects *veh8*'s request (*veh8* change its color from orange to yellow). Afterward, the leader obtains the request from *veh9* and accepts this request. Therefore *veh9* also starts to merge with the platoon from the rear. Figure 4.3 illustrates the joining of *veh6*, *veh8* and *veh9* as well as the rejection of *veh7* in SUMO.

So far, we have only examined joining from the rear and side of the platoon. Now, *veh10* sends a join request to the leader, which its position in front of the leader. After the leader receives a join request from *veh10*, leader looks on *veh10* position and see, if *veh10* is in front/behind of the leader. In this case, *veh10* is front of the leader. Therefore, *veh10* becomes the new leader when the leader accepts the join

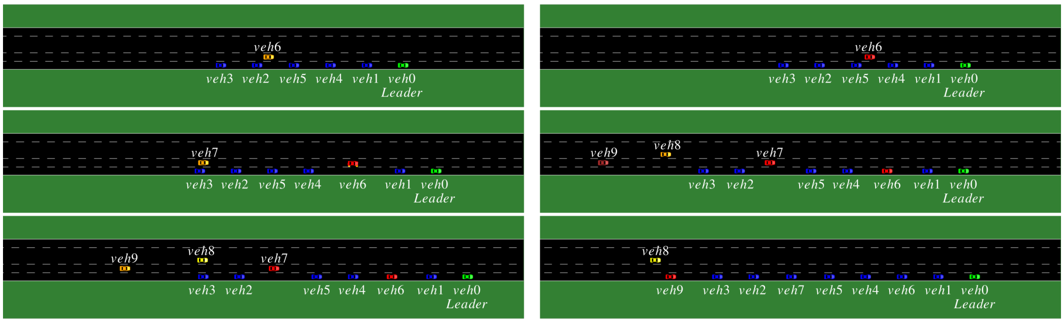


Figure 4.3.: *veh6* and *veh8* joining from side, *veh9* joining from rear, rejection of *veh7* request.

request of *veh10*. As was discussed *veh10* becomes the new leader. Before that, the current leader *veh0* sends its data (platoon data) to *veh10* as is discussed in sec 3.1. In the next time step the follower will send their data to the new leader (*veh10*), in the same time *veh10* sends to the follower their new speed to close the gap between the leader (*veh10*) and the first follower *veh0*. Figure 4.4 illustrate the joining of new leader *veh10* in SUMO.

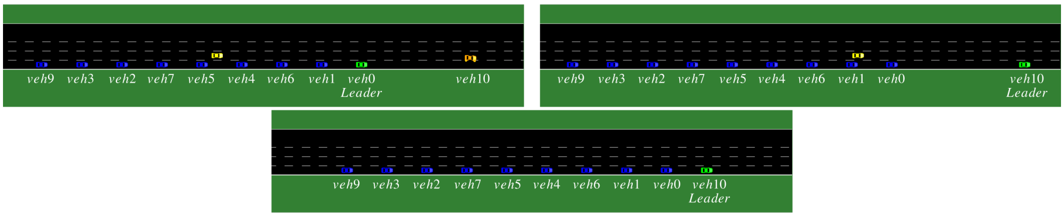


Figure 4.4.: *veh10* becomes the new leader.

So far, we have a platoon with a leader and 9 followers and the join operation had been done by joining from rear/side/front. Now let us study the joining and leaving maneuver in the same time. *Veh11* and *veh0* send stimulant a request. *Veh11* send a join request. Meanwhile, *veh0* send a request to leave from the platoon. Since *veh0* is near toward the leader, *veh10* receives this request and blocks the others. As was mentioned, the leader vehicles accepts both request. For *veh0* as well as *veh1* the tracking reference modify from ξ_d to ξ_{lg} to create enough gap. Hence, *veh0* can change its lane safely, when there is enough space to leave the platoon. After *veh0* leaves successfully the tracking reference of *veh1* return to ξ_d. Therefore *veh1* starts to accelerate to close the gap, which is left due to leaving of *veh0*. Figure 4.5 illustrates the joining and leaving maneuver of *veh11* and *veh0* in SUMO.

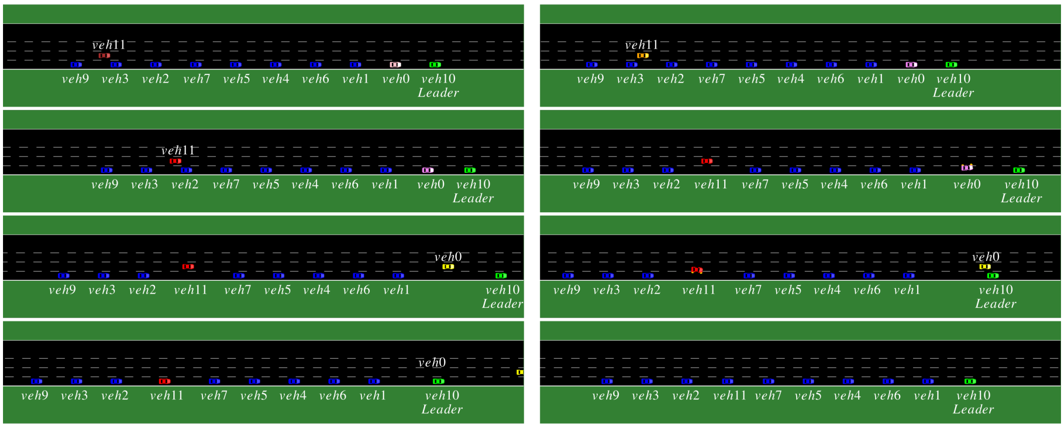


Figure 4.5.: *veh0* leaving from side and *veh11* joining from side.

Four vehicles send a leave request in the same time, which are *veh1*, *veh4*, *veh3* and *veh2*. The leader receive the nearest vehicle (*veh1*) and block the other request. After the leader response to *veh1*, which is a positive response, the leader receive the request from *veh4*. All requests returns with a positive response and these vehicles start the leaving maneuver (violate color). Figure 4.6 illustrates the leaving maneuver for *veh1*, *veh4*, *veh3* and *veh2* in SUMO.

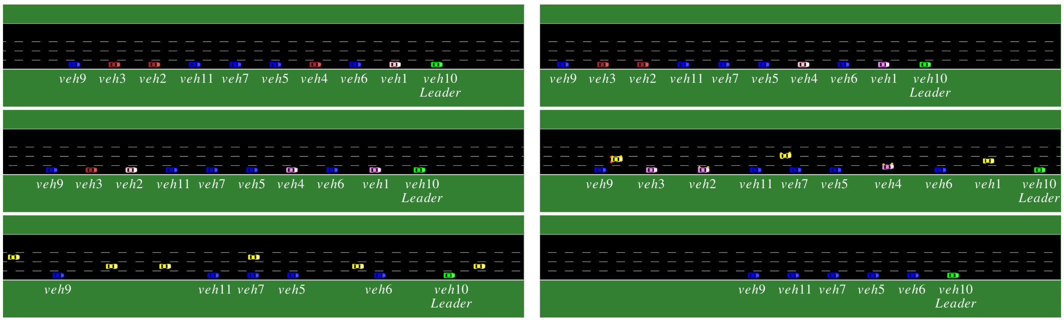


Figure 4.6.: *veh1*, *veh4*, *veh3* and *veh2* leaving from side of the platoon.

veh9 decides to leave the platoon. Therefore it broadcast a leave request. The leader vehicle receives the request and accept it. After the acceptance *veh9*'s tracking reference modify from ξ_d to ξ_{lg}. Figure 4.7 illustrates the leaving maneuver for *veh9* in SUMO.

The current leader decides to leave the platoon. Therefore, platoon leader notify

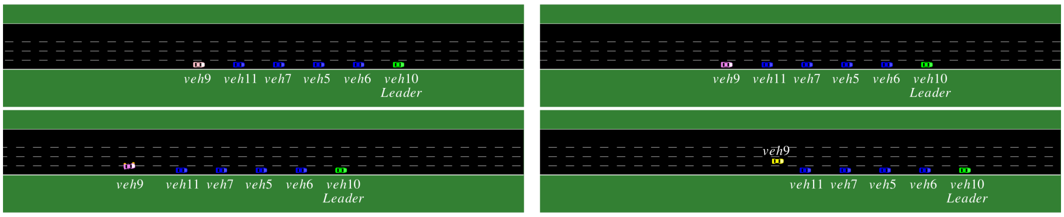


Figure 4.7.: *veh9* leave from rear of the platoon.

the first follower (*veh6*) that it will be the new leader. Afterward, *veh6*'s tracking reference modify from ξ_d to ξ_{lg}. When *veh6*'s inter-distance converge to modifying reference ξ_{lg} the leader (*veh10*) change its lane. *Veh6* takes the roll of the platoon after the confirmation, that *veh10* change its lane safely, and other followers start to send their information to the new leader *veh6*. Figure 4.8 illustrates the leaving maneuver of the platoon leader and *veh6* takes the roll of the platoon.

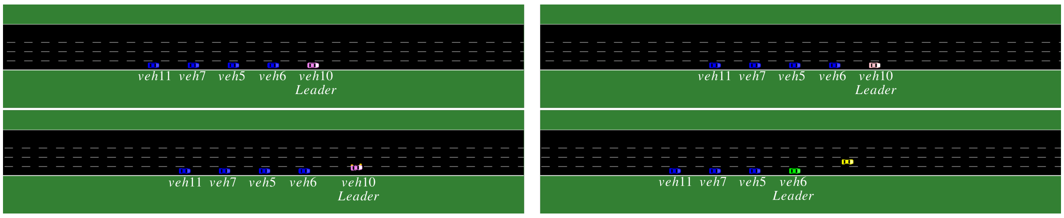


Figure 4.8.: Leaving of the platoon leader.

In the end of the simulation, the rest vehicles send a request to leave the platoon, which is return with a positive acknowledgment. Figure 4.9 illustrates the leaving maneuver for the rest follower in SUMO.

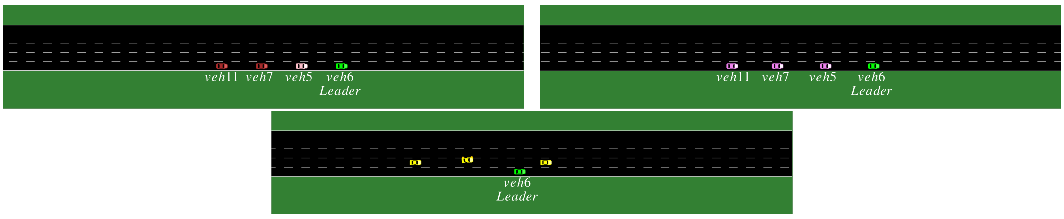
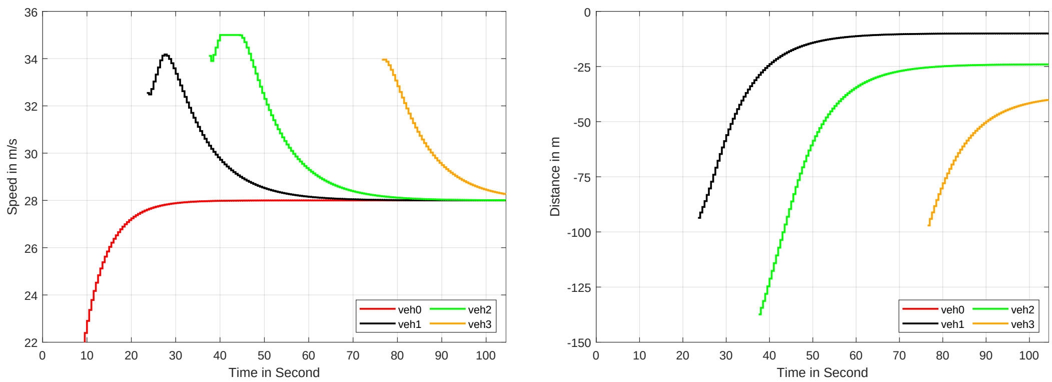


Figure 4.9.: Leaving of the rest followers.

4.4. Result

This section describes the behavior of the Platoons during the application run time, focusing and discussing the maneuvers and the possible reasons behind those results.



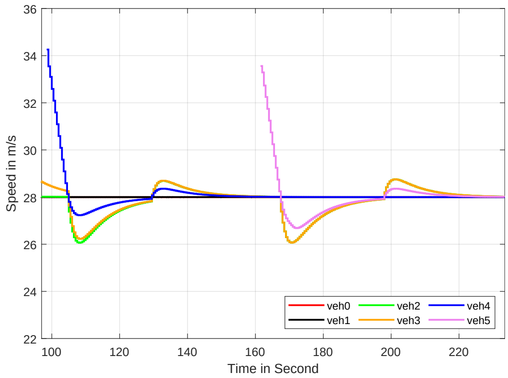
(a) Velocity

(b) Distance reference to the leader

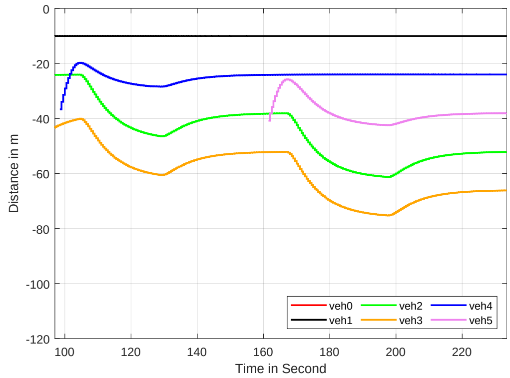
Figure 4.10.: The behavior of the joiners from the rear.

Figure 4.10 represents the behavior of *veh1*, *veh2* and *veh3*, when they was joining the platoon. Figure 4.10a illustrates the velocity of the followers, from the moment that they receive an acceptance from the leader. Figure 4.10b illustrates the behavior of follower's distance in reference to the leader. From this figure it can be observed there is no issue with the joining from the rear of a platoon. A vehicle accelerates if it is far from the last follower and start to decelerate when it is near to the desired distance.

Figure 4.11 represents the behavior of *veh4* and *veh5*, when they at the same time was joining from the tail of the platoon. Figure 4.11a illustrates the velocity of the followers, from the moment that they receive an acceptance from the leader. Figure 4.11b illustrates the behavior of the follower's distance in reference to the leader. As we can see, *veh4* starts to deceleration the moment it obtains an acceptance to join the platoon. Additionally, *veh2* deceleration the moment that the difference between the velocity of *veh4* and the *veh1* is 1 m/s. After *veh4* changes its lane, it starts to accelerate to close the gap. The same behavior is valid for *veh5*.

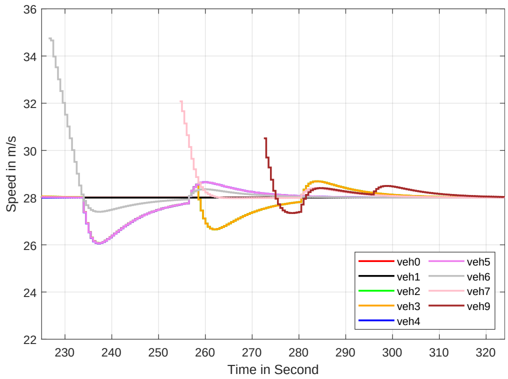


(a) Velocity

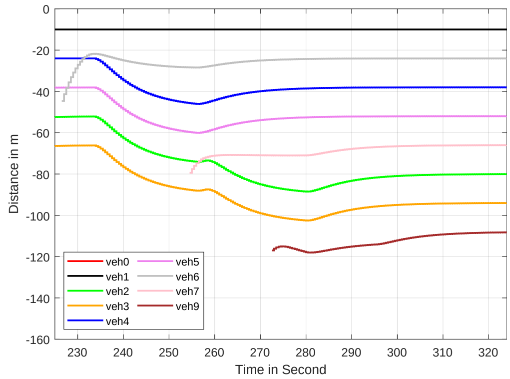


(b) Distance reference to the leader

Figure 4.11.: The behavior of the joiners from the side.



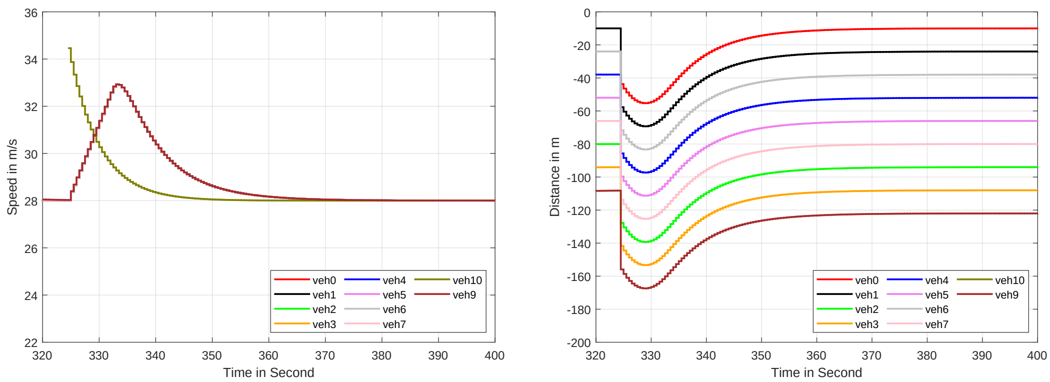
(a) Velocity



(b) Distance reference to the leader

Figure 4.12.: The behavior of two vehicles join from the side.

Figure 4.12 represents the behavior of *veh6*, *veh7* and *veh9*, where *veh6* and *veh7* was joining from the side of the platoon and *veh9* from the tail of the platoon. Figure 4.12a illustrates the velocity of the followers, from the moment that they receive an acceptance from the leader. Figure 4.12b illustrates the behavior of follower's distance in reference to the leader. As it can be observed, the maneuver of *veh6* is the same as of *veh5*, there is no issue. However, the moment that *veh6* changes its lane *veh7* starts its merging. As it can be observed *veh7* dose not behave the same as *veh6*. That is, *veh5* accelerate due to the acceleration of *veh6* to close the gap. Therefore *veh7* dose not require to decelerate to reach the modifying reference. Before *veh7* changes its lane *veh9* starts the merging phase. As is shown in figure 4.3 *veh9* is not in the platoon's lane. Hence, it is controlled to reach the modifying reference ξ_{jg}. While *veh9* attempt to reach ξ_{jg}, *veh7* change its lane and starts to accelerate to close the gap. Therefore *veh9* starts also to accelerate to maintain the gap. After *veh9* receives the rights to change its lane, it accelerates again to close the gap.

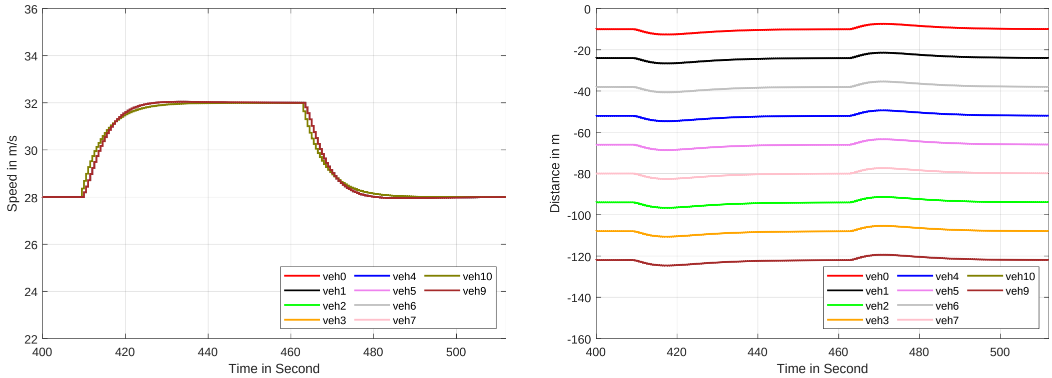


(a) Velocity (b) Distance reference to the leader

Figure 4.13.: The behavior of the follower with the new leader.

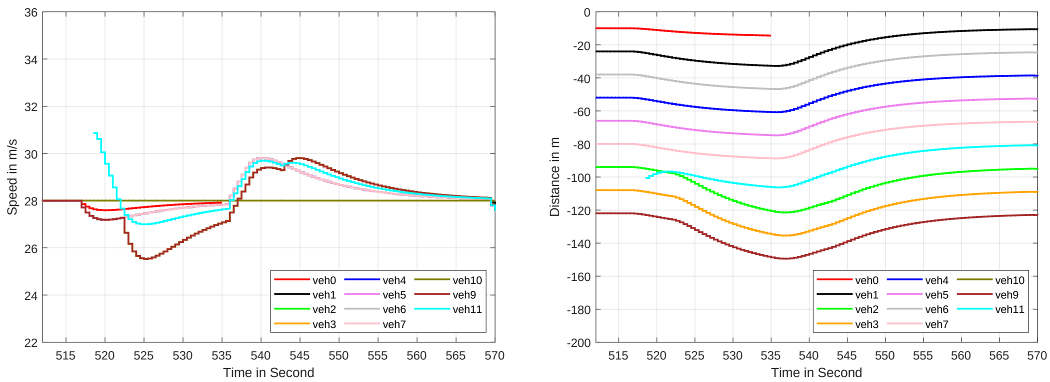
Figure 4.13 represents the behavior of *veh10*, when it became the new leader of the platoon was joining the platoon. Figure 4.13a illustrates the velocity of the followers, from the moment that *veh7* became the leader. Figure 4.13b illustrates the behavior of the follower's distance in reference to the new leader. As we can see from this figure 4.13, the followers increase their velocity to reach the new leader.

Figure 4.14 represents the followers' behavior when the leader changes its speed. As it can be observed, the changing in the velocity with 4 m/s does not affect the inter-distance since each vehicle's acceleration obtains the predecessor's acceleration.



(a) Velocity (b) Distance reference to the leader

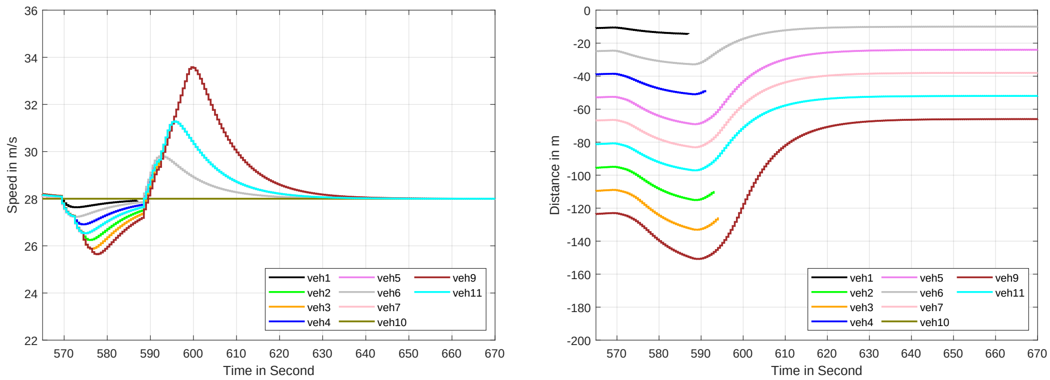
Figure 4.14.: The behavior of the follower when the leader changes its velocity.



(a) Velocity (b) Distance reference to the leader

Figure 4.15.: The behavior of joining and leaving vehicles.

Figure 4.15 represents the behavior of *veh11*, when it was joining the platoon and of *veh0*, when it was leaving the platoon. Figure 4.15a illustrates the velocity of the followers. Figure 4.15b illustrates the behavior of follower's distance in reference to the leader. As it can be seen *veh0*, it is starting to slow down due to modifying the reference ξ_{lg} and its predecessor. When *veh0* leaves the platoon, *veh1* starts to accelerate to close the gap, that left it by *veh0*.

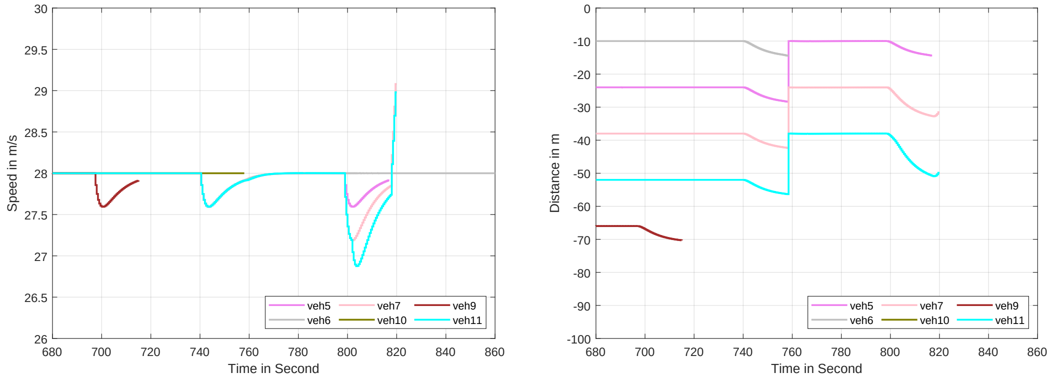


(a) Velocity (b) Distance reference to the leader

Figure 4.16.: The behavior of leaving four vehicles.

Figure 4.16 represents the behavior of *veh4*, *veh5*, *veh6* and *veh8*, when they were leaving the platoon. Figure 4.16a illustrates the velocity of the followers, and figure 4.16b illustrates the behavior of the follower's distance in reference to the leader. It can be noticed that process too many leave requests simultaneously; the last will be far from the leader. Furthermore, it will obstruct the traffic flow in the road.

Figure 4.17 represents the rest follower's behavior when they were leaving the platoon. Figure 4.17a illustrates the velocity of the followers, and Figure 4.17b illustrates the behavior of the follower's distance in reference to the leader. It can be observed *veh9* was slowing down to reach ξ_{lg}. When *veh9* reaches it, it changes its lane and leaves the platoon. For the case that the platoon leader was leaving; all followers were slowing down because of the modifying reference between leader *veh10* and *veh6*. The speed behavior of the follower, as is shown in figure 4.17a, is the same speed behavior of *veh9* when it was leaving. For the case, the rest vehicles were leaving; all these vehicles were slowing because of the modifying reference. Consequently, when *veh5* left the platoon it left a gap between the leader vehicle and



(a) Velocity (b) Distance reference to the leader

Figure 4.17.: All vehicles leaving.

veh7. Therefore *veh7* started to accelerate to reach ξ_{lg} and in the same was changing its lane. That is explain why *veh7* and *veh11* accelerated after the leaving of *veh7*.

5. Conclusion and future work

5.1. Conclusion

This thesis's objective is to implement an algorithm, which is to maneuver vehicles into a platoon. In chapter 3 it is covered the algorithm and how it works. Further, the approach of the joining/leaving maneuver from the side was discussed by modifying the tracking reference to create space. With that space, a joiner/leaver can join/leave safely. Chapter 4 covered the simulation of the algorithm and the result of the simulation. The joining from the tail vehicles accelerates or decelerates depending on the result, which is the leader given to them from the controller. Thus the joining from the tail depending the most on the controller. In the joining from the side, the joiner and the behind one, are modified their tracking reference. The tracking reference will not trigger until the difference in speed between predecessor and joiner converge to some specific value. When the modified reference is triggered, the vehicle behind the joiner decelerate. Therefore, all other vehicles also decelerate. This approach is triggered even if a vehicle joins from the tail, but it is not in the lane platoon, and the longitude distance is smaller than the tracking reference as was shown in figure 4.12. Joining from the front is depending on the controller. Because, after the joiner obtains an acceptance, it becomes the new platoon's leader. The follower, as well as the previous leader, accelerate to maintain the platoon. The approach of leaving is also modified the tracking reference. This approach is triggered after is obtaining acceptance from the leader. The leaver and behind one decelerate depending on the value, which is obtained from the leader. After the vehicles converge to modified tracking, the leaver change its lane and disconnect from the platoon. Hence, the vehicles accelerate to close the space, which is a lift due to leaving a vehicle. The same approach is valid for leaving from the tail of the platoon. Although in this case, only the leaver will be modified its tracking reference.

5.2. Future work

The following topics are some suggestion to investigate in the future:

1. Splitting a platoon into two parts or merging two platoons into one platoon.
2. the issue when there is a non-follower between followers.

Appendix

Appendix A.

State feedback controller

Assume the following linear time invariant system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \tag{A.1}$$

$$y = \mathbf{c}^T\mathbf{x} \tag{A.2}$$

and the system is controllable, i.e, the matrix

$$\mathbf{S} = [\,\mathbf{b} \quad \mathbf{A}\mathbf{b} \quad \ldots \quad \mathbf{A}^{n-1}\mathbf{b}\,] \tag{A.3}$$

is a full rank matrix (rank($\mathbf{S}$) = n). Full state feedback is applied by controlling the input u with a gain vector $\mathbf{k}^T$

$$u(t) = -\mathbf{k}^T\mathbf{x}(t). \tag{A.4}$$

Substitute eq. A.4 in A.1 is obtained the following closed loop system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x} - \mathbf{b}\mathbf{k}^T\mathbf{x}(t) = (\mathbf{A} - \mathbf{b}\mathbf{k}^T)\mathbf{x}(t). \tag{A.5}$$

Thus, the gain vector $\mathbf{k}^T$ is chosen such that the matrix $\mathbf{A} - \mathbf{b}\mathbf{k}^T$ is a Hurwitz matrix. That is, all eigenvalues of matrix $\mathbf{A} - \mathbf{b}\mathbf{k}^T$ is located in the left side of s-plane.

To make the output y track a constant reference $r(t) = r$ the input extends to

$$u = -\mathbf{k}^T\mathbf{x} + Vr \tag{A.6}$$

and the closed loop system

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{b}\mathbf{k}^T)\mathbf{x} + \mathbf{b}Vr \tag{A.7}$$

$$y = \mathbf{c}^T\mathbf{x}. \tag{A.8}$$

The desired output after long time

$$\lim_{t\to\infty} y(t) = \mathbf{c}^T \mathbf{x}_\infty = r. \tag{A.9}$$

Since the closed loop matrix is a Hurwitz matrix, the state vector $\mathbf{x}$ will converge to an equilibrium point $\mathbf{x}_\infty$. From A.7 we obtain after long time

$$\lim_{t\to\infty} \dot{\mathbf{x}} = 0 = (\mathbf{A} - \mathbf{b}\mathbf{k}^T)\mathbf{x}_\infty + \mathbf{b}Vr \tag{A.10}$$

Hence, the equilibrium point $\mathbf{x}_\infty$ is

$$\mathbf{x}_\infty = -(\mathbf{A} - \mathbf{b}\mathbf{k}^T)^{-1}\mathbf{b}Vr \tag{A.11}$$

Substitute $\mathbf{x}_\infty$ A.9 in A.10 we obtain the gain factor V.

$$-\mathbf{c}^T(\mathbf{A} - \mathbf{b}\mathbf{k}^T)^{-1}\mathbf{b}Vr = r \quad \Longrightarrow V = -\frac{1}{\mathbf{c}^T(\mathbf{A} - \mathbf{b}\mathbf{k}^T)^{-1}\mathbf{b}}. \tag{A.12}$$

A.1. Linear Quadratic Regulator

The **L**inear **q**uadratic **R**egulator (LQR) is an optimal problem to find the optimal solution of the state feedback control

$$u = -\mathbf{k}^T\mathbf{x}$$

The gain $\mathbf{k}^T$ is obtained from the minimal cost function J

$$\begin{aligned} &\min_u J = \int_0^\infty \mathbf{x}^T\mathbf{Q}\mathbf{x} + Ru^2\,\mathrm{d}t \\ &\text{Subject to } \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \end{aligned} \tag{A.13}$$

Where $\mathbf{Q} > 0$ and $R > 0$. Solving the optimal problem, the gain $\mathbf{k}^T$ is defined by

$$\mathbf{k}^T = R^{-1}\mathbf{b}^T\mathbf{P} \tag{A.14}$$

and the matrix $\mathbf{P}$ is found by solving the algebraic Riccati equation:

$$\mathbf{PA} + \mathbf{A}^T\mathbf{P} - \mathbf{PbR}^{-1}\mathbf{b}^T\mathbf{P} = -\mathbf{Q} \tag{A.15}$$