October 15, 2018

MASTER THESIS

Modeling automated vehicles merging into a platoon on the highway

Annelien van der Es

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)
Department of Applied Mathematics (AM)
Chair Hybrid Systems (HS)

Supervisor: prof. dr. A. A. Stoorvogel

Graduation committee:
prof. dr. A. A. Stoorvogel
dr. ir. G. Meinsma
dr. K. Smetana

UNIVERSITY OF TWENTE.

Summary

A platoon consists of several automated vehicles driving as one group on the road. Because of communication between cars and lack of human drivers that require time to react, the distance between vehicles can be decreased, improving traffic flow. One problem regarding platoons is the situation when a new vehicle wants to enter the highway. Space between vehicles is small, so the platoon has to make a gap for the merging vehicle, while the merging car has to arrive at the gap in the platoon at a specific time. The goal is to extend and improve a current model for platoons such that a new vehicle is able to merge into the platoon when it enters the highway.

First, the platoon is modeled. Based on the dynamics that describe a single vehicle and the desired distance between two vehicles, it is possible to define an error and derive the error dynamics. Since the error should converge to zero, a distributed controller is introduced such that the resulting system is stable. This model for a platoon is extended by defining a new error that takes into account both the vehicle in front and the vehicle behind a given vehicle. With this new error and the same distributed controller, a more general model is derived and it is proven that this system is asymptotically stable when certain constraints are met.

Next, the model for the platoon is adapted such that a vehicle is able to merge into the platoon on the highway. This requires creating a gap in the platoon for the merging vehicle and controlling this merging vehicle such that it will drive next to the created gap at some point. Furthermore, since the merging vehicle is merging onto a highway, the merge must be completed before the merging vehicle reaches the end of the acceleration lane. Two slightly different models are proposed: the ‘open-loop’ model aims to complete the merge at a given time, whereas in the ‘closed-loop’ model the merging vehicle merges between two given locations. Both models follow a similar approach:

1. A time or location is chosen where the vehicle should merge into the platoon.
2. Based on this time or location, and possibly additional (estimated) variables, two vehicles of the platoon are selected to increase the distance between them to make room for the merging vehicle.
3. The platoon and the merging vehicle are controlled such that at the chosen time or position the merging vehicle can merge into the platoon between the two chosen vehicles.

For merging onto the highway, using the ‘closed-loop’ model is the best choice. Simulations are performed to show that with the proposed models a vehicle is able to successfully merge into a platoon that is driving on the highway.

Contents

Nomenclature

A	system matrix of the error dynamics	
A_u	part of the system matrix of the closed-loop platoon dynamics	
B	input matrix of the error dynamics	
B_r	matrix used in the closed-loop platoon dynamics with a time-dependent standstill distance r	
B_u	part of the input matrix of the closed-loop platoon dynamics	
I_n	identity matrix of size $n \times n$	
$I_{(-1),\,n}$	matrix of size $n \times n$ with ones on the diagonal below the main diagonal and zeros elsewhere	
$I_{(1),\,n}$	matrix of size $n \times n$ with ones on the diagonal above the main diagonal and zeros elsewhere	
J	matrix used in the closed-loop platoon dynamics with a time-dependent standstill distance r	
J_m	zero matrix of size $n \times n$ with the exception that $J_m(m,\, m-1) = \beta$ and $J_m(m-1,\, m-1) = 1-\beta$	
L_{com}	maximum distance that the merging vehicle can be away from the platoon and still communicate to the platoon	m
L_{gap}	additional distance needed to create a gap	m
L_i	length of vehicle i	m
L_{merging}	length of the merging vehicle	m
M	matrix	
P, Q	positive definite matrix	
R	vector containing the second and third time derivatives of all standstill distances r_i	
R_m	vector containing the standstill distance r_m of vehicle m and its first two time derivatives	
R_{merging}	vector containing the standstill distance r_{merging} of the merging vehicle and its first two time derivatives	
U	lumped input state vector	
X	lumped error state vector	

a_0	acceleration of the virtual reference vehicle	m/s^2
a_i	acceleration of vehicle i	m/s^2
a_{i0}	similar to a_{ij}, defines communication of the virtual reference vehicle to vehicle i	
a_{ij}	element of the adjacency matrix $\mathcal{A}$	
a_{merging}	acceleration of the merging vehicle	m/s^2
a_{ramp}	constant acceleration of the merging vehicle used to estimate t_{ramp}	m/s^2
$c_0, \ldots, c_5$	polynomial coefficients	
d_i	distance between vehicles i and $i-1$	m
d_i^d	desired distance between vehicles i and $i-1$	m
d_{merge}	distance required between vehicle m and the merging vehicle and between the merging vehicle and vehicle $m-1$ before the merging vehicle is able to merge	m
d_{merging}	distance between the merging vehicle and the virtual reference vehicle driving in front of the merging vehicle	m
d^d_{merging}	desired distance between the merging vehicle and the virtual reference vehicle driving in front of the merging vehicle	m
e_i	error of vehicle i	m
h	time gap	s
i, j	vehicle index	
k	controller gain vector	
k_1, k_2, k_3	element of the controller gain vector	
l_{ij}	element of the Laplacian matrix $\mathcal{L}$	
$\hat{l}_{ij}$	element of $\hat{\mathcal{L}}$	
m	the vehicle in front of which the merging vehicle will merge into the platoon	
m_{ij}	element of matrix M	
n	number of vehicles in the platoon	
p_i	diagonal element of the pinning matrix $\mathcal{P}$	
q_0	position of the virtual reference vehicle	m
q_{end}	position of the end of the acceleration lane	m
q_i	position of vehicle i	m
q_{ij}	element of matrix Q	
q_{merging}	position of the merging vehicle	m
q_{ramp}	position of the end of the entrance ramp and beginning of the acceleration lane	m
q_{start}	position of the beginning of the entrance ramp	m

r	fixed default standstill distance	m
r_i	standstill distance for vehicle i	m
r_{merging}	standstill distance for the merging vehicle	m
t	time	s
$t_{\text{accelerate}}$	the amount of time it takes the merging vehicle to accelerate to the platoon velocity with constant acceleration a_{ramp}	s
t_{align}	the amount of time given for the final adjustments in aligning the merging vehicle with the gap in the platoon	s
t_{merge}	the time where the merging vehicle merges into the platoon	s
t_{ramp}	the time where the merging vehicle leaves the entrance ramp and enters the acceleration lane	s
t_{start}	the time where the merging vehicle enters the entrance ramp to the highway and the platoon and the merging vehicle start preparing for the merge	s
$\tilde{t}_{\text{ramp}}$	estimated value of t_{ramp}	s
u_0	desired acceleration of the virtual reference vehicle driving in front of the platoon	$\mathrm{m/s^2}$
$u_{0,\,\text{merging}}$	desired acceleration of the virtual reference vehicle driving in front of the merging vehicle	$\mathrm{m/s^2}$
u_i	desired acceleration of vehicle i	$\mathrm{m/s^2}$
u_{merging}	desired acceleration of the merging vehicle	$\mathrm{m/s^2}$
$\bar{u}_i$	new input signal for the error dynamics	$\mathrm{m/s^2}$
v_0	velocity of the virtual reference vehicle driving in front of the platoon	$\mathrm{m/s}$
$v_{0,\,\text{merging}}$	velocity of the virtual reference vehicle driving in front of the merging vehicle	$\mathrm{m/s}$
v_i	velocity of vehicle i	$\mathrm{m/s}$
v_{merging}	velocity of the merging vehicle	$\mathrm{m/s}$
v_{platoon}	platoon velocity defined to be the velocity v_0 of the virtual reference vehicle driving in front of the platoon	$\mathrm{m/s}$
v^d_{platoon}	desired platoon velocity	$\mathrm{m/s}$
$\bar{v}_0$	constant velocity of the virtual reference vehicle	$\mathrm{m/s}$
$\bar{v}_{\text{platoon}}$	constant platoon velocity	$\mathrm{m/s}$
x	vector	
x_i	error state vector of vehicle i / element of vector x	
x_{merging}	error state vector of the merging vehicle	
α	scalar weight between 0 and 1 used in the communication topology	
β	scalar weight between 0 and 1 used in the error	

δ_{end}	distance from the merging vehicle to the end of the acceleration lane q_{end}	m
δ_i	distance from vehicle i to the beginning of the acceleration lane q_{ramp}	m
$\delta_{\mathrm{merging}}$	distance from the merging vehicle to the beginning of the acceleration lane q_{ramp}	m
λ_i	eigenvalue of $\hat{\mathcal{L}}$	
τ	drive-line dynamics time constant	s
ϕ	drive-line dynamics time delay	s

$\mathcal{A}$	adjacency matrix
$\mathcal{E}$	edge set representing the communication links between vehicles
$\mathcal{E}_0$	same as $\mathcal{E}$ but includes additional edges for communication with the virtual reference vehicle
$\mathcal{L}$	Laplacian matrix
$\hat{\mathcal{L}}$	sum of Laplacian matrix $\mathcal{L}$ and pinning matrix $\mathcal{P}$
$\mathcal{N}_i$	neighboring set of vehicle i
$\mathrm{O}_{m\times n}$	zero matrix of size $m \times n$
$\mathcal{P}$	pinning matrix
$\mathcal{V}$	set of vehicles that form the platoon
$\mathcal{V}_0$	set of vehicles that form the platoon plus the virtual reference vehicle

$\otimes$	Kronecker product
$\mathbb{C}$	set of all complex numbers
$\mathbb{R}$	set of all real numbers
$\mathbb{R}^{m\times n}$	set of all real $m \times n$ matrices
$\mathbb{R}^n$	set of all real $n \times 1$ matrices
$A(:, i)$	i-th column of a matrix A
$A(i, :)$	i-th row of a matrix A
$A(i, j)$	(i, j)-th element of a matrix A
A^T	transpose of a matrix A
$\dot{x}$	time derivative of a variable x
$\ddot{x}$	second time derivative of a variable x
$\dddot{x}$	third time derivative of a variable x

1. Introduction

As roads are getting more crowded day by day, it becomes increasingly difficult to handle the growing numbers of vehicles while avoiding traffic congestion and guaranteeing the safety of users. Improving the existing infrastructure, however, is not always possible due to, for example, high costs or a lack of space. Another possible solution is to introduce (partially) automated vehicles, which will accelerate and decelerate based on the surrounding traffic. Such vehicles use the signals obtained by on-board sensors or through communication with nearby vehicles to adjust their speed. Using fully automated vehicles would eliminate slow reactions and mistakes from human drivers and also enable vehicles to better anticipate upcoming obstacles or future actions of other vehicles. As a result, vehicles will then be able to drive closer together, which would improve traffic flow and increase the road capacity allowing more vehicles to drive on any given part of the road, while it would still be possible to guarantee the safety of the passengers.

With recent technological advancements in current cars such as automatic parking, lane keeping assist, and adaptive cruise control, which is able to adjust its speed based on how fast the vehicles ahead are driving, a first step towards fully automated vehicles has been made. The next step would be to let these automated vehicles communicate with each other. This is possible with the cooperative adaptive cruise control, which is an extension of the adaptive cruise control that allows vehicles to receive information such as the velocity or acceleration of nearby vehicles through wireless communication.

When vehicles are at least automated in the longitudinal direction, meaning that their velocities are automatically adjusted to the surrounding traffic, and they are able to communicate with each other in some way, it is possible to let these vehicles drive grouped together in a so-called platoon. This is the case, for example, when vehicles are equipped with cooperative adaptive cruise control. A platoon consists of several automated vehicles that are driving close together on the road in a single line. The vehicles that are driving in the platoon will all try to maintain the same velocity and will accelerate or decelerate together. The first vehicle of the platoon, the leader of the platoon, determines the speed of the platoon. The remaining vehicles, the followers, follow behind this vehicle at a desired distance and hence will strive to maintain the same velocity as the leader. Each vehicle in the platoon is only able to communicate with the vehicles that are nearby, often only with its direct neighbors. This means that each vehicle only has access to local information. Not all vehicles know, for example, the desired velocity of the platoon.

Platoons are formed by controlling the distances between the vehicles. The desired distance between vehicles can be defined by using a constant distance, which is mostly used in truck platooning, or by using a constant time gap, as is used in the cooperative adaptive cruise control [7]. By using a spacing policy that is based on a constant time gap the desired distance between two vehicles becomes dependent on their velocities. In combination with a constant standstill distance this results in an often used spacing policy that is known to improve safety and string stability [6, 10, 11, 21, 22]. Since the leader of the platoon does not follow behind a vehicle, for the leader only the velocity needs to be controlled. It is also possible to introduce a virtual leader for the platoon that drives at the desired platoon velocity such that the distance of the first vehicle of the platoon to the virtual leader can be controlled in the same way as the distances between the other vehicles. The main goal of platooning is to decrease the distances between vehicles such that they drive closer together. Advantages of letting vehicles drive in platoons therefore include a higher road capacity, improved traffic flow, increased safety, and reduced fuel consumption due to improved aerodynamics [7, 19, 21].

Platoons have mainly been modeled as distributed systems, since they only exchange local information and no centralized control is required. For the same reasons, many approaches to controlling the vehicles in platoons include distributed controllers or control algorithms. Since vehicles have to agree on the distances between them and their velocities, often distributed consensus algorithms are used when dealing with cooperative vehicles [12, 13]. Such methods from distributed consensus control are applied specifically to platoons in [19, 21, 22, 24]. In [1] a slightly different approach is used based on partial differential equations to model vehicles equipped with cooperative adaptive cruise control. And also [18] uses a method based on partial differential equations for modeling platoons. Other methods for controlling the vehicles of a platoon include model predictive control which is based on solving constrained optimization problems [2, 3].

There have been many successful trials with platoons of automated vehicles, where passenger cars were equipped with the necessary hardware and control systems [6, 8, 9, 10, 22, 23], but there are also still many problems that need to be solved before platoons will be used on a large scale. For example, vehicles will need to be able to join and leave platoons, platoons need to be able to merge together into one big platoon or split into smaller ones, and vehicles entering a highway need to be able to merge with a platoon that is already driving on the highway. Some problems have already been studied in recent research. In [23] velocity constraints are introduced to model the case where a slow or faulty vehicle joins the platoon. Instead of the homogeneous platoons that are often considered, [19] models a heterogeneous platoon taking into account differences between vehicles. In [1] the effect of parametric uncertainties on the stability of platoons is studied. And [2] focuses on control of a mixed traffic flow that consists of both autonomous vehicles and human driven vehicles, since it is very likely that they will coexist in the future. The focus of this thesis will be the control of a platoon on a highway and a separate vehicle entering the highway such that the vehicle will merge with the platoon.

Some previous research has been done regarding the merging of platoons. In [17], a lane change maneuver is proposed for several different cases of lane changes. Some maneuvers such as forming a platoon, merging, or splitting are considered in [19]. The merging of a vehicle into a platoon in the adjacent lane is implemented in [19] as follows: the merging vehicle decides that it will become the i-th vehicle in the platoon, a virtual copy of the $(i-1)$-th vehicle is created in the lane of the merging vehicle such that it will adjust its position and velocity accordingly, then a virtual copy of the merging vehicle is inserted into the platoon such that it will create a gap for the merging vehicle, and finally the vehicle merges into the platoon. This is simulated in [19] for a platoon consisting of three vehicles. Scenarios considered in [5] do include the merge of a vehicle into a platoon and a lane reduction where two platoons need to merge because one of the lanes ends, which could be applied to merging onto a highway, but the research is mostly focused on the design of an interaction protocol. For merging onto the highway, however, the merging protocols as described above are not very practical. On a highway, such as the one in figure 1.1, it is only possible for a vehicle to enter the highway while it is driving in the acceleration lane. And if a platoon is driving in the adjacent lane of the highway at the same time, the vehicle will have to merge into the platoon before the end of the acceleration lane, which means that there is a limited window to complete the merge.

As mentioned before, this thesis will model the merge of an automated vehicle into a platoon that is driving on the highway. The situation that will be discussed is illustrated in figure 1.1. After driving on the entrance ramp to the highway, a vehicle is now driving in the acceleration lane of the highway. This vehicle wants to enter the highway by changing to the adjacent lane, which should be done before the vehicle reaches the end of the acceleration lane. In the adjacent lane, however, a platoon is driving. And because the distances between vehicles in the platoon are small, the vehicle is not able to change lanes. The goal is therefore to control the vehicles of the platoon and the vehicle in the acceleration lane such that it can merge into the platoon.

To model the merge of a vehicle into a platoon, however, we will first need to consider the behavior of a platoon. Since the velocities of vehicles need to be automated in order to form a platoon, only

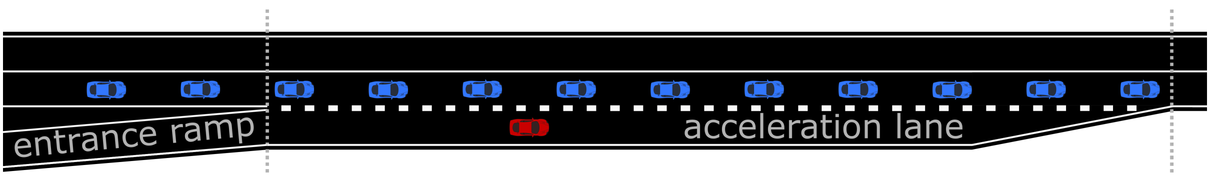


Figure 1.1: The highway.

the longitudinal motion of the platoon will be considered. Based on the distributed methods used in [21, 22, 24] a model will be derived for a platoon. This model will be able to capture the behavior of a platoon, but will not include any maneuvers that are necessary for letting a vehicle merge into a platoon. In particular, the following problems will have to be considered. First of all, the platoon should be able to increase the distance between two vehicles such that a gap is created for the merging vehicle. Secondly, the merging vehicle, which usually has a lower initial velocity than the platoon on the highway, has to accelerate to match the speed of the platoon and at the same time make sure that it is driving in the correct position with respect to the platoon such that it will be able to merge into the platoon. In addition, it should be decided between which vehicles of the platoon the vehicle will merge. And finally, the merge should take place before the merging vehicle reaches the end of the acceleration lane.

The outline of this thesis is as follows. First, the platoon is modeled in chapter 2. Based on the dynamics that describe the longitudinal motion of one vehicle and the desired distance between two vehicles, it is possible to define an error and derive the error dynamics. Since the error should converge to zero, a distributed controller is introduced such that the resulting system is stable. This model for a platoon is extended by defining a new error that takes into account both the vehicle in front and the vehicle behind a given vehicle. With this new error and the same distributed controller, a more general model is derived and it is proven that this system is asymptotically stable when certain constraints are met. Next, in chapter 3 the model for the platoon is adapted such that a vehicle is able to merge into the platoon on the highway. This requires creating a gap in the platoon for the merging vehicle and controlling this merging vehicle such that it will drive next to the created gap at some point. Furthermore, since the merging vehicle is merging onto a highway, the merge must be completed before the merging vehicle reaches the end of the acceleration lane. Two slightly different models are proposed: the 'open-loop' model of section 3.2 aims to complete the merge at a given time, whereas in the 'closed-loop' model of section 3.3 the merging vehicle merges between two given locations. Both models follow a similar approach:

1. A time or location is chosen where the vehicle should merge into the platoon.
2. Based on this time or location, and possibly additional (estimated) variables, two vehicles of the platoon are selected to increase the distance between them to make room for the merging vehicle.
3. The platoon and the merging vehicle are controlled such that at the chosen time or position the merging vehicle can merge into the platoon between the two chosen vehicles.

For merging onto the highway, using the 'closed-loop' model is the best choice. Chapter 4 gives some details on how the models are implemented and in chapter 5 some simulations are performed to show that with the proposed models a vehicle is able to successfully merge into a platoon that is driving on the highway. Finally, the conclusions are given in chapter 6.

2. The platoon

Before including the merging in any model, we will first focus on the platoon. A platoon consists of multiple automated vehicles driving in a group together on the road. The first vehicle of the platoon, or the leader of the platoon, sets the pace for the platoon. The remaining vehicles follow in a line behind the first vehicle, while each vehicle is trying to maintain a certain distance to its direct neighbors. This is usually done based on signals that each vehicle can measure and on information received from others, often their direct neighbors. Vehicles exchange local information only, which could mean that only the first vehicle of the platoon knows the desired velocity of the whole platoon or that, when a vehicle in the front of the platoon suddenly brakes, a vehicle further to the back does not know until the vehicle directly in front of it starts to brake. Vehicles do not need to be fully autonomous in order to drive in a platoon, but they need at least to be automated in the longitudinal direction. This means that any accelerating or braking is done fully automatically, while a human driver might still be needed to steer the vehicle, though also this lateral motion could be controlled. Nevertheless, in this thesis only the longitudinal motion of vehicles is considered.

The outline of this chapter is as follows. Section 2.1 starts by explaining the mathematical description of a platoon. The signals that are available to each vehicle in the platoon are listed in section 2.2. Section 2.3 states a model that can be used for one vehicle. Based on this information and the desired distance between two vehicles defined in section 2.4, it is possible to define an error and derive the platoon dynamics, which is done in section 2.5. Section 2.6 introduces a controller such that a stable system can be derived that describes the complete platoon. Next, in section 2.7 the communication between the vehicles of the platoon is discussed and a new error is introduced that takes into account both direct neighbors of a vehicle. With this general error, section 2.8 derives, in the same way as before, a stable system that models the complete platoon. And finally, in section 2.9, it is proven that this system is asymptotically stable when certain constraints are met.

2.1 Structure of a platoon

To model a platoon, first of all, it is necessary that we are able to clearly describe the platoon that is to be considered. The structure of the platoon can be represented by a directed graph consisting of a node set $\mathcal{V}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Suppose that the platoon consists of n vehicles, then $\mathcal{V} = \{1, \ldots, n\}$ is the set of all vehicles in the platoon. Vehicle i is defined to be the i-th vehicle from the front of the platoon. The communication between vehicles is represented by the edges: if edge $(j, i) \in \mathcal{E}$, then vehicle j is able to communicate to vehicle i. All vehicles that communicate to vehicle i form the neighboring set $\mathcal{N}_i$ of vehicle i, that is, $j \in \mathcal{N}_i$ whenever $(j, i) \in \mathcal{E}$. The communication topology is also given by the adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$, for which holds that

$$a_{ij} = \begin{cases} 1 & (j,i) \in \mathcal{E} \quad \text{(or } j \in \mathcal{N}_i) \\ 0 & \text{otherwise.} \end{cases} \tag{2.1.1}$$

Thus, if $a_{ij} = 1$, vehicle i receives information from vehicle j. Vehicles do not communicate to themselves, so the graph does not contain any edges (i, i) and all diagonal elements a_{ii} are equal to zero. The adjacency matrix can be used to assign weights to the communications links. In this case, any element $a_{ij} > 0$ of adjacency matrix $\mathcal{A}$ represents a communication link from vehicle j to vehicle i with weight a_{ij}.

Another representation of the communication topology is the Laplacian matrix $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{n \times n}$, which is defined as

$$l_{ij} := \begin{cases} -a_{ij} & i \neq j, \\ \sum_{k=1}^{n} a_{ik} & i = j. \end{cases} \tag{2.1.2}$$

The diagonal elements l_{ii} of the Laplacian matrix give an indication of how much information a vehicle i receives: when all nonzero elements a_{ij} are equal to 1, as in (2.1.1), l_{ii} equals the number of neighbors of vehicle i or the number of incoming communication links of vehicle i. The neighbors of vehicle i are then all the vehicles j for which $l_{ij} = -1$.

Clearly, for the Laplacian matrix we have that $l_{ij} \leq 0$ for $i \neq j$ and $l_{ij} \geq 0$ for $i = j$. From the definition in (2.1.2) and the fact that $a_{ii} = 0$ for all vehicles, it follows that the Laplacian matrix also satisfies

$$\sum_{j=1}^{n} l_{ij} = 0,$$

and hence has zero-sum rows. As a consequence, zero is an eigenvalue of $\mathcal{L}$ with corresponding eigenvector $[1, 1, \ldots, 1]^T$. Furthermore, the Laplacian matrix is diagonally dominant, meaning that

$$|l_{ii}| \geq \sum_{j=1, j \neq i}^{n} |l_{ij}| \quad \forall i = 1, \ldots, n.$$

And all eigenvalues of the Laplacian matrix $\mathcal{L}$ have nonnegative real parts.

Consider, for example, the platoon of figure 2.1, where each vehicle communicates to the vehicle directly behind. The corresponding adjacency matrix is given by

$$\mathcal{A} = \begin{pmatrix} 0 & \ldots & \ldots & \ldots & 0 \\ 1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & 1 & 0 \end{pmatrix} \tag{2.1.3}$$

and from the definition in (2.1.2) it follows that the Laplacian matrix is given by

$$\mathcal{L} = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ -1 & 1 & 0 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 1 & 0 \\ 0 & \ldots & 0 & -1 & 1 \end{pmatrix}. \tag{2.1.4}$$

Note that each vehicle now receives information from the vehicle directly in front, except for the first vehicle of the platoon. This would mean that, later on in this chapter, we will have to design separate controllers for the first vehicle of the platoon and for the remaining vehicles. Instead, however, it is possible to introduce a virtual reference vehicle that will be defined to drive in front of the first vehicle of the platoon and that will act as leader of the platoon (which will be done in section 2.3). This will also allow us to define the distance d_i of vehicle i to vehicle $i-1$ for all vehicles in the platoon including the first vehicle. With a virtual reference vehicle, the same dynamics apply to all vehicles in the platoon and the same controller can be used.

The virtual reference vehicle, denoted by the index $i = 0$, and the communication links to the vehicles in the platoon can be included in the graph representation by adding a node 0 and any necessary edges $(0, i)$ to the graph. This results in a new node set $\mathcal{V}_0 = \mathcal{V} \cup \{0\}$ and edge set $\mathcal{E}_0 \subseteq \mathcal{V}_0 \times \mathcal{V}$. Only communication from the virtual vehicle to the platoon, represented by edges $(0, i)$, is considered as our main interest is the platoon. Also, the virtual car as described in section 2.3 is assumed to be uncontrolled and does not require any communication from the platoon to the virtual reference vehicle.

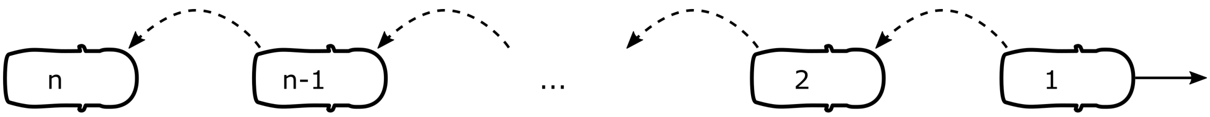


Figure 2.1: Schematic drawing of a platoon with n vehicles, where each vehicle communicates to the vehicle behind (depicted by the dashed arrows).

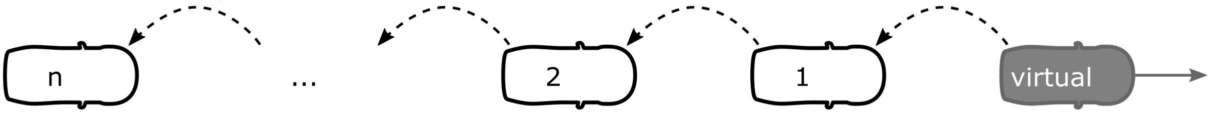


Figure 2.2: Schematic drawing of a platoon with n vehicles and a virtual reference vehicle that drives in front of the first vehicle, where each vehicle communicates to the vehicle behind (depicted by the dashed arrows).

The adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$ of the platoon does not change when the virtual vehicle is added: the connections within the platoon (which does not include the virtual car) are the same as before. Only the total number of communication links to a vehicle in the platoon, which is represented by the diagonal elements of the Laplacian matrix, might change. Therefore, we define a_{i0} for all $i \in \mathcal{V}$, similar to the elements a_{ij} of the adjacency matrix given in (2.1.1), as

$$a_{i0} := \begin{cases} 1 & (0,i) \in \mathcal{E}_0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1.5}$$

Then it is possible to define a new Laplacian matrix $\hat{\mathcal{L}} = [\hat{l}_{ij}] \in \mathbb{R}^{n\times n}$ for the platoon as

$$\hat{l}_{ij} := \begin{cases} -a_{ij} & i \neq j, \\ \sum_{k=0}^{n} a_{ik} & i = j, \end{cases} \tag{2.1.6}$$

where, compared to (2.1.2), the sum now starts at $k = 0$ instead of $k = 1$ in order to include the communication links from the virtual leader to the platoon. For the matrix $\hat{\mathcal{L}}$ we again have that $\hat{l}_{ij} \leq 0$ for $i \neq j$ and $\hat{l}_{ij} \geq 0$ for $i = j$. All eigenvalues of $\hat{\mathcal{L}}$ have nonnegative real parts, and $\hat{\mathcal{L}}$ is diagonally dominant.

The matrix $\hat{\mathcal{L}}$, however, can be split into the Laplacian matrix $\mathcal{L}$, which was defined in (2.1.2), and the pinning matrix $\mathcal{P}$ such that $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$. The pinning matrix $\mathcal{P} \in \mathbb{R}^{n\times n}$ is a diagonal matrix,

$$\mathcal{P} = \begin{pmatrix} p_1 & 0 & \dots & 0 \\ 0 & p_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & p_n \end{pmatrix},$$

where

$$p_i := \begin{cases} 1 & (0,i) \in \mathcal{E}_0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1.7}$$

As (2.1.5) and (2.1.7) show that $p_i = a_{i0}$, it is clear that adding the pinning matrix (2.1.7) to the Laplacian matrix of (2.1.2) results in the new Laplacian matrix in (2.1.6). The pinning matrix $\mathcal{P}$ represents how the virtual reference vehicle communicates to each of the vehicles in the platoon. For the diagonal entries of $\mathcal{P}$ it holds that $p_i = 1$ if vehicle i receives information from the virtual leader. In that case, we say that vehicle i is pinned to the leader.

Consider, for example, the platoon of figure 2.2, where each vehicle receives information from the vehicle directly in front. Since this is the same as the platoon of figure 2.1 but with an added virtual reference vehicle in front of the first vehicle, the adjacency matrix is again given by (2.1.3) and the Laplacian matrix by (2.1.4). The virtual reference vehicle communicates to the first vehicle of the platoon

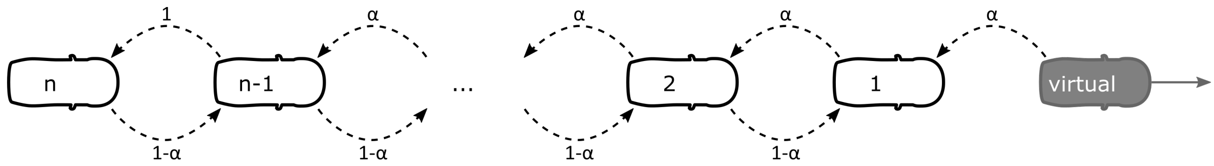


Figure 2.3: Schematic drawing of a platoon with n vehicles and a virtual reference vehicle that drives in front of the first vehicle. Each vehicle receives information from the vehicle directly in front with weight α and from the vehicle directly behind with weight $1-\alpha$ except for vehicle n which only receives information from vehicle $n-1$. The connections between vehicles are depicted by the dashed arrows.

and hence the pinning matrix is given by

$$\mathcal{P} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}. \tag{2.1.8}$$

In the remainder of this chapter, we will consider the platoon of figure 2.3. This platoon consists of n vehicles, where the first vehicle follows a virtual reference vehicle. Each vehicle i of the platoon receives information from vehicle $i-1$ with weight α, where $\alpha \in [0,1]$, and from vehicle $i+1$ with weight $1-\alpha$, except for vehicle n that only receives information from vehicle $n-1$ with weight 1. The weights determine whether the information received from the vehicle in front or from the vehicle behind is considered to be more important. Using a weight 1 for vehicle n will allow us further on to use the same equations for all vehicles of the platoon: for vehicle n we simply set $\alpha = 1$.

The adjacency matrix of the platoon in figure 2.3 is given by

$$\mathcal{A} = \begin{pmatrix} 0 & 1-\alpha & 0 & \dots & 0 \\ \alpha & 0 & 1-\alpha & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \alpha & 0 & 1-\alpha \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} \tag{2.1.9}$$

and hence, by definitions (2.1.2) and (2.1.7), the Laplacian matrix and the pinning matrix are given by

$$\mathcal{L} = \begin{pmatrix} 1-\alpha & -(1-\alpha) & 0 & \dots & 0 \\ -\alpha & 1 & -(1-\alpha) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\alpha & 1 & -(1-\alpha) \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix} \tag{2.1.10}$$

and

$$\mathcal{P} = \begin{pmatrix} \alpha & 0 & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}. \tag{2.1.11}$$

Note that for $\alpha = 1$, (2.1.10) and (2.1.11) are the same as (2.1.4) and (2.1.8).

2.2 Measurements

Each vehicle of the platoon decides based on the available signals whether it should accelerate or decelerate in order to keep its neighbors at the desired distance. These signals can be obtained through

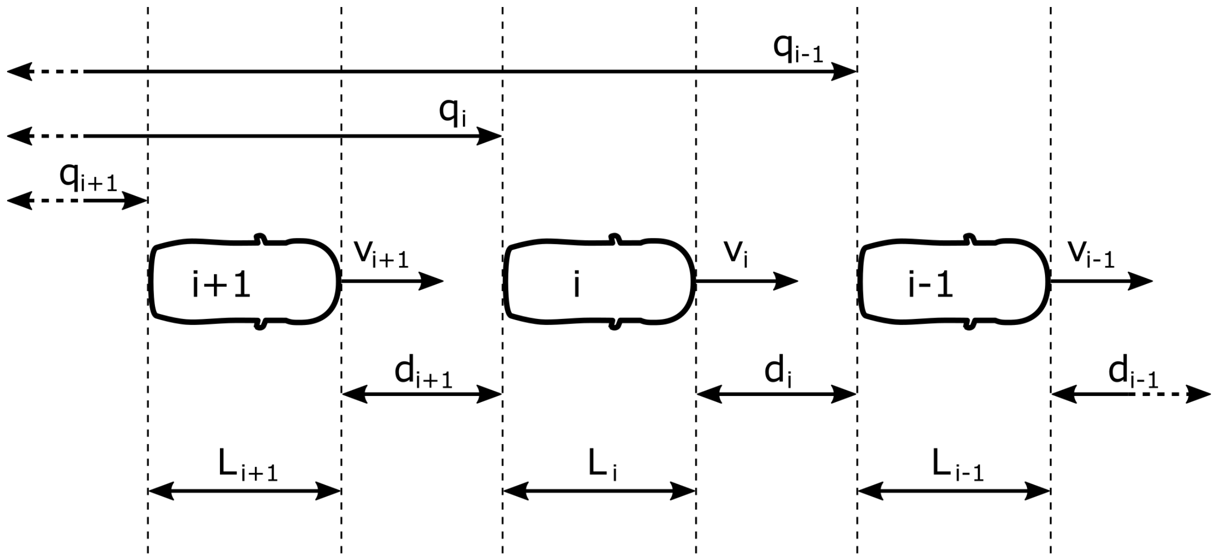


Figure 2.4: Schematic drawing of a part of a platoon. The position q, velocity v, distance d to its predecessor, and length L of each vehicle are depicted by the arrows. The subscripts of the variables indicate to which vehicle an arrow applies.

camera, radar, and lidar systems or through wireless communication with other vehicles in the platoon. When modeling a platoon it is important to take into consideration that some signals might be difficult to measure, might be inaccurate, or might not be available at all. It is assumed here that each car i in the platoon is able to measure

- the distance $d_i(t)$ to vehicle $i-1$ directly in front of vehicle i and the distance $d_{i+1}(t)$ to vehicle $i+1$ directly behind vehicle i,
- the relative velocities $v_i(t) - v_{i-1}(t)$ and $v_i(t) - v_{i+1}(t)$ with respect to vehicle $i-1$ directly in front and vehicle $i+1$ directly behind vehicle i,
- the velocity $v_i(t)$,
- and the acceleration $a_i(t)$.

The distance and relative velocities can be measured using radar and lidar systems, while the velocity and acceleration can be obtained from on-board sensors. The actual position $q_i(t)$ of vehicle i is assumed to be unknown since GPS data might be inaccurate, but could be derived from the definition of the distance $d_i(t)$ between vehicles i and $i-1$,

$$d_i(t) := q_{i-1}(t) - q_i(t) - L_i \quad \forall i \in \mathcal{V} \setminus \{1\}, \tag{2.2.1}$$

if the position of one vehicle is given and also the length L_i of each vehicle is known. The position $q_i(t)$ is defined as the position of the most rear part of the car. Some of these variables are illustrated in figure 2.4.

2.3 Vehicle dynamics

Based on the assumptions of section 2.2 that for each vehicle i the distance $d_i(t)$ to the car in front, the velocity $v_i(t)$ and the acceleration $a_i(t)$ are available, we define the vehicle state of vehicle i as

$$\begin{pmatrix} d_i(t) \\ v_i(t) \\ a_i(t) \end{pmatrix}.$$

The longitudinal motion of each vehicle of the platoon can then be modeled as

(vehicle dynamics)

$$\begin{pmatrix} \dot{d}_i(t) \\ \dot{v}_i(t) \\ \dot{a}_i(t) \end{pmatrix} = \begin{pmatrix} v_{i-1}(t) - v_i(t) \\ a_i(t) \\ -\frac{1}{\tau}a_i(t) + \frac{1}{\tau}u_i(t-\phi) \end{pmatrix}, \tag{2.3.1}$$

where τ is the drive-line dynamics time constant, ϕ is the drive-line dynamics time delay, and $u_i(t)$ is the input signal [10, 21, 22]. Since the engine of a vehicle reacts to driver inputs according to certain dynamics, the input signal $u_i(t)$ is not a direct force. Instead it can be interpreted as the desired acceleration of vehicle i since the acceleration $a_i(t)$ of vehicle i will eventually follow $u_i(t)$, though with a delay ϕ. The engine characteristics are represented by τ and ϕ. Assuming that the platoon is homogeneous, τ and ϕ are the same for all vehicles of the platoon. For now, it is assumed that $\phi = 0$. The dynamics in (2.3.1) are derived from nonlinear vehicles dynamics [15, 16], but still provide a more realistic model for the longitudinal behavior of a vehicle than pure integrator dynamics and it has been shown that these dynamics are suitable for modeling the longitudinal motion of vehicles [6, 10, 21].

Note that for the first vehicle of the platoon the distance $d_1(t)$ is not defined. According to the definition in (2.2.1), the distance $d_1(t)$ depends on the position $q_0(t)$ of some vehicle 0, while it follows from (2.3.1) that its derivative $\dot{d}_1(t)$ depends on velocity $v_0(t)$. Therefore, a virtual reference vehicle with index $i = 0$ is introduced with vehicle dynamics

$$\begin{pmatrix} \dot{q}_0(t) \\ \dot{v}_0(t) \\ \dot{a}_0(t) \end{pmatrix} = \begin{pmatrix} v_0(t) \\ a_0(t) \\ -\frac{1}{\tau}a_0(t) + \frac{1}{\tau}u_0(t) \end{pmatrix}. \tag{2.3.2}$$

These dynamics are the same as (2.3.1) with $\phi = 0$, except that the vehicle state now includes the position $q_0(t)$ of the virtual car instead of the distance $d_0(t)$, which is not defined for the virtual reference vehicle. The virtual car is assumed to be uncontrolled, cruising at a constant velocity $\bar{v}_0$.

2.4 Desired distance between vehicles

The platoon is formed by controlling the distances $d_i(t)$ between the vehicles. A commonly used velocity-dependent spacing policy defining the desired distance $d_i^d(t)$ between vehicles i and $i-1$ is

$$d_i^d(t) = r + hv_i(t), \tag{2.4.1}$$

which consists of a constant term r, specifying the standstill distance between cars, and the time-dependent part $hv_i(t)$ with time gap h and velocity $v_i(t)$ of vehicle i [21, 22]. This spacing policy is known to increase road safety and platoon stability [6, 10, 11, 21, 22]. Ideally, the values of h and r are as small as possible while still guaranteeing the safety of passengers. For homogeneous platoons, r and h are the same for all vehicles in the platoon.

Instead of controlling the distances $d_i(t)$ between vehicles directly, we define an error $e_i(t)$ for each vehicle i as

$$\begin{aligned} e_i(t) &= d_i(t) - d_i^d(t) \\ &= d_i(t) - r - hv_i(t), \end{aligned} \tag{2.4.2}$$

which represents the difference between the actual distance to the predecessor of vehicle i and the desired distance. Since a virtual reference vehicle was introduced in (2.3.2), this error is defined for all vehicles including the first vehicle, and hence it will be possible to control all vehicles in the same way. The goal is to design a controller such that the error $e_i(t)$ will converge to zero for all vehicles i. When the errors $e_i(t)$ in (2.4.2) are all equal to zero, the distances between vehicles i and $i-1$ will be equal to the desired distances $d_i^d(t)$ defined in (2.4.1) for all $i \in \mathcal{V}$. Note, however, that it is not necessary that the virtual error of the first vehicle goes to zero, only $\dot{e}_1(t)$ needs to converge to zero, i.e., the first car should reach the same speed as the reference vehicle. Nevertheless, it is assumed that also $e_1(t)$ should converge to zero such that the same controller can be used for the whole platoon.

Furthermore, in addition to controlling the distances between vehicles, the velocity $v_{\text{platoon}}(t)$ of the platoon should be controlled such that it follows the desired platoon velocity $v^d_{\text{platoon}}(t)$. As each vehicle follows behind the leading vehicle, in this case the virtual reference vehicle, the platoon velocity $v_{\text{platoon}}(t)$ is defined to be the velocity of the leading vehicle:

$$v_{\text{platoon}}(t) \coloneqq v_0(t).$$

Because the platoon is supposed to maintain its formation while driving, the velocity $v_i(t)$ of each vehicle i should also be equal to the desired platoon velocity $v^d_{\text{platoon}}(t)$. To summarize, the control objective of the platoon is that all vehicles follow the desired platoon velocity $v^d_{\text{platoon}}(t)$ and that the distances between vehicles match the spacing policy defined in (2.4.1). This objective can also be formulated as

$$\begin{aligned}
&\lim_{t\to\infty} e_i(t) = 0 \quad \forall i \in \mathcal{V},\\
&\lim_{t\to\infty} v_i(t) = v^d_{\text{platoon}}(t) \quad \forall i \in \mathcal{V}.
\end{aligned}$$

For now, it is assumed that the desired velocity of the platoon is constant: $v^d_{\text{platoon}}(t) = \bar{v}_{\text{platoon}}$. As mentioned in section 2.3, it is also assumed that the virtual car is uncontrolled and driving at a constant velocity $\bar{v}_0$. With this virtual reference vehicle, when all errors $e_i(t)$ defined in (2.4.2) converge to zero, all vehicles will automatically be driving at the same constant velocity as the virtual reference car. Therefore, by choosing the initial state of the virtual car as $[q_0(0),\, \bar{v}_{\text{platoon}},\, 0]^T$, the reference vehicle will have velocity $v_0(t) = \bar{v}_{\text{platoon}}$, and all vehicles will eventually reach the same velocity.

2.5 Error dynamics

Consider a platoon of n vehicles, which are described by the vehicle dynamics of (2.3.1), with a communication topology described by the adjacency matrix $\mathcal{A}$ in (2.1.9), the Laplacian matrix $\mathcal{L}$ of (2.1.10) and the pinning matrix $\mathcal{P}$ of (2.1.11) (see also figure 2.3). The virtual reference vehicle is given by (2.3.2). The drive-line dynamics time constant is denoted by τ and the drive-line dynamics time delay ϕ is assumed to be equal to zero (as was mentioned in section 2.3). For a vehicle i, the error $e_i(t)$ is defined in (2.4.2) as

$$e_i = d_i - r - hv_i, \tag{2.5.1}$$

where r is the standstill distance and h is the time gap. With this error we can now derive the error dynamics of the platoon. Time arguments are dropped to improve readability.

Using the expressions for the time derivatives of distance d_i, velocity v_i, and acceleration a_i given by (2.3.1), the time derivatives of e_i can be derived as

$$\begin{aligned}
\dot{e}_i &= v_{i-1} - v_i - ha_i, &&(2.5.2)\\
\ddot{e}_i &= a_{i-1} - a_i - h\left(-\frac{1}{\tau}a_i + \frac{1}{\tau}u_i\right)\\
&= a_{i-1} + \frac{h-\tau}{\tau}a_i - \frac{h}{\tau}u_i, &&(2.5.3)
\end{aligned}$$

and

$$\begin{aligned}
\dddot{e}_i &= \left(-\frac{1}{\tau}a_{i-1} + \frac{1}{\tau}u_{i-1}\right) + \frac{h-\tau}{\tau}\left(-\frac{1}{\tau}a_i + \frac{1}{\tau}u_i\right) - \frac{h}{\tau}\dot{u}_i\\
&= -\frac{1}{\tau}\left(a_{i-1} + \frac{h-\tau}{\tau}a_i - \frac{h}{\tau}u_i\right) + \frac{1}{\tau}u_{i-1} - \frac{1}{\tau}u_i - \frac{h}{\tau}\dot{u}_i
\end{aligned}$$

which becomes

$$= -\frac{1}{\tau}\ddot{e}_i + \frac{1}{\tau}\left(u_{i-1} - u_i - h\dot{u}_i\right)$$

by substituting the expression for $\ddot{e}_i$ given in (2.5.3) and, finally,

$$= -\frac{1}{\tau}\ddot{e}_i + \frac{1}{\tau}\bar{u}_i \tag{2.5.4}$$

after defining a new input $\bar{u}_i$ as

$$\bar{u}_i := u_{i-1} - u_i - h\dot{u}_i.$$

This new input $\bar{u}_i$ can be rewritten as the filter

$$\dot{u}_i = -\frac{1}{h}u_i + \frac{1}{h}(u_{i-1} - \bar{u}_i), \tag{2.5.5}$$

which determines the desired acceleration u_i for each vehicle i based on the new input $\bar{u}_i$ and the desired acceleration u_{i-1} of vehicle $i-1$.

Next, we define the error state x_i of vehicle i as

$$x_i := \begin{pmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{pmatrix}. \tag{2.5.6}$$

Combining this error state x_i with the expression for $\dddot{e}_i$ given in (2.5.4) results in

(error dynamics)

$$\dot{x}_i = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix}}_{A} x_i + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix}}_{B} \bar{u}_i. \tag{2.5.7}$$

These error dynamics can also be written as

$$\dot{x}_i = Ax_i + B\bar{u}_i$$

with matrices A and B defined in (2.5.7).

The error dynamics in (2.5.7) and the filter in (2.5.5) are defined for all vehicles in the platoon as a result of the introduction of a virtual reference vehicle in section 2.3. Therefore, the dynamics of the complete platoon are given by

(platoon dynamics)

$$\begin{pmatrix} \dot{x}_i \\ \dot{u}_i \end{pmatrix} = \begin{pmatrix} A & \mathrm{O}_{3\times 1} \\ \mathrm{O}_{1\times 3} & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} x_i \\ u_i \end{pmatrix} + \begin{pmatrix} B \\ -\frac{1}{h} \end{pmatrix} \bar{u}_i + \begin{pmatrix} \mathrm{O}_{3\times 1} \\ \frac{1}{h} \end{pmatrix} u_{i-1} \quad \forall i \in \mathcal{V}, \tag{2.5.8}$$

where $\mathrm{O}_{m\times n}$ denotes a zero matrix of size $m \times n$.

Note that to retrieve the values of the vehicle state $[d_i,\, v_i,\, a_i]^T$ from the error state $x_i = [e_i,\, \dot{e}_i,\, \ddot{e}_i]^T$ and the input u_i, (2.5.1) to (2.5.3) can be rewritten as

$$\begin{cases} a_i &= \frac{\tau}{h-\tau}(\ddot{e}_i - a_{i-1}) + \frac{h}{h-\tau}u_i, \\ v_i &= -\dot{e}_i + v_{i-1} - ha_i, \\ d_i &= e_i + r + hv_i, \end{cases}$$

given that a_0 and v_0 are known.

2.6 Closed-loop platoon dynamics

Continuing with the situation of section 2.5, in this section we will define a controller for the input $\bar{u}_i$ in the platoon dynamics given by (2.5.8) and derive the closed-loop platoon dynamics.

Consider the following distributed controller for the input $\bar{u}_i$ of (2.5.8):

(controller)

$$\bar{u}_i = -\sum_{j \in \mathcal{V}} \left[a_{ij} k^T (x_i - x_j) \right] - p_i k^T x_i, \tag{2.6.1}$$

where

$$k^T := [k_1,\, k_2,\, k_3] \tag{2.6.2}$$

is the controller gain vector, a_{ij} is an element of the adjacency matrix $\mathcal{A}$ and p_i is the (i,i)-th entry of the pinning matrix $\mathcal{P}$ [21, 22]. The controller given in (2.6.1) can be divided into the distributed part $-\sum_{j\in\mathcal{V}} \left[a_{ij}k^T(x_i - x_j)\right]$ and the pinning constraint $-p_i k^T x_i$. The pinning constraint is added to ensure that the state converges to zero [21, 22]. The controller gains k^T are the same for the distributed term and the pinning constraint, but could be designed differently [21].

Note that in (2.6.1) the controller gain k_3 of the vector k^T in (2.6.2) is associated with the $\ddot{e}_i$ element of the error state vector x_i in (2.5.6), and that $\ddot{e}_i$ in turn depends on $a_{i-1} - a_i$ for each vehicle i (see (2.5.3)). However, in section 2.2, it was assumed that vehicle i is able to measure the acceleration a_i, but not a_{i-1} or $a_{i-1} - a_i$. For this reason, the controller gain k_3 is set to zero for all controllers.

Substituting the control law of (2.6.1) into the platoon dynamics of (2.5.8) gives

$$\begin{aligned}
\begin{pmatrix} \dot{x}_i \\ \dot{u}_i \end{pmatrix} &= \begin{pmatrix} A & \mathrm{O}_{3\times1} \\ \mathrm{O}_{1\times3} & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} x_i \\ u_i \end{pmatrix} + \begin{pmatrix} B \\ -\frac{1}{h} \end{pmatrix} \left(-\sum_{j\in\mathcal{V}} \left[a_{ij}k^T(x_i - x_j)\right] - p_i k^T x_i \right) + \begin{pmatrix} \mathrm{O}_{3\times1} \\ \frac{1}{h} \end{pmatrix} u_{i-1} \\
&= \begin{pmatrix} A - p_i B k^T & \mathrm{O}_{3\times1} \\ p_i \frac{1}{h} k^T & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} x_i \\ u_i \end{pmatrix} + \begin{pmatrix} B \\ -\frac{1}{h} \end{pmatrix} \left(-\sum_{j\in\mathcal{V}} l_{ij} k^T x_j \right) + \begin{pmatrix} \mathrm{O}_{3\times1} \\ \frac{1}{h} \end{pmatrix} u_{i-1},
\end{aligned} \tag{2.6.3}$$

where the pinning constraint $-p_i k^T x_i$ of the controller is moved to the first term and the elements a_{ij} of the adjacency matrix $\mathcal{A}$ are replaced by the elements l_{ij} of the Laplacian matrix $\mathcal{L}$, for which, by definition (2.1.2), holds that $l_{ij} = -a_{ij}$ for $i \neq j$ and $l_{ij} = \sum_{k=1}^{n} a_{ik}$ for $i = j$.

Next, we introduce the lumped error state $X^T := \left[x_1^T, \ldots, x_n^T\right]$ and the lumped input state $U :=$

$[u_1, \ldots, u_n]^T$ such that (2.6.3) can be written as

$$
\begin{pmatrix} \dot{X} \\ \dot{U} \end{pmatrix} = \left(\begin{array}{cccc|cccc}
A - p_1 B k^T & \mathrm{O}_{3\times 3} & \ldots & \mathrm{O}_{3\times 3} & & & & \\
\mathrm{O}_{3\times 3} & \ddots & \ddots & \vdots & & \mathrm{O}_{3n\times n} & & \\
\vdots & \ddots & \ddots & \mathrm{O}_{3\times 3} & & & & \\
\mathrm{O}_{3\times 3} & \ldots & \mathrm{O}_{3\times 3} & A - p_n B k^T & & & & \\
\hline
p_1 \frac{1}{h} k^T & \mathrm{O}_{1\times 3} & \ldots & \mathrm{O}_{1\times 3} & -\frac{1}{h} & 0 & \ldots & 0 \\
\mathrm{O}_{1\times 3} & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \mathrm{O}_{1\times 3} & \vdots & \ddots & \ddots & 0 \\
\mathrm{O}_{1\times 3} & \ldots & \mathrm{O}_{1\times 3} & p_n \frac{1}{h} k^T & 0 & \ldots & 0 & -\frac{1}{h}
\end{array}\right) \begin{pmatrix} X \\ U \end{pmatrix}
$$

$$
+ \left(\begin{array}{cccc|c}
-l_{11} B k^T & -l_{12} B k^T & \ldots & -l_{1n} B k^T & \\
-l_{21} B k^T & \ddots & \ddots & \vdots & \mathrm{O}_{3n\times n} \\
\vdots & \ddots & \ddots & -l_{(n-1)n} B k^T & \\
-l_{n1} B k^T & \ldots & -l_{n(n-1)} B k^T & -l_{nn} B k^T & \\
\hline
l_{11} \frac{1}{h} k^T & l_{12} \frac{1}{h} k^T & \ldots & l_{1n} \frac{1}{h} k^T & \\
l_{21} \frac{1}{h} k^T & \ddots & \ddots & \vdots & \mathrm{O}_{n\times n} \\
\vdots & \ddots & \ddots & l_{(n-1)n} \frac{1}{h} k^T & \\
l_{n1} \frac{1}{h} k^T & \ldots & l_{n(n-1)} \frac{1}{h} k^T & l_{nn} \frac{1}{h} k^T &
\end{array}\right) \begin{pmatrix} X \\ U \end{pmatrix}
$$

$$
+ \left(\begin{array}{c|ccccc}
\mathrm{O}_{3n\times 3n} & & & \mathrm{O}_{3n\times n} & & \\
\hline
 & 0 & \ldots & \ldots & \ldots & 0 \\
 & \frac{1}{h} & \ddots & & & \vdots \\
\mathrm{O}_{n\times 3n} & 0 & \ddots & \ddots & & \vdots \\
 & \vdots & \ddots & \ddots & \ddots & \vdots \\
 & 0 & \ldots & 0 & \frac{1}{h} & 0
\end{array}\right) \begin{pmatrix} X \\ U \end{pmatrix}
$$

$$
+ \left(\begin{array}{c}
\mathrm{O}_{3n\times 1} \\
\hline
\frac{1}{h} \\
0 \\
\vdots \\
0
\end{array}\right) u_0.
$$

By using the Kronecker product $\otimes$, this can be written more compactly as

$$
\begin{aligned}
\begin{pmatrix} \dot{X} \\ \dot{U} \end{pmatrix} &= \begin{pmatrix} I_n \otimes A - \mathcal{P} \otimes B k^T & \mathrm{O}_{3n\times n} \\ \mathcal{P} \otimes \frac{1}{h} k^T & -\frac{1}{h} I_n \end{pmatrix} \begin{pmatrix} X \\ U \end{pmatrix} \\
&+ \begin{pmatrix} -\mathcal{L} \otimes B k^T & \mathrm{O}_{3n\times n} \\ \mathcal{L} \otimes \frac{1}{h} k^T & \mathrm{O}_{n\times n} \end{pmatrix} \begin{pmatrix} X \\ U \end{pmatrix} \\
&+ \begin{pmatrix} \mathrm{O}_{3n\times 3n} & \mathrm{O}_{3n\times n} \\ \mathrm{O}_{n\times 3n} & \frac{1}{h} I_{(-1),n} \end{pmatrix} \begin{pmatrix} X \\ U \end{pmatrix} \\
&+ \begin{pmatrix} \mathrm{O}_{3n\times 1} \\ B_u \end{pmatrix} u_0,
\end{aligned} \tag{2.6.4}
$$

where I_n is the $n \times n$ identity matrix, $I_{(-1),n}$ is given by

$$
I_{(-1),n} := \begin{pmatrix}
0 & \ldots & \ldots & \ldots & 0 \\
1 & \ddots & & & \vdots \\
0 & \ddots & \ddots & & \vdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & 1 & 0
\end{pmatrix},
$$

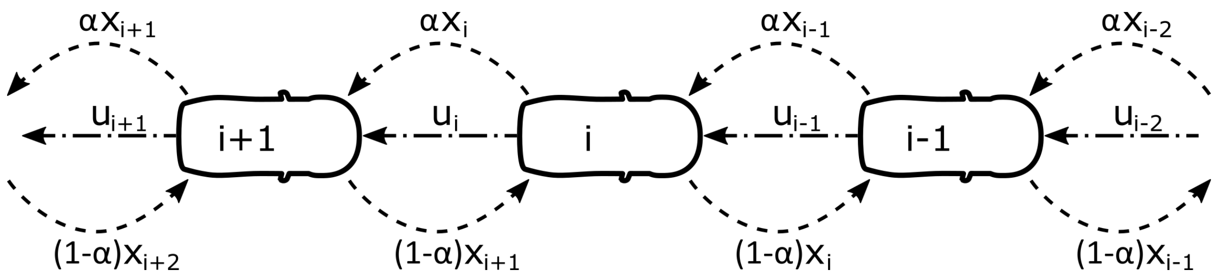


Figure 2.5: Schematic drawing of part of the platoon described by the closed-loop dynamics of (2.6.5) with a communication topology given by Laplacian matrix (2.1.10) and pinning matrix (2.1.11). Each vehicle i receives the weighted error states $(1-\alpha)x_{i+1}$ and αx_{i-1} (dashed arrows) and the desired acceleration u_{i-1} (dash-dotted arrow).

and

$$B_u := \left[\frac{1}{h},\ 0,\ \ldots,\ 0\right]^T.$$

Finally, adding the first three terms of (2.6.4) together and using $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$ (see section 2.1) results in the closed-loop platoon dynamics

(closed-loop platoon dynamics)

$$\begin{pmatrix} \dot{X} \\ \dot{U} \end{pmatrix} = \begin{pmatrix} I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T & \mathrm{O}_{3n\times n} \\ \hat{\mathcal{L}} \otimes \frac{1}{h}k^T & -\frac{1}{h}I_n + \frac{1}{h}I_{(-1),n} \end{pmatrix} \begin{pmatrix} X \\ U \end{pmatrix} + \begin{pmatrix} \mathrm{O}_{3n\times 1} \\ B_u \end{pmatrix} u_0. \tag{2.6.5}$$

Note that the closed-loop system of (2.6.5) only requires u_0, which is the desired acceleration of the virtual reference vehicle, as an external input.

2.7 Communication

The derivation of the closed-loop platoon dynamics of (2.6.5) uses the fact that each vehicle of the platoon is able to communicate with some of the other vehicles in the platoon. The equations in sections 2.5 and 2.6 use, in addition to the signals that each vehicle can measure with on-board sensors described in section 2.2, also some signals that must be provided by its neighbors. This includes the desired acceleration u_{i-1} of vehicle $i-1$ that should be communicated from vehicle $i-1$ to vehicle i, as follows from the platoon dynamics in (2.5.8). Furthermore, the controller in (2.6.1) shows that vehicle i requires the error states x_j of all its neighbors $j \in \mathcal{N}_i$ since $a_{ij} > 0$ whenever $j \in \mathcal{N}_i$. Considering a platoon that is described by the closed-loop dynamics of (2.6.5) with a communication topology given by Laplacian matrix (2.1.10) and pinning matrix (2.1.11), each vehicle i thus communicates the input signal u_i to vehicle $i+1$ with weight 1 and the error state x_i to vehicle $i+1$ with weight α and to vehicle $i-1$ with weight $1-\alpha$. This is illustrated by figure 2.5.

Which vehicles communicate their states to some vehicle i is completely determined by the communication topology. To change this in the closed-loop platoon dynamics of (2.6.5) or to change the weights that are used only the matrix $\hat{\mathcal{L}}$ needs to be adapted. Changing which input signals are communicated to vehicle i, however, requires a different error to be used. This will become apparent in section 2.8, where the closed-loop platoon dynamics are derived for a different error.

One error that could be used instead of the error defined in (2.4.2) is given by

$$\begin{aligned} e_i(t) &= \beta\left[d_i(t) - d_i^d(t)\right] + (1-\beta)\left[d_{i+1}(t) - d_{i+1}^d(t)\right] \\ &= \beta\left[d_i(t) - r - hv_i(t)\right] + (1-\beta)\left[d_{i+1}(t) - r - hv_{i+1}(t)\right] \end{aligned} \tag{2.7.1}$$

with weight $\beta \in [0,1]$. Each term of error (2.7.1) computes the difference between the actual distance between two vehicles and the desired distance. Recall that the distance d_i was defined as the distance

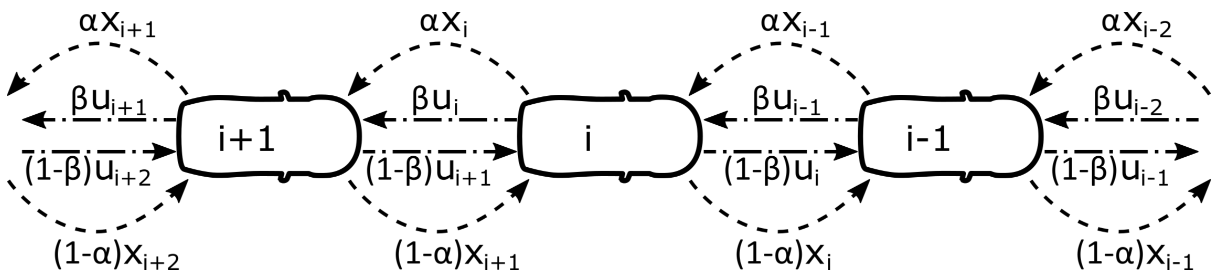


Figure 2.6: Schematic drawing of part of the platoon described by the closed-loop dynamics of (2.8.7) with a communication topology given by Laplacian matrix (2.1.10) and pinning matrix (2.1.11). Each vehicle i receives the weighted error states $(1-\alpha)x_{i+1}$ and αx_{i-1} (dashed arrows) and the weighted desired accelerations βu_{i-1} and $(1-\beta)u_{i+1}$ (dash-dotted arrows).

between vehicles i and $i-1$ and that the desired distance d_i^d between vehicles i and $i-1$ was defined in (2.4.1) as

$$d_i^d(t) = r + hv_i(t)$$

with standstill distance r and time gap h. Similar to the weights α and $1-\alpha$ that were used in the communication topology of figure 2.3, error (2.7.1) employs weights to control how much each of the neighbors of a vehicle contributes to the total error. The weight β in the first term applies to the error in the distance d_i between vehicles i and $i-1$ and the weight $\beta-1$ in the second term to the error in the distance d_{i+1} between vehicles i and $i+1$. This allows error (2.7.1) to take into account both the preceding vehicle and the vehicle behind, whereas the error defined in (2.4.2) only considers the preceding vehicle.

Instead of error (2.7.1), however, we will use a slightly different error in the remainder of this thesis. This error is defined as

$$\begin{aligned} e_i(t) &= \beta\left[d_i(t) - d_i^d(t)\right] + (1-\beta)\left[d_{i+1}(t) - d_i^d(t)\right] \\ &= \beta\left[d_i(t) - r - hv_i(t)\right] + (1-\beta)\left[d_{i+1}(t) - r - hv_i(t)\right] \end{aligned} \tag{2.7.2}$$

with $\beta \in [0,1]$. This error is the same as error (2.7.1) except that d_{i+1}^d in the second term is replaced by d_i^d. Here, the desired distance d_i^d should be interpreted as the distance that vehicle i requires between itself and all its direct neighbors, and not just to the vehicle in front. This means that the desired distance d_i^d should be compared to both the actual distance to the preceding vehicle and to the vehicle behind, which then results in error (2.7.2). Using this error has the advantage that the dynamics that will be derived in section 2.8 are slightly more convenient than if the same were done with error (2.7.1). Furthermore, in contrast to error (2.7.1), error (2.7.2) does not include v_{i+1}, which cannot be measured directly by vehicle i, as was mentioned in section 2.2. Velocity v_{i+1} could be derived from v_i and $v_i - v_{i+1}$, but might be inaccurate. Note that, for $\beta = 1$, error (2.7.2) is the same as the error in (2.4.2). Also note that the error in (2.7.2) is not defined for the first and last vehicle of a platoon. This can be solved for the first vehicle by using the virtual reference vehicle described by (2.3.2) and for the last vehicle by setting $\beta = 1$ (which is the same as using (2.4.2) instead of (2.7.2)).

As will be shown in section 2.8, it turns out that, when using error (2.7.2), the weights β and $1-\beta$ should be applied to the communicated input signals u_{i-1} and u_{i+1} respectively. This results in the situation of figure 2.6, where vehicle i receives the weighted error states αx_{i-1} and $(1-\alpha)x_{i+1}$ and the weighted input signals βu_{i-1} and $(1-\beta)u_{i+1}$. Note that figure 2.5 shows the case where $\beta = 1$. It is possible to choose the same value for the weights α and β, i.e. $\alpha = \beta$, such that the same weight is applied to all communication that occurs between any two vehicles i and $i-1$. However, this is not necessary. Weight α has a slightly different interpretation than weight β. Weight α is applied mostly to communicated signals. It is used in controller (2.6.1) and decides how much influence each communicated error state has on the input signal. Weight β in error (2.7.2) only applies to signals that a vehicle can measure itself. It determines whether the distance to the vehicle in front should count more heavily towards the total error than the distance to the vehicle behind.

2.8 General closed-loop platoon dynamics

Now that we have introduced an error that considers both the vehicle in front and the vehicle behind, which was defined in (2.7.2) as

$$\begin{aligned} e_i &= \beta\left[d_i - d_i^d\right] + (1-\beta)\left[d_{i+1} - d_i^d\right] \\ &= \beta\left[d_i - r - hv_i\right] + (1-\beta)\left[d_{i+1} - r - hv_i\right], \end{aligned} \tag{2.8.1}$$

it is possible to derive the general closed-loop platoon dynamics. The case $\beta = 1$, where error (2.8.1) simplifies to the error in (2.4.2), was considered in sections 2.5 and 2.6. Deriving the closed-loop dynamics for the general error in (2.8.1) is done in a similar manner. Time arguments are dropped to improve readability.

Again, consider a platoon of n vehicles, which are described by the vehicle dynamics of (2.3.1). The drive-line dynamics time delay ϕ is still assumed to be equal to zero. A virtual reference vehicle given by (2.3.2) drives in front of the platoon. For the last vehicle of the platoon we use $\beta = 1$. The communication topology is given by the Laplacian matrix $\mathcal{L}$ of (2.1.10) and the pinning matrix $\mathcal{P}$ of (2.1.11) and is depicted in figure 2.3, though this topology is not explicitly used in the derivation of the dynamics and hence matrices $\mathcal{L}$ and $\mathcal{P}$ could be replaced by any matrices for which $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$ satisfies definition (2.1.6).

Using the vehicle dynamics in (2.3.1), the time derivatives of the error in (2.8.1) are given by

$$\begin{aligned} \dot{e}_i &= \beta\left[v_{i-1} - v_i - ha_i\right] + (1-\beta)\left[v_i - v_{i+1} - ha_i\right], \\ \ddot{e}_i &= \beta\left[a_{i-1} + \frac{h-\tau}{\tau}a_i - \frac{h}{\tau}u_i\right] + (1-\beta)\left[\frac{h+\tau}{\tau}a_i - a_{i+1} - \frac{h}{\tau}u_i\right], \end{aligned}$$

and

$$\begin{aligned} \dddot{e}_i &= \beta\left[-\frac{1}{\tau}a_{i-1} + \frac{1}{\tau}u_{i-1} - \frac{1}{\tau}\left(\frac{h-\tau}{\tau}a_i\right) + \frac{1}{\tau}\left(\frac{h}{\tau}u_i\right) - \frac{1}{\tau}u_i - \frac{h}{\tau}\dot{u}_i\right] \\ &\quad + (1-\beta)\left[-\frac{1}{\tau}\left(\frac{h+\tau}{\tau}\right)a_i + \frac{1}{\tau}\left(\frac{h}{\tau}u_i\right) + \frac{1}{\tau}u_i + \frac{1}{\tau}a_{i+1} - \frac{1}{\tau}u_{i+1} - \frac{h}{\tau}\dot{u}_i\right] \\ &= -\frac{1}{\tau}\left(\beta\left[a_{i-1} + \frac{h-\tau}{\tau}a_i - \frac{h}{\tau}u_i\right] + (1-\beta)\left[\frac{h+\tau}{\tau}a_i - a_{i+1} - \frac{h}{\tau}u_i\right]\right) \\ &\quad + \frac{1}{\tau}\left(\beta\left[u_{i-1} - u_i - h\dot{u}_i\right] + (1-\beta)\left[u_i - u_{i+1} - h\dot{u}_i\right]\right) \\ &= -\frac{1}{\tau}\ddot{e}_i + \frac{1}{\tau}\bar{u}_i, \end{aligned} \tag{2.8.2}$$

where the new input $\bar{u}_i$ is now defined as

$$\bar{u}_i := \beta\left[u_{i-1} - u_i - h\dot{u}_i\right] + (1-\beta)\left[u_i - u_{i+1} - h\dot{u}_i\right]. \tag{2.8.3}$$

Define the error state x_i of vehicle i as

$$x_i := \begin{pmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{pmatrix}.$$

The error dynamics for vehicle i then follow from (2.8.2) as

(error dynamics)

$$\dot{x}_i = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix}}_{A} x_i + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix}}_{B} \bar{u}_i \tag{2.8.4}$$

or

$$\dot{x}_i = Ax_i + B\bar{u}_i,$$

where matrices A and B are defined in (2.8.4).

The new input $\bar{u}_i$ defined in (2.8.3) can be rewritten as

$$\bar{u}_i = \beta u_{i-1} - (2\beta - 1)u_i - (1-\beta)u_{i+1} - h\dot{u}_i,$$

or as the filter

$$\dot{u}_i = -\frac{2\beta - 1}{h}u_i + \frac{1}{h}\left(\beta u_{i-1} - (1-\beta)u_{i+1} - \bar{u}_i\right). \tag{2.8.5}$$

Because of the virtual reference vehicle, the error in (2.8.1), and hence also the error dynamics in (2.8.4) and the filter in (2.8.5), are defined for all vehicles $i \in \mathcal{V} \setminus \{n\}$. For vehicle n, we set $\beta = 1$ such that the second term of (2.8.1) disappears. The error dynamics in (2.8.4) and the filter in (2.8.5) together form the platoon dynamics, which are given by

(platoon dynamics)

$$\begin{pmatrix} \dot{x}_i \\ \dot{u}_i \end{pmatrix} = \begin{cases} \begin{pmatrix} A & \mathrm{O}_{3\times 1} \\ \mathrm{O}_{1\times 3} & -\frac{2\beta-1}{h} \end{pmatrix}\begin{pmatrix} x_i \\ u_i \end{pmatrix} + \begin{pmatrix} B \\ -\frac{1}{h} \end{pmatrix}\bar{u}_i + \begin{pmatrix} \mathrm{O}_{3\times 1} \\ \frac{\beta}{h} \end{pmatrix}u_{i-1} + \begin{pmatrix} \mathrm{O}_{3\times 1} \\ -\frac{1-\beta}{h} \end{pmatrix}u_{i+1} & \forall i \in \mathcal{V} \setminus \{n\}, \\ \begin{pmatrix} A & \mathrm{O}_{3\times 1} \\ \mathrm{O}_{1\times 3} & -\frac{1}{h} \end{pmatrix}\begin{pmatrix} x_i \\ u_i \end{pmatrix} + \begin{pmatrix} B \\ -\frac{1}{h} \end{pmatrix}\bar{u}_i + \begin{pmatrix} \mathrm{O}_{3\times 1} \\ \frac{1}{h} \end{pmatrix}u_{i-1} & i = n. \end{cases} \tag{2.8.6}$$

Note that the values of the vehicle state $[d_i,\, v_i,\, a_i]^T$ for all vehicles $i \in \mathcal{V} \setminus \{n\}$ can be found by solving

$$\begin{cases} \beta a_{i-1} - \left(2\beta - 1 - \frac{h}{\tau}\right)a_i - (1-\beta)a_{i+1} & = \ddot{e}_i + \frac{h}{\tau}u_i, \\ \beta v_{i-1} - (2\beta - 1)v_i - (1-\beta)v_{i+1} & = \dot{e}_i + ha_i, \\ \beta d_i + (1-\beta)d_{i+1} & = e_i + r + hv_i, \end{cases}$$

given that a_0 and v_0 are known. For vehicle n, the same equations can be used with $\beta = 1$.

By using the lumped states $X^T := \left[x_1^T,\, \dots,\, x_n^T\right]$ and $U := [u_1,\, \dots,\, u_n]^T$, and by substituting the same distributed controller as used before in (2.6.1),

$$\bar{u}_i = -\sum_{j\in\mathcal{V}}\left[a_{ij}k^T(x_i - x_j)\right] - p_i k^T x_i$$

with controller gain vector

$$k^T := [k_1,\, k_2,\, k_3],$$

into the platoon dynamics of (2.8.6), we get the general closed-loop platoon dynamics:

(general closed-loop platoon dynamics)

$$\begin{pmatrix} \dot{X} \\ \dot{U} \end{pmatrix} = \begin{pmatrix} I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T & \mathrm{O}_{3n\times n} \\ \hat{\mathcal{L}} \otimes \frac{1}{h}k^T & A_u \end{pmatrix}\begin{pmatrix} X \\ U \end{pmatrix} + \begin{pmatrix} \mathrm{O}_{3n\times 1} \\ \beta B_u \end{pmatrix}u_0, \tag{2.8.7}$$

where

$$A_u := -\frac{1}{h}\begin{pmatrix} 2\beta-1 & 1-\beta & 0 & \dots & \dots & 0 \\ -\beta & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1-\beta & 0 \\ \vdots & & \ddots & -\beta & 2\beta-1 & 1-\beta \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix} \tag{2.8.8}$$

and

$$B_u := \left[\frac{1}{h},\, 0,\, \ldots,\, 0\right]^T.$$

Note that for $\beta = 1$, the closed-loop platoon dynamics in (2.8.7) derived from the error defined in (2.8.1) indeed simplify to (2.6.5), which was based on the error in (2.5.1). Also note that, as was mentioned in section 2.7, it follows from the platoon dynamics in (2.8.6) that the desired acceleration u_{i-1} of vehicle $i-1$ and the desired acceleration u_{i+1} of vehicle $i+1$ should be communicated to vehicle i with weights β and $1-\beta$ respectively. In addition, also the weighted error states αx_{i-1} and $(1-\alpha)x_{i+1}$ should be communicated when using the Laplacian matrix in (2.1.10) and the pinning matrix in (2.1.11). All communicated signals are shown in figure 2.6.

2.9 Stability

Now that we have derived the closed-loop platoon dynamics in (2.8.7), we still need to find values for the weights $\alpha \in [0,1]$ and $\beta \in [0,1]$, and for the controller gain vector $k^T = [k_1,\, k_2,\, k_3]$ such that the closed-loop system is stable. Fortunately, since (2.8.7) is lower block triangular, we only need to consider the diagonal blocks. First of all, we consider the subsystem $\dot{X} = (I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T)X$. It turns out that, when using Laplacian matrix (2.1.10) and pinning matrix (2.1.11) with $\alpha > 0$, the system is asymptotically stable if and only if the values of k^T satisfy

$$\begin{cases} k_1 > 0 \\ k_2 > \frac{k_1 \tau}{\min_i\{\lambda_i k_3 + 1\}} \\ k_3 > -\frac{1}{\max_i\{\lambda_i\}} \end{cases} \tag{2.9.1}$$

where λ_i is the i-th eigenvalue of $\hat{\mathcal{L}}$ and τ is the drive-line dynamics time constant. This is proven in appendix A. Next, we need to prove that matrix A_u, which is defined in (2.8.8) and forms the right lower block of the system matrix of (2.8.7), is stable. Since subsystem $\dot{X} = (I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T)X$ is asymptotically stable, X will converge to zero. Hence, in steady state the first term of subsystem $\dot{U} = (\hat{\mathcal{L}} \otimes \frac{1}{h}k^T)X + A_u U$ equals zero and thus we only need to consider the matrix A_u. In appendix A, it is proven that subsystem $\dot{U} = A_u U$ is asymptotically stable for $\beta > \frac{1}{2}$.

To summarize, the closed-loop platoon dynamics of (2.8.7) are asymptotically stable when $\alpha \in (0,1]$, $\beta \in \left(\frac{1}{2}, 1\right]$, and $k^T := [k_1,\, k_2,\, k_3]$ satisfies (2.9.1).

3. Merging onto the highway

In this chapter, the model of the platoon from chapter 2 is adapted such that an automated vehicle is able to merge into a platoon on the highway. This requires creating a gap in the platoon for the merging vehicle and controlling this merging vehicle such that it will drive next to the created gap at some point before the end of the acceleration lane. Furthermore, in order to guarantee a successful merge it is desirable that the platoon and the merging vehicle are able to interact. The merging vehicle should adjust to any disturbances that occur in the platoon, and vice versa. Two slightly different models are proposed in sections 3.2 and 3.3. Both models follow a similar approach:

1. A time or location is chosen where the vehicle should merge into the platoon.
2. Based on this time or location, and possibly additional (estimated) variables, two vehicles of the platoon are selected to increase the distance between them to make room for the merging vehicle.
3. The platoon and the merging vehicle are controlled such that at approximately the chosen time or position the merging vehicle can merge into the platoon between the two chosen vehicles.

A more detailed description is given in section 3.1. The difference between the models is that section 3.2 considers the case that the merging vehicle aims to merge at the end of a specified time interval, whereas section 3.3 covers the scenario that the merging vehicle merges somewhere between two given locations. For modeling the merge of a vehicle onto the highway, the model of section 3.3 is the obvious choice: the merging vehicle that is driving on the entrance ramp to the highway cannot merge before it reaches the acceleration lane and must complete the merge before the end of the acceleration lane. The model of section 3.2 requires an estimate of the time that the merging vehicle reaches the acceleration lane, and therefore results in a more open-loop model.

3.1 Problem description

Modeling the situation where an automated vehicle merges into a platoon on the highway requires partly the same methods as used in chapter 2. Similar to chapter 2 the goal for the models in this chapter is that eventually all vehicles, including the merging vehicle, drive behind each other with the same constant velocity and that the distances between vehicles equal the desired distances. There are, however, also some new issues that need to be addressed. In particular, we need to consider how to

- choose a time or location for the merge,
- choose between which vehicles of the platoon the merging vehicle will merge,
- control the merging vehicle such that it will accelerate to the right speed,
- control the platoon such that it will create the gap,
- control both such that the two chosen vehicles of the platoon and the merging vehicle will arrive at the same location at the same time,
- check if the merging vehicle is able to merge,
- and, finally, ensure that the vehicle can merge before it reaches the end of the acceleration lane.

Since the merge must be completed before the end of the acceleration lane, it might be a good idea to design the model such that it aims to merge the vehicle and the platoon as soon as possible. For the highway this means as soon as the merging vehicle enters the acceleration lane. This way, if any problems arise that result in the merging vehicle not being able to merge at this time, the entire length of the acceleration lane can be used to resolve these issues. As a consequence, the platoon needs to start forming a gap before the merging vehicle enters the acceleration lane. Preferably the platoon increases the distance between two vehicles gradually, and hence it will take the platoon some time to create the gap. Therefore, it is desirable that the platoon starts forming the gap as soon as the merging vehicle arrives at the beginning of the entrance ramp. At this moment it becomes certain that the vehicle will enter the highway, and hence the platoon and the merging vehicle can start preparing for the merge. The merging vehicle chooses where to merge and starts accelerating, and the platoon starts forming a gap between the chosen vehicles at the same time.

These choices lead to the following situation, which is illustrated by figure 3.1:

1. A platoon consisting of n vehicles is driving on the highway. At time t_{start} a vehicle arrives at the beginning of the entrance ramp, which is located at position q_{start}. At the same time this vehicle decides between which vehicles of the platoon it will merge onto the highway, i.e., it chooses a vehicle m and will then try to merge into the platoon between vehicles m and $m-1$. The merging vehicle starts accelerating to the platoon velocity and the platoon starts increasing the distance between vehicles m and $m-1$.

2. Next, at time t_{ramp}, the merging vehicle arrives at position q_{ramp}, which marks the end of the entrance ramp and the beginning of the acceleration lane. From this point onwards, the merging vehicle is allowed to move onto the highway, provided that there is enough space. The platoon has formed a gap between the chosen vehicles m and $m-1$, and the gap is roughly at the same position as the merging vehicle.

3. Most likely, the merging vehicle is not yet able to merge at time t_{ramp}, possibly because the gap is not large enough or because the vehicle is not properly aligned with the gap. It then takes until time t_{merge} before the vehicle is able to merge (though it is also possible that $t_{\mathrm{ramp}} = t_{\mathrm{merge}}$).

4. Finally, at time t_{merge} the vehicle merges into the platoon before it reaches the end of the acceleration lane, which is denoted by q_{end}, and continues on the highway as part of the platoon.

Note that the positions q_{start}, q_{ramp}, and q_{end} are fixed, but that the times t_{start}, t_{ramp}, and t_{end} are not. These times are defined as the moment that the merging vehicle reaches q_{start}, q_{ramp}, or q_{end}, and therefore depend on the way the vehicles are controlled. The same holds for t_{merge}, which is defined as the time that the merging vehicle detects enough space in the platoon in order to merge. The goal for the remainder of this chapter is to model the merging process as described above starting at time t_{start} and to control both the platoon and the merging car such that the merge can be safely executed.

Before continuing with developing the models for merging onto the highway, some notation is introduced and illustrated by figure 3.2. Most of the variables were also used in chapter 2. The platoon, given by the set $\mathcal{V} = \{1, \ldots, n\}$, consists of n vehicles. A virtual reference vehicle is indicated by index $i = 0$. The merging vehicle, marked by the 'merging' label in figure 3.2, chooses to merge in front of vehicle m, which means that a gap is created between vehicles m and $m-1$ of the platoon. The subscript of a variable indicates to which vehicle the variable relates. Figure 3.2 shows for vehicle m, the merging vehicle, and vehicle $m-1$ the position $q(t)$, the velocity $v(t)$, the distance $d(t)$ to the (virtual) car in front, and the length L of the vehicle. The virtual car that drives in front of the merging vehicle is a copy of vehicle $m-1$, hence $v_{0,\,\mathrm{merging}}(t) = v_{m-1}(t)$, and will be used in the models of sections 3.2 and 3.3. The platoon velocity is defined to be the velocity of the virtual car that is driving in front of the platoon (which is not shown in figure 3.2):

$$v_{\mathrm{platoon}}(t) \coloneqq v_0(t).$$

In addition, also the desired distance $d^d(t)$ between a vehicle and its neighbors, the acceleration $a(t)$ and the desired acceleration $u(t)$ will be used. The desired distance of vehicle i was defined in (2.4.1) as

$$d_i^d(t) \coloneqq r + hv_i(t)$$

with standstill distance r and time gap h. The drive-line dynamics time constant is given by τ and the drive-line dynamics time delay ϕ is still assumed to be zero. The distance from a vehicle to the beginning

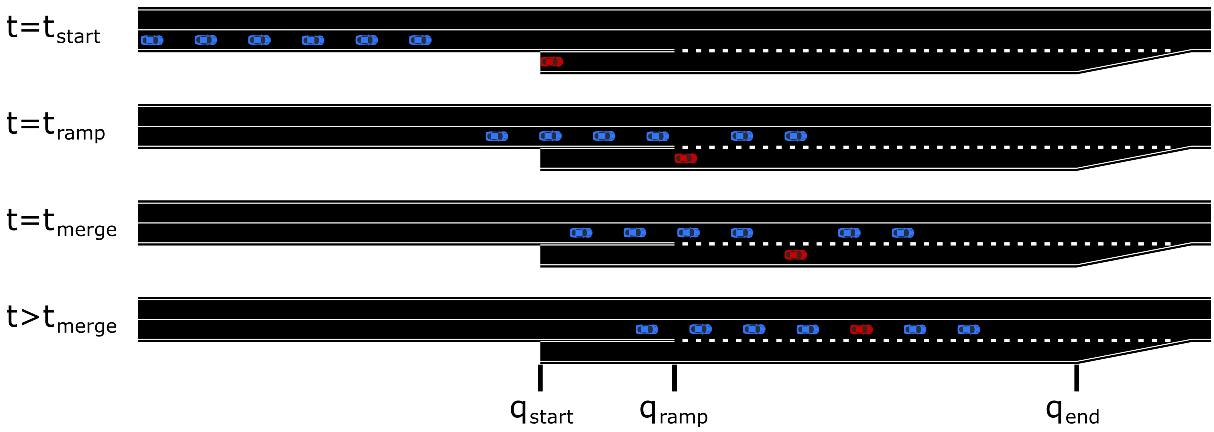


Figure 3.1: A vehicle merges into a platoon on the highway. From top to bottom: (1) at time t_{start}, a vehicle enters the entrance ramp, which is marked by q_{start}, (2) at time t_{ramp}, the car reaches the beginning of the acceleration lane at q_{ramp}, where the platoon has made a gap, (3) the merging car aligns itself with the gap and the platoon adjusts the gap to the merging car, such that the car is able to merge at time t_{merge}, (4) the car merges before it reaches the end of the acceleration lane at position q_{end} and continues on the highway as part of the platoon.

of the acceleration lane is given by

$$\delta_i(t) := q_{\text{ramp}} - q_i(t)$$

for the platoon and by

$$\delta_{\text{merging}}(t) := q_{\text{ramp}} - q_{\text{merging}}(t)$$

for the merging vehicle. The distance $\delta_{\text{end}}(t)$ from the merging vehicle to the end of the acceleration lane is defined as

$$\delta_{\text{end}}(t) := q_{\text{end}} - q_{\text{merging}}(t).$$

Note, however, that it was assumed in section 2.2 that the positions $q_i(t)$ of the vehicles in the platoon and the position $q_{\text{merging}}(t)$ of the merging car are not known due to inaccurate GPS data. The distances $\delta_i(t)$, $\delta_{\text{merging}}(t)$, and $\delta_{\text{end}}(t)$ should therefore be obtained directly from, for example, radar measurements of the distances between vehicles and q_{ramp} or q_{end}.

Sections 3.2 and 3.3 will both describe a model for the platoon and the merging vehicle during the time interval $[t_{\text{start}}, t_{\text{merge}}]$. Outside of this interval, assuming that no other vehicles will merge with the platoon, the platoon can be modeled by the closed-loop dynamics of section 2.8. Since the methods and notations used in sections 3.2 and 3.3 are similar to those of section 2.8, this model is repeated below. In the model for merging onto the highway, it is also used to model the platoon until time t_{start}. For $t > t_{\text{merge}}$, a similar model is used but then the platoon consists of $n+1$ vehicles instead of n vehicles since the merging vehicle is then part of the platoon. Dropping the time arguments to improve readability, recall that the platoon can be modeled by the dynamics that were given in (2.8.7) as

(closed-loop platoon dynamics)

$$\begin{pmatrix} \dot{X} \\ \dot{U} \end{pmatrix} = \begin{pmatrix} I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T & \mathrm{O}_{3n\times n} \\ \hat{\mathcal{L}} \otimes \frac{1}{h} k^T & A_u \end{pmatrix} \begin{pmatrix} X \\ U \end{pmatrix} + \begin{pmatrix} \mathrm{O}_{3n\times 1} \\ \beta B_u \end{pmatrix} u_0 \tag{3.1.1}$$

with lumped states $X^T := [x_1^T, \dots, x_n^T]$ and $U := [u_1, \dots, u_n]^T$, controller gain vector $k^T := [k_1, k_2, k_3]$, weights $\alpha \in [0,1]$ and $\beta \in [0,1]$, and the matrices

$$A := \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix},$$

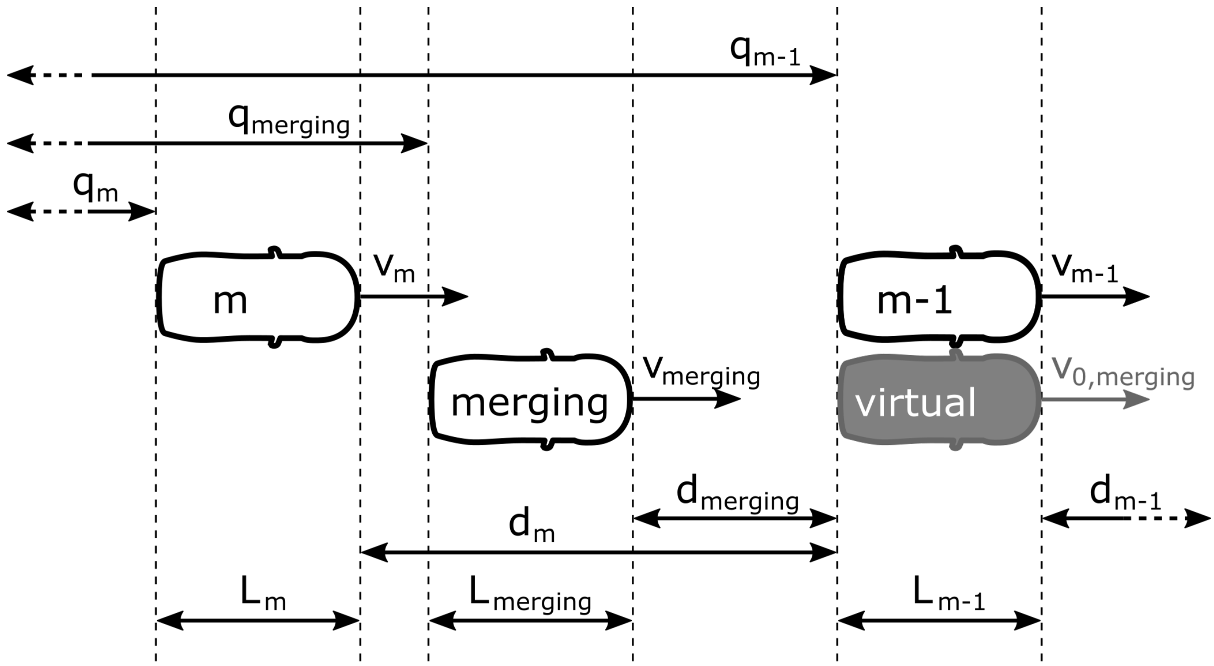


Figure 3.2: Schematic drawing of a vehicle (with label 'merging') merging between vehicles m and $m-1$ of the platoon. The position q, velocity v, distance d to its (virtual) predecessor, and length L of each vehicle are depicted by the arrows. The subscripts of the variables indicate to which vehicle an arrow applies. A virtual copy of vehicle $m-1$ with velocity $v_{0,\,\mathrm{merging}} = v_{m-1}$ drives in front of the merging vehicle.

$$A_u \coloneqq -\frac{1}{h}\begin{pmatrix} 2\beta-1 & 1-\beta & 0 & \dots & \dots & 0 \\ -\beta & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1-\beta & 0 \\ \vdots & & \ddots & -\beta & 2\beta-1 & 1-\beta \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix}, \tag{3.1.2}$$

$B \coloneqq \left[0,\ 0,\ \frac{1}{\tau}\right]^T$, and $B_u \coloneqq \left[\frac{1}{h},\ 0,\ \dots,\ 0\right]^T$. These dynamics were based on the error

$$e_i = \beta\left[d_i - r - hv_i\right] + (1-\beta)\left[d_{i+1} - r - hv_i\right],$$

the error state

$$x_i \coloneqq \begin{pmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{pmatrix},$$

and controller

$$\bar{u}_i = -\sum_{j\in\mathcal{V}}\left[a_{ij}k^T(x_i - x_j)\right] - p_i k^T x_i, \tag{3.1.3}$$

where a_{ij} are the entries of the adjacency matrix

$$\mathcal{A} = \begin{pmatrix} 0 & 1-\alpha & 0 & \dots & 0 \\ \alpha & 0 & 1-\alpha & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \alpha & 0 & 1-\alpha \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}.$$

The communication topology of the platoon is described by the matrix $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$ with Laplacian matrix

$$\mathcal{L} = \begin{pmatrix} 1-\alpha & -(1-\alpha) & 0 & \dots & 0 \\ -\alpha & 1 & -(1-\alpha) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\alpha & 1 & -(1-\alpha) \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix}$$

and pinning matrix

$$\mathcal{P} = \begin{pmatrix} \alpha & 0 & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}.$$

It is assumed that the platoon drives with constant velocity $\bar{v}_{\text{platoon}}$, that is, all vehicles of the platoon follow a virtual reference vehicle that has constant velocity $\bar{v}_{\text{platoon}}$. The velocity of each vehicle might vary somewhat, but when the distances between vehicles are equal to the desired distances, all vehicles will be driving with velocity $\bar{v}_{\text{platoon}}$.

3.2 Merging at a fixed time

The model given in (3.1.1) for the behavior of a platoon forms the starting point for modeling the merge of a vehicle into a platoon on the highway. As mentioned before, in this section we consider the situation where the merging vehicle aims to merge into the platoon at a fixed time and design a model that describes both the platoon and the merging vehicle during the time interval $[t_{\text{start}}, t_{\text{merge}}]$. Outside of this interval both the platoon and the merging vehicle, which can be considered as a platoon consisting of one vehicle, are modeled by the dynamics given in (3.1.1). The platoon and the merging vehicle start preparing for the merge when the merging vehicle enters the entrance ramp to the highway at time t_{start} and aim to execute the merge as soon as the merging vehicle reaches the acceleration lane at time t_{ramp}. More details of the situation that is considered are given in section 3.1.

The model is designed as follows:

1. At time t_{start} an estimate $\tilde{t}_{\text{ramp}}$ of time t_{ramp} is computed, which will be the desired time for the merge, and the vehicle m is selected.
2. Next, the platoon is controlled during the time interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$ such that it creates a gap between vehicles m and $m-1$ of the platoon.
3. The merging vehicle is controlled during the time interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$ such that it aligns itself with the gap in the platoon.
4. Finally, both the merging vehicle and the platoon are controlled during the time interval $[\tilde{t}_{\text{ramp}}, t_{\text{merge}}]$ until the merging vehicle is able to merge into the platoon between vehicles m and $m-1$ at time t_{merge}.

Due to delays, noise in signals, or any other disturbances, it is not guaranteed that the merging vehicle is able to merge into the platoon at time $\tilde{t}_{\text{ramp}}$. Even if there is enough distance between the two chosen vehicles and the merging vehicle is at the right position relative to the platoon, the merging vehicle might not be driving in the acceleration lane at time $\tilde{t}_{\text{ramp}}$. Therefore, we also need to consider how to control the platoon and the merging vehicle after this time and how to decide that the merging vehicle is able to merge into the platoon.

The outline of this section is as follows. First, section 3.2.1 describes how to choose the vehicle m. Next, sections 3.2.2 and 3.2.3 describe how the platoon can be controlled such that it creates a gap between vehicles m and $m-1$. The merging car is controlled in section 3.2.4 such that it aligns itself with the formed gap in the platoon. In section 3.2.5 the control strategy of the platoon is slightly changed such that it takes into account the position of the merging vehicle relative to the platoon when creating a gap in the platoon. Finally, section 3.2.6 discusses the control of the merging vehicle and the platoon after time $\tilde{t}_{\mathrm{ramp}}$ until they merge at time t_{merge}.

3.2.1 Choosing a merge location

At a certain moment in time, it needs to be decided where the merging vehicle will merge into the platoon. As it takes the platoon some time to create the gap, it is desirable that this is done carefully and as soon as possible. If either the merging car is not able to reach the gap or the platoon does not make a gap, the car will not be able to merge, which, depending on the length of the acceleration lane, might end badly. Therefore, it is assumed that the merging car will choose where to merge when it enters the entrance ramp to the highway at time t_{start} and that the platoon will start forming a gap at the same time. Ideally, when the merging vehicle enters the acceleration lane at time t_{ramp}, the gap is fully formed and positioned exactly next to the merging vehicle such that it can merge at this time, i.e., such that $t_{\mathrm{merge}} = t_{\mathrm{ramp}}$. This also leaves some space to resolve any problems that might arise: during the complete length of the acceleration lane, both the merging vehicle and the platoon are able to make any adjustments in order to ensure a successful merge.

To choose a merge location, the merging vehicle estimates the time it takes to drive from the beginning of the entrance ramp q_{start} to the beginning of the acceleration lane q_{ramp} in case it uses a constant acceleration until it reaches the same speed as the platoon. The merging car also computes where each vehicle in the platoon will approximately be relative to q_{ramp} at the estimated time assuming that the platoon maintains a constant velocity. This assumes that the distances of all vehicles to the beginning of the acceleration lane are known. With these approximations, the merging car chooses to merge in front of the car that is expected to be the last car of the platoon that will have passed the beginning of the acceleration lane before arriving there itself.

The platoon is cruising at constant velocity $\bar{v}_{\mathrm{platoon}}$. At time t_{start}, a vehicle enters the entrance ramp at q_{start} with velocity $v_{\mathrm{merging}}(t_{\mathrm{start}})$. The end of the entrance ramp and beginning of the acceleration lane is marked by q_{ramp}. The distance between the back of a vehicle and q_{ramp} at that time is given by $\delta_i(t_{\mathrm{start}}) := q_{\mathrm{ramp}} - q_i(t_{\mathrm{start}})$ for each vehicle i in the platoon and $\delta_{\mathrm{merging}}(t_{\mathrm{start}}) := q_{\mathrm{ramp}} - q_{\mathrm{merging}}(t_{\mathrm{start}}) = q_{\mathrm{ramp}} - q_{\mathrm{start}}$ for the merging vehicle. These distances $\delta_i(t_{\mathrm{start}})$ and $\delta_{\mathrm{merging}}(t_{\mathrm{start}})$ are assumed to be known. At the same time, the merging vehicle estimates the time it takes to reach q_{ramp} assuming it will first accelerate with a constant acceleration a_{ramp} until it reaches the velocity $\bar{v}_{\mathrm{platoon}}$ of the platoon and then continues to drive with constant velocity.

The time $t_{\mathrm{accelerate}}$ it takes to accelerate from velocity $v_{\mathrm{merging}}(t_{\mathrm{start}})$ to $\bar{v}_{\mathrm{platoon}}$ with acceleration a_{ramp} can be estimated as

$$t_{\mathrm{accelerate}} = \frac{\bar{v}_{\mathrm{platoon}} - v_{\mathrm{merging}}(t_{\mathrm{start}})}{a_{\mathrm{ramp}}}. \tag{3.2.1}$$

During that time, the merging vehicle will travel approximately a distance of

$$v_{\mathrm{merging}}(t_{\mathrm{start}})\; t_{\mathrm{accelerate}} + 0.5\; a_{\mathrm{ramp}}\; t_{\mathrm{accelerate}}^2.$$

For the remaining distance it then needs to travel before reaching q_{ramp},

$$q_{\mathrm{ramp}} - q_{\mathrm{start}} - v_{\mathrm{merging}}(t_{\mathrm{start}})\; t_{\mathrm{accelerate}} - 0.5\; a_{\mathrm{ramp}}\; t_{\mathrm{accelerate}}^2,$$

the merging vehicle will travel with velocity $\bar{v}_{\mathrm{platoon}}$ and hence the time t_{ramp} at which the merging vehicle arrives at q_{ramp} can be estimated as

$$t_{\mathrm{ramp}} \approx \tilde{t}_{\mathrm{ramp}} = t_{\mathrm{start}} + t_{\mathrm{accelerate}} + \frac{q_{\mathrm{ramp}} - q_{\mathrm{start}} - v_{\mathrm{merging}}(t_{\mathrm{start}})\; t_{\mathrm{accelerate}} - 0.5\; a_{\mathrm{ramp}}\; t_{\mathrm{accelerate}}^2}{\bar{v}_{\mathrm{platoon}}}. \tag{3.2.2}$$

Given this estimation $\tilde{t}_{\mathrm{ramp}}$, the platoon will travel a distance of approximately $\bar{v}_{\mathrm{platoon}} \cdot (\tilde{t}_{\mathrm{ramp}} - t_{\mathrm{start}})$ until the merging vehicle reaches q_{ramp} at time t_{ramp}. Assuming that it is preferred that the gap in the platoon is slightly in front of the merging car such that the merging car can accelerate into the gap, the merging vehicle will merge in front of the last car that will most likely have passed q_{ramp} at time t_{ramp}, i.e., the vehicle with the largest index i for which $\bar{v}_{\text{platoon}} \cdot (\tilde{t}_{\text{ramp}} - t_{\text{start}}) > \delta_i(t_{\text{start}})$. This vehicle is then defined to be the vehicle m, in front of which the merging vehicle will merge into the platoon.

Note that this, however, does create some problems. For example, if the entire platoon is expected to have passed q_{ramp} at time t_{ramp}, then the merging vehicle tries to merge in front of vehicle n, while in that case it is easier to slow down a bit and enter the highway behind the platoon. Similarly, when none of the vehicles or just the first vehicle of the platoon is expected to have passed q_{ramp} at time t_{ramp}, the platoon should be able to accelerate and enter the highway in front of the platoon. Therefore, it is assumed that, when the merging vehicle enters the acceleration lane at time t_{ramp}, it will be driving next to the platoon, or more specifically, between the second and last vehicle of the platoon. That is, at time t_{ramp} it should hold that $q_2(t_{\text{ramp}}) > q_{\text{merging}}(t_{\text{ramp}}) = q_{\text{ramp}} > q_n(t_{\text{ramp}})$. To move onto the highway, the vehicle has to merge into the platoon in front of a vehicle m for which $2 \leq m \leq n-1$. In all other cases, the merging vehicle does not need to merge into the platoon, but will be able to either accelerate and enter the highway in front of the platoon or decelerate and enter the highway behind the platoon. Possibly, the merging vehicle could in that case join the platoon in front of the first vehicle or behind the last vehicle, but this is also not included in the model.

To summarize, the merging vehicle chooses to merge in front of the vehicle m, which is defined as

(choosing a vehicle m)

$$m = \max_{i=2,\,\ldots,n-1} \{i\} \quad \text{subject to} \quad \bar{v}_{\text{platoon}} \cdot (\tilde{t}_{\text{ramp}} - t_{\text{start}}) > \delta_i(t_{\text{start}}), \tag{3.2.3}$$

where $\tilde{t}_{\text{ramp}}$ is the estimate of the time t_{ramp} where the merging vehicle reaches the acceleration ramp and is given in (3.2.2). The estimated $\tilde{t}_{\text{ramp}}$ is also the time at which the merging vehicle aims to merge into the platoon.

3.2.2 Creating a gap

In the previous section, it was described how, at time t_{start}, the estimate $\tilde{t}_{\text{ramp}}$ is computed and the vehicle m is selected. Next, during $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$, the platoon should create a gap between vehicles m and $m-1$ such the merging vehicle will be able to merge into the platoon between vehicles m and $m-1$. The gap between vehicles m and $m-1$ is created by increasing the standstill distance r_m of vehicle m. This can be done directly (as in this section) or by letting the standstill distance depend on the position of the merging vehicle (see section 3.2.5).

A gap is created when the distance between two vehicles is increased such that a different vehicle can merge into the platoon. This can be done in several ways, such as changing the value of the desired distance between two cars or inserting a virtual car into the platoon. However, it is desirable to find a method that will slowly increase the distance between cars to avoid any sudden braking or acceleration. To make the gap, the standstill distance r_m of vehicle m is slowly increased over the time interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$, which means that standstill distance r_m becomes a function of time. In section 3.2.3 it will be discussed how to choose the function $r_m(t)$. Based on the derivations in the remainder of this section some constraints for $r_m(t)$ and its derivatives will be formulated in section 3.2.3, which will be used to find a suitable function $r_m(t)$.

The desired distance d_m^d between vehicles m and $m-1$ is therefore now given by

$$d_m^d(t) = r_m(t) + hv_m(t)$$

and the errors e_m and e_{m-1} of vehicles m and $m-1$ then become

$$e_m(t) = \beta \left[d_m(t) - r_m(t) - hv_m(t)\right] + (1-\beta)\left[d_{m+1}(t) - r - hv_m(t)\right]$$

and

$$e_{m-1}(t) = \beta \left[d_{m-1}(t) - r - hv_{m-1}(t)\right] + (1-\beta)\left[d_m(t) - r_m(t) - hv_{m-1}(t)\right],$$

where weight $\beta \in [0,1]$. Note that the second term of error e_m and the first term of error e_{m-1} still contain the constant standstill distance r, whereas the first term of e_m and the second term of e_{m-1}

now include the time-dependent standstill distance $r_m(t)$. The reason for this is that both vehicle m and vehicle $m-1$ now try to maintain a different distance to each of its neighbors. The constant standstill distance r still applies to distances d_{m+1} and d_{m-1}, while only the distance d_m between vehicles m and $m-1$ needs to be increased.

Now, the error $e_i(t)$ of each vehicle i in the platoon, including vehicles m and $m-1$, can be given by

$$e_i(t) = \beta\left[d_i(t) - r_i(t) - hv_i(t)\right] + (1-\beta)\left[d_{i+1}(t) - r_{i+1}(t) - hv_i(t)\right], \tag{3.2.4}$$

where r_i is the standstill distance of vehicle i and it holds that

$$r_i(t) = r \quad \forall i \in \mathcal{V} \setminus \{m\}.$$

With this error, the closed-loop platoon dynamics can be derived in the same way as was done in section 2.8. Leaving out the time arguments to improve readability, the time derivatives of error (3.2.4) are given by

$$\begin{aligned}
\dot{e}_i &= \beta\left[v_{i-1} - v_i - \dot{r}_i - ha_i\right] + (1-\beta)\left[v_i - v_{i+1} - \dot{r}_{i+1} - ha_i\right], \\
\ddot{e}_i &= \beta\left[a_{i-1} + \frac{h-\tau}{\tau}a_i - \frac{h}{\tau}u_i - \ddot{r}_i\right] + (1-\beta)\left[\frac{h+\tau}{\tau}a_i - a_{i+1} - \frac{h}{\tau}u_i - \ddot{r}_{i+1}\right], \\
\dddot{e}_i &= -\frac{1}{\tau}\ddot{e}_i + \frac{1}{\tau}\bar{u}_i + \beta\left[-\frac{1}{\tau}\ddot{r}_i - \dddot{r}_i\right] + (1-\beta)\left[-\frac{1}{\tau}\ddot{r}_{i+1} - \dddot{r}_{i+1}\right],
\end{aligned} \tag{3.2.5}$$

where input $\bar{u}_i$ is defined as

$$\bar{u}_i := \beta\left[u_{i-1} - u_i - h\dot{u}_i\right] + (1-\beta)\left[u_i - u_{i+1} - h\dot{u}_i\right].$$

Define the error state x_i of vehicle i as

$$x_i := \begin{pmatrix} e_i \\ \dot{e}_i \\ \ddot{e}_i \end{pmatrix}.$$

This leads to error dynamics

$$\dot{x}_i = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix}}_{A} x_i + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix}}_{B} \bar{u}_i + \beta \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{1}{\tau} & -1 \end{pmatrix}}_{B_r} \begin{pmatrix} \ddot{r}_i \\ \dddot{r}_i \end{pmatrix} + (1-\beta) \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{1}{\tau} & -1 \end{pmatrix}}_{B_r} \begin{pmatrix} \ddot{r}_{i+1} \\ \dddot{r}_{i+1} \end{pmatrix}. \tag{3.2.6}$$

The input $\bar{u}_i$ can again be rewritten as the filter

$$\dot{u}_i = -\frac{2\beta-1}{h}u_i + \frac{1}{h}\left(\beta u_{i-1} - (1-\beta)u_{i+1} - \bar{u}_i\right). \tag{3.2.7}$$

The dynamics of the complete platoon are thus given by

$$\begin{aligned}
\begin{pmatrix} \dot{x}_i \\ \dot{u}_i \end{pmatrix} &= \begin{pmatrix} A & \mathrm{O}_{3\times1} \\ \mathrm{O}_{1\times3} & -\frac{2\beta-1}{h} \end{pmatrix} \begin{pmatrix} x_i \\ u_i \end{pmatrix} + \begin{pmatrix} B \\ -\frac{1}{h} \end{pmatrix} \bar{u}_i + \begin{pmatrix} \mathrm{O}_{3\times1} \\ \frac{\beta}{h} \end{pmatrix} u_{i-1} + \begin{pmatrix} \mathrm{O}_{3\times1} \\ -\frac{1-\beta}{h} \end{pmatrix} u_{i+1} \\
&\quad + \begin{pmatrix} \beta B_r \\ \mathrm{O}_{3\times2} \end{pmatrix} \begin{pmatrix} \ddot{r}_i \\ \dddot{r}_i \end{pmatrix} + \begin{pmatrix} (1-\beta)B_r \\ \mathrm{O}_{3\times2} \end{pmatrix} \begin{pmatrix} \ddot{r}_{i+1} \\ \dddot{r}_{i+1} \end{pmatrix}
\end{aligned} \tag{3.2.8}$$

for all $i \in \mathcal{V}$ except that for $i = n$ we set $\beta = 1$.

Note that when r_i is constant in time for all $i \in \mathcal{V}$, its derivatives are equal to zero and then these dynamics in (3.2.8) are the same as (2.8.6). This allows us to use (3.2.8) for all vehicles in the platoon. In addition, with the introduction of a virtual reference vehicle as was done in section 2.3, the error in (3.2.4), and hence (3.2.8), is also defined for the first vehicle of the platoon.

By using the lumped states $X^T := \left[x_1^T, \ldots, x_n^T\right]$ and $U := \left[u_1, \ldots, u_n\right]^T$ and the same controller as before, given in (3.1.3) as

$$\bar{u}_i = -\sum_{j\in\mathcal{V}}\left[a_{ij}k^T(x_i - x_j)\right] - p_i k^T x_i,$$

and by defining the vector $R \coloneqq [\ddot{r}_1, \dddot{r}_1, \ldots, \ddot{r}_n, \dddot{r}_n]^T$, the closed-loop platoon dynamics become

$$\begin{pmatrix} \dot{X} \\ \dot{U} \end{pmatrix} = \begin{pmatrix} I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T & \mathrm{O}_{3n\times n} \\ \hat{\mathcal{L}} \otimes \frac{1}{h}k^T & A_u \end{pmatrix} \begin{pmatrix} X \\ U \end{pmatrix} + \begin{pmatrix} \mathrm{O}_{3n\times 1} & J \otimes B_r \\ \beta B_u & \mathrm{O}_{n\times 2n} \end{pmatrix} \begin{pmatrix} u_0 \\ R \end{pmatrix}, \tag{3.2.9}$$

where again $k^T \coloneqq [k_1, k_2, k_3]$, $\hat{\mathcal{L}} \coloneqq \mathcal{L} + \mathcal{P}$, $B_u \coloneqq [\frac{1}{h}, 0, \ldots, 0]^T$, B_r is defined in (3.2.6), A_u is given by (3.1.2), I_n is the $n \times n$ identity matrix, and

$$J = \begin{pmatrix} \beta & 1-\beta & 0 & \ldots & 0 \\ 0 & \beta & 1-\beta & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & \beta & 1-\beta \\ 0 & \ldots & 0 & 0 & 1 \end{pmatrix}.$$

Note that when $r_i(t) = r$ for all $i \in \mathcal{V}$ and hence all derivatives equal zero, that then (3.2.9) equals (3.1.1).

It is, however, not necessary to include the second and third derivatives of r_i for all vehicles $i \neq m$, $i \in \mathcal{V}$, since $r_i(t) = r$ for all $i \neq m$ and all these derivatives are equal to zero. Furthermore, since the second derivative cannot be chosen independently of the third derivative, we define the vector $R_m \coloneqq [r_m, \dot{r}_m, \ddot{r}_m]^T$ and add R_m to the state. This way the model requires only u_0 and $\dddot{r}_m$ as external inputs and the values of r_m can be obtained from the state. The closed-loop platoon dynamics are now given by

(closed-loop platoon dynamics with time-dependent standstill distance)

$$\begin{aligned}\begin{pmatrix} \dot{X} \\ \dot{U} \\ \dot{R}_m \end{pmatrix} &= \begin{pmatrix} I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T & \mathrm{O}_{3n\times n} & J(:,m) \otimes [\mathrm{O}_{3\times 2} \;\; B_r(:,1)] \\ \hat{\mathcal{L}} \otimes \frac{1}{h}k^T & A_u & \mathrm{O}_{n\times 3} \\ \mathrm{O}_{3\times 3n} & \mathrm{O}_{3\times n} & I_{(1),3} \end{pmatrix} \begin{pmatrix} X \\ U \\ R_m \end{pmatrix} \\ &\quad + \begin{pmatrix} \mathrm{O}_{3n\times 1} & J(:,m) \otimes B_r(:,2) \\ \beta B_u & \mathrm{O}_{n\times 1} \\ \mathrm{O}_{3\times 1} & I_3(:,3) \end{pmatrix} \begin{pmatrix} u_0 \\ \dddot{r}_m \end{pmatrix},\end{aligned} \tag{3.2.10}$$

where $J(:,m)$ denotes the m-th column of matrix J and

$$I_{(1),3} \coloneqq \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

To let multiple vehicles create a gap, either standstill distance r_i and its derivatives can be added to (3.2.10) for more vehicles or (3.2.9) can be used.

To convert the state variables of (3.2.10) back to vehicle state variables d, v, and a, one needs to solve

$$\begin{cases} \beta a_{i-1} + \left(\beta \frac{h-\tau}{\tau} + (1-\beta)\frac{h+\tau}{\tau}\right) a_i - (1-\beta) a_{i+1} & = \ddot{e}_i + \frac{h}{\tau} u_i + \beta \ddot{r}_i + (1-\beta)\ddot{r}_{i+1} \\ \beta v_{i-1} - (2\beta - 1) v_i - (1-\beta) v_{i+1} & = \dot{e}_i + h a_i + \beta \dot{r}_i + (1-\beta)\dot{r}_{i+1} \\ \beta d_i + (1-\beta) d_{i+1} & = e_i + h v_i + \beta r_i + (1-\beta) r_{i+1} \end{cases}$$

for $i = 1, \ldots, n-1$ while using $\beta = 1$ when $i = n$.

3.2.3 Defining the standstill distance function

With the new closed-loop platoon dynamics in (3.2.10) it becomes possible to define the standstill distance r_m for vehicle m as a function of time. Since the platoon should make a gap between vehicles m and $m-1$ during the time interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$, the standstill distance $r_m(t)$ is increased from the initial value r at t_{start} to $r + L_{\text{gap}}$ at $\tilde{t}_{\text{ramp}}$. Here, r is the fixed default standstill distance for all vehicles and L_{gap} is the additional distance needed to create the gap. Outside the interval, $r_m(t)$ is constant: $r_m(t) = r$ for $t < t_{\text{start}}$ and $r_m(t) = r + L_{\text{gap}}$ for $t > \tilde{t}_{\text{ramp}}$. This means that $r_m(t)$ should satisfy

$$r_m(t_{\text{start}}) = r \tag{3.2.11}$$

and

$$r_m(\tilde{t}_{\text{ramp}}) = r + L_{\text{gap}}. \tag{3.2.12}$$

In addition, according to (3.2.10), $\dddot{r}_m$ should exist, so we know that at least r_m, $\dot{r}_m$, and $\ddot{r}_m$ should be continuous. This means that, since $r_m(t)$ is constant outside of the interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$, the following should hold for the time derivatives:

$$\dot{r}_m(t_{\text{start}}) = 0, \qquad \dot{r}_m(\tilde{t}_{\text{ramp}}) = 0, \tag{3.2.13}$$

$$\ddot{r}_m(t_{\text{start}}) = 0, \qquad \ddot{r}_m(\tilde{t}_{\text{ramp}}) = 0. \tag{3.2.14}$$

A good choice would be to define $r_m(t)$ on the interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$ as a 5th-degree polynomial,

$$r_m(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0,$$

that satisfies the constraints given in (3.2.11) to (3.2.14). Then r_m, $\dot{r}_m$, and $\ddot{r}_m$ are continuous, whereas $\dddot{r}_m$ might be discontinuous at times t_{start} and $\tilde{t}_{\text{ramp}}$. A 5th-degree polynomial is used since there are six constraints and, unless $t_{\text{start}} = \tilde{t}_{\text{ramp}}$, it is always possible to find such a polynomial that satisfies all constraints. The coefficients of the polynomial can be determined by solving a constrained least squares problem.

3.2.4 The merging car

As mentioned in section 3.2.1, the merging vehicle decides at time t_{start} where it will merge into the platoon such that the platoon can start forming a gap as soon as possible. This decision is based on the estimated $\tilde{t}_{\text{ramp}}$ and constant platoon velocity $\bar{v}_{\text{platoon}}$. Sections 3.2.2 and 3.2.3 provide a way to let the platoon create the gap between vehicles m and $m-1$ during the time interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$. The next step is to model and control the merging vehicle. Assuming that the merging car has the same vehicle dynamics as the vehicles of the platoon, uses the same controller, and also follows a virtual reference vehicle, the merging vehicle can be considered as a platoon of size $n = 1$. Therefore, all previous models could also be applied to the merging vehicle. During $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$ the merging vehicle should be controlled such that it will arrive at the beginning of the acceleration lane q_{ramp} at approximately time $\tilde{t}_{\text{ramp}}$. Since the model aims to merge at time $\tilde{t}_{\text{ramp}}$, the merging vehicle should also be driving next to the gap in the platoon at this time. The actual time t_{ramp} at which the merging vehicle reaches q_{ramp} should be as close to estimation $\tilde{t}_{\text{ramp}}$ as possible, since the vehicle m in front of which the merging vehicle will merge was chosen based on this estimation and the platoon will not have formed the gap before $\tilde{t}_{\text{ramp}}$.

Both the time $t_{\text{accelerate}}$ that the merging vehicle needs to accelerate from its initial velocity at t_{start} to the platoon velocity given in (3.2.1) and the estimated $\tilde{t}_{\text{ramp}}$ given in (3.2.2) were derived using a constant acceleration a_{ramp} for the merging vehicle. One possibility for controlling the merging vehicle would hence be to use the input function

$$u_{0,\text{merging}}(t) = \begin{cases} a_{\text{ramp}} & t_{\text{start}} \le t \le t_{\text{start}} + t_{\text{accelerate}} \\ 0 & t > t_{\text{start}} + t_{\text{accelerate}} \end{cases} \tag{3.2.15}$$

in the closed-loop platoon dynamics of (3.1.1), which simplify to

$$\begin{pmatrix} \dot{x}_{\text{merging}} \\ \dot{u}_{\text{merging}} \end{pmatrix} = \begin{pmatrix} A - Bk^T & \mathrm{O}_{3\times 1} \\ \frac{1}{h}k^T & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} x_{\text{merging}} \\ u_{\text{merging}} \end{pmatrix} + \begin{pmatrix} \mathrm{O}_{3\times 1} \\ \frac{1}{h} \end{pmatrix} u_{0,\text{merging}} \tag{3.2.16}$$

for the merging vehicle. Since $u_{0,\text{merging}}$ is the desired acceleration of the virtual reference vehicle that is followed by the merging vehicle, the virtual car and hence also the merging vehicle will then accelerate with approximately the constant acceleration a_{ramp} until time $t_{\text{start}} + t_{\text{accelerate}}$ and then continue with a constant velocity.

This, however, is not a very robust method. After time t_{start}, both the vehicle m and the function $u_{0,\text{merging}}$ are fixed and the merging vehicle drives independently of the platoon to q_{ramp}. This can lead to problems when one of the vehicles does not drive as expected. If, for example, the platoon is not able to drive with constant velocity $\bar{v}_{\text{platoon}}$, while the merging vehicle arrives at q_{ramp} at time $\tilde{t}_{\text{ramp}}$, the merging vehicle could be too far from the gap in the platoon, since m was chosen based on a constant platoon velocity $\bar{v}_{\text{platoon}}$. And the same problem might arise if the platoon does continue with constant velocity $\bar{v}_{\text{platoon}}$, but the merging vehicle is not able to reach q_{ramp} at time $\tilde{t}_{\text{ramp}}$, since m also depends on the estimated $\tilde{t}_{\text{ramp}}$. In addition, the input function $u_{0,\text{merging}}$ in (3.2.16) is not continuous, while the actual acceleration a_{merging} of the merging vehicle will be continuous. Even if $u_{0,\text{merging}}$ was chosen to be continuous, it is not guaranteed that $a_{\text{merging}} = u_{0,\text{merging}}$. As a consequence, the merging vehicle might not reach the desired platoon velocity or arrive at q_{ramp} at time $\tilde{t}_{\text{ramp}}$, since $t_{\text{accelerate}}$ and $\tilde{t}_{\text{ramp}}$ were based on $a_{\text{merging}} = u_{0,\text{merging}}$.

Instead, we will use a different method that takes into account the position of the platoon. The merging vehicle is modeled as a platoon of size 1 by the closed-loop dynamics of (3.2.10), which simplify to

(closed-loop dynamics for the merging vehicle)

$$\begin{pmatrix}\dot{x}_{\text{merging}}\\ \dot{u}_{\text{merging}}\\ \dot{R}_{\text{merging}}\end{pmatrix} = \begin{pmatrix} A - Bk^T & \mathrm{O}_{3\times 1} & [\mathrm{O}_{3\times 2}\;\; B_r(:,1)] \\ \frac{1}{h}k^T & -\frac{1}{h} & \mathrm{O}_{1\times 3} \\ \mathrm{O}_{3\times 3} & \mathrm{O}_{3\times 1} & I_{(1),3}\end{pmatrix} \begin{pmatrix} x_{\text{merging}}\\ u_{\text{merging}}\\ R_{\text{merging}}\end{pmatrix} + \begin{pmatrix}\mathrm{O}_{3\times 1} & B_r(:,2)\\ \frac{1}{h} & \mathrm{O}_{1\times 1}\\ \mathrm{O}_{3\times 1} & I_3(:,3)\end{pmatrix}\begin{pmatrix} u_{0,\,\text{merging}}\\ \dddot{r}_{\text{merging}}\end{pmatrix} \tag{3.2.17}$$

for the merging vehicle, where $u_{0,\,\text{merging}}$ is the desired acceleration of the virtual reference vehicle driving in front of the merging vehicle and $R_{\text{merging}} := [r_{\text{merging}}, \dot{r}_{\text{merging}}, \ddot{r}_{\text{merging}}]^T$. Whereas the virtual vehicle of the platoon was assumed to be uncontrolled and driving at a constant velocity, the virtual vehicle for the merging car can be used to align the merging car with the gap in the platoon. This is done by defining the virtual car such that it is driving next to vehicle $m-1$ with exactly the same speed and acceleration. The virtual reference vehicle of the merging car will then be a virtual copy of vehicle $m-1$. This virtual car was also shown in figure 3.2. The merging vehicle will then be driving next to the gap in the platoon with the same velocity as vehicle $m-1$ when the error state of the merging vehicle goes to zero. In (3.2.17) the input $u_{0,\,\text{merging}}$ is therefore defined as

$$u_{0,\,\text{merging}} := u_{m-1}.$$

The standstill distance of the merging car can be defined as a function of time, similar to the process of creating a gap in the platoon, such that the merging car does not accelerate too fast but steadily moves towards the gap. Since the merging car should reach the gap at the end of the interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$, the standstill distance r_{merging} of the merging vehicle is increased from initial value $q_{m-1}(t_{\text{start}}) - q_{\text{start}} - L_{\text{merging}} - hv_{\text{merging}}(t_{\text{start}})$ at time t_{start} to the default value r at time $\tilde{t}_{\text{ramp}}$. The initial value is defined such that the error of the merging vehicle equals zero at time t_{start}. A 5-th degree polynomial that satisfies the constraints

$$r_{\text{merging}}(t_{\text{start}}) = q_{m-1}(t_{\text{start}}) - q_{\text{start}} - L_{\text{merging}} - hv_{\text{merging}}(t_{\text{start}})$$

and

$$r_{\text{merging}}(\tilde{t}_{\text{ramp}}) = r,$$

and whose first two derivatives equal zero at the boundary according to the constraints

$$\begin{aligned} \dot{r}_{\text{merging}}(t_{\text{start}}) &= 0, & \dot{r}_{\text{merging}}(\tilde{t}_{\text{ramp}}) &= 0,\\ \ddot{r}_{\text{merging}}(t_{\text{start}}) &= 0, & \ddot{r}_{\text{merging}}(\tilde{t}_{\text{ramp}}) &= 0,\end{aligned}$$

can be derived in the same way as is described in section 3.2.3 for r_m.

Note that, since the initial velocity of the merging vehicle is usually lower than the velocity of the platoon, the merging vehicle will start in front of the platoon, and hence the initial value of the standstill distance between the merging vehicle and vehicle $m-1$ is negative. Also note that $\tilde{t}_{\text{ramp}}$ was estimated based on a constant acceleration for the merging vehicle, but that the merging vehicle is not actually controlled to use this constant acceleration. The estimate $\tilde{t}_{\mathrm{ramp}}$ represents a realistic time at which the merging vehicle can reach q_{ramp} without accelerating too fast or having to slow down. This time is used as the desired time of the merge because it is known that the merging vehicle is able to arrive at q_{ramp} at this time.

3.2.5 Taking into account the position of the merging car

Thus far, in controlling the platoon as described in section 3.2.2, it was assumed that the standstill distance $r_m(t)$ and its derivatives are known. To create a gap in the platoon between vehicles m and $m-1$, standstill distance $r_m(t)$ can be chosen as described in section 3.2.3 such that it will increase from the default value r to $r + L_{\mathrm{gap}}$ in the time interval $[t_{\mathrm{start}}, \tilde{t}_{\mathrm{ramp}}]$. At the same time, the standstill distance of the merging car is increased such that it will drive next to the gap at time $\tilde{t}_{\mathrm{ramp}}$ as described in section 3.2.4. In practice, however, it is not guaranteed that the merging car will arrive at the gap within the given time interval, hence it might be desirable to let r_m depend on the distance of the merging car to vehicle $m-1$.

Consider again the situation that the merging vehicle will merge in front of vehicle m, that the platoon is driving at a constant velocity $\bar{v}_{\mathrm{platoon}}$, and that the merging vehicle follows a virtual copy of vehicle $m-1$ as described in section 3.2.4. Furthermore, since the platoon usually has a higher initial velocity than the merging vehicle, we assume that the merging vehicle starts in front of the platoon.

To define r_m as a function of the distance between the merging vehicle and vehicle $m-1$, the desired distance of the merging vehicle can be defined as $d^d_{\mathrm{merging}} = r + h\bar{v}_{\mathrm{platoon}}$. The desired velocity of the merging vehicle equals the platoon velocity $\bar{v}_{\mathrm{platoon}}$, which is why $\bar{v}_{\mathrm{platoon}}$ is used instead of v_{merging}. When $d_{\mathrm{merging}}(t) = d^d_{\mathrm{merging}}$, the merging vehicle is driving at the place where it can merge into the platoon, so the standstill distance r_m should then be equal to $r + L_{\mathrm{gap}}$. The same holds for $d_{\mathrm{merging}}(t) > d^d_{\mathrm{merging}}$, since then the merging vehicle has already passed the gap.

Suppose that there is some distance $L_{\mathrm{com}} \gg d^d_{\mathrm{merging}}$ that represents the maximum distance between the merging vehicle and any vehicle of the platoon where they are still able to communicate. If the distance between the merging vehicle and vehicle $m-1$ is larger than L_{com}, i.e., $d_{\mathrm{merging}}(t) \leq -L_{\mathrm{com}}$, they are not yet able to communicate with each other and hence vehicle m does not need to make a gap. The standstill distance should then be equal to the default value r. These requirements for r_m can be written as

$$\begin{aligned} r_m(d_{\mathrm{merging}} \geq r + h\bar{v}_{\mathrm{platoon}}) &= r + L_{\mathrm{gap}}, \\ r_m(d_{\mathrm{merging}} \leq -L_{\mathrm{com}}) &= r. \end{aligned}$$

For $-L_{\mathrm{com}} < d_{\mathrm{merging}} < r + h\bar{v}_{\mathrm{platoon}}$ a linear function can be used:

$$r_m(d_{\mathrm{merging}}) = r + \frac{L_{\mathrm{gap}}}{r + h\bar{v}_{\mathrm{platoon}} + L_{\mathrm{com}}} \left(d_{\mathrm{merging}} + L_{\mathrm{com}}\right).$$

Since $d_{\mathrm{merging}} = q_{m-1} - q_{\mathrm{merging}} - L_{\mathrm{merging}}$, the time derivatives of this function are given by

$$\begin{aligned} \dot{r}_m &= \frac{L_{\mathrm{gap}}}{r + h\bar{v}_{\mathrm{platoon}} + L_{\mathrm{com}}} \left(v_{m-1} - v_{\mathrm{merging}}\right), \\ \ddot{r}_m &= \frac{L_{\mathrm{gap}}}{r + h\bar{v}_{\mathrm{platoon}} + L_{\mathrm{com}}} \left(a_{m-1} - a_{\mathrm{merging}}\right), \end{aligned}$$

and

$$\begin{aligned} \dddot{r}_m &= \frac{L_{\mathrm{gap}}}{r + h\bar{v}_{\mathrm{platoon}} + L_{\mathrm{com}}} \left(-\frac{1}{\tau}a_{m-1} + \frac{1}{\tau}u_{m-1} + \frac{1}{\tau}a_{\mathrm{merging}} - \frac{1}{\tau}u_{\mathrm{merging}}\right) \\ &= -\frac{1}{\tau}\ddot{r}_m + \frac{1}{\tau}\left(\frac{L_{\mathrm{gap}}}{r + h\bar{v}_{\mathrm{platoon}} + L_{\mathrm{com}}}\right)\left(u_{m-1} - u_{\mathrm{merging}}\right), \end{aligned} \tag{3.2.18}$$

where derivative $\dot{r}_m$ is defined as $\dot{r}_m := \frac{dr_m}{dd_{\mathrm{merging}}}\dot{d}_{\mathrm{merging}}$ as r_m is now a function of d_{merging} and d_{merging} depends on time. Similarly, $\ddot{r}_m := \frac{d\dot{r}_m}{dv_{m-1}}\dot{v}_{m-1} + \frac{d\dot{r}_m}{dv_{\mathrm{merging}}}\dot{v}_{\mathrm{merging}}$ and $\dddot{r}_m := \frac{d\ddot{r}_m}{da_{m-1}}\dot{a}_{m-1} + \frac{d\ddot{r}_m}{da_{\mathrm{merging}}}\dot{a}_{\mathrm{merging}}$.

These derivatives contain v_{m-1}, a_{m-1}, and u_{m-1} since the virtual reference car of the merging car is a copy of vehicle $m-1$.

By substituting (3.2.18) into (3.2.5), since only $\ddot{r}_m$ and $\dddot{r}_m$ are nonzero, for vehicle m

$$\dddot{e}_m = -\frac{1}{\tau}\ddot{e}_m + \frac{1}{\tau}\bar{u}_m - \beta\left(\frac{1}{\tau}\right)\left(\frac{L_{\text{gap}}}{r + h\bar{v}_{\text{platoon}} + L_{\text{com}}}\right)(u_{m-1} - u_{\text{merging}})$$

and for vehicle $m-1$

$$\dddot{e}_{m-1} = -\frac{1}{\tau}\ddot{e}_{m-1} + \frac{1}{\tau}\bar{u}_{m-1} - (1-\beta)\left(\frac{1}{\tau}\right)\left(\frac{L_{\text{gap}}}{r + h\bar{v}_{\text{platoon}} + L_{\text{com}}}\right)(u_{m-1} - u_{\text{merging}}).$$

This then leads to the error dynamics for vehicles m and $m-1$ given by

$$\begin{aligned}
\begin{pmatrix} \dot{e}_m \\ \ddot{e}_m \\ \dddot{e}_m \end{pmatrix} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix}\begin{pmatrix} e_m \\ \dot{e}_m \\ \ddot{e}_m \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix}\bar{u}_m \\
&\quad - \beta\begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix}\left(\frac{L_{\text{gap}}}{r + h\bar{v}_{\text{platoon}} + L_{\text{com}}}\right)(u_{m-1} - u_{\text{merging}}), \\
\begin{pmatrix} \dot{e}_{m-1} \\ \ddot{e}_{m-1} \\ \dddot{e}_{m-1} \end{pmatrix} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix}\begin{pmatrix} e_{m-1} \\ \dot{e}_{m-1} \\ \ddot{e}_{m-1} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix}\bar{u}_{m-1} \\
&\quad - (1-\beta)\begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix}\left(\frac{L_{\text{gap}}}{r + h\bar{v}_{\text{platoon}} + L_{\text{com}}}\right)(u_{m-1} - u_{\text{merging}}),
\end{aligned}$$

or, when using $x_m = [e_m, \dot{e}_m, \ddot{e}_m]^T$ and matrices A and B, by

$$\dot{x}_m = Ax_m + B\bar{u}_m - \beta B\left(\frac{L_{\text{gap}}}{r + h\bar{v}_{\text{platoon}} + L_{\text{com}}}\right)(u_{m-1} - u_{\text{merging}}), \tag{3.2.19}$$

$$\dot{x}_{m-1} = Ax_{m-1} + B\bar{u}_{m-1} - (1-\beta)B\left(\frac{L_{\text{gap}}}{r + h\bar{v}_{\text{platoon}} + L_{\text{com}}}\right)(u_{m-1} - u_{\text{merging}}). \tag{3.2.20}$$

The error dynamics of (3.2.6) still apply to all remaining vehicles. The filter of (3.2.7) applies to all vehicles of the platoon including vehicles m and $m-1$. The derivative in (3.2.18) can be rewritten to

$$\dot{R}_m = AR_m + B\left(\frac{L_{\text{gap}}}{r + h\bar{v}_{\text{platoon}} + L_{\text{com}}}\right)(u_{m-1} - u_{\text{merging}}) \tag{3.2.21}$$

by using $R_m = [r_m, \dot{r}_m, \ddot{r}_m]^T$. Consider now the lumped states

$$\begin{aligned} X &= [e_1, \dot{e}_1, \ddot{e}_1, \ldots, e_n, \dot{e}_n, \ddot{e}_n]^T, \\ U &= [u_1, \ldots, u_n]^T, \end{aligned}$$

and

$$R_m = [r_m, \dot{r}_m, \ddot{r}_m]^T.$$

By using these lumped states, the dynamics in (3.2.6), (3.2.7) and (3.2.19) to (3.2.21), and the controller defined in (3.1.3), the closed-loop platoon dynamics become

(closed-loop platoon dynamics with standstill distance dependent on merging vehicle)

$$\begin{pmatrix} \dot{X} \\ \dot{U} \\ \dot{R}_m \end{pmatrix} = \begin{pmatrix} I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T & -J_m \otimes \frac{L_{\text{gap}}}{r+h\bar{v}_{\text{platoon}}+L_{\text{com}}} B & \mathrm{O}_{3n\times 3} \\ \hat{\mathcal{L}} \otimes \frac{1}{h} k^T & A_u & \mathrm{O}_{n\times 3} \\ \mathrm{O}_{3\times 3n} & I_n(m-1,\,:) \otimes \frac{L_{\text{gap}}}{r+h\bar{v}_{\text{platoon}}+L_{\text{com}}} B & A \end{pmatrix} \begin{pmatrix} X \\ U \\ R_m \end{pmatrix}$$
$$+ \begin{pmatrix} \mathrm{O}_{3n\times 1} & J_m(:,\, m-1) \otimes \frac{L_{\text{gap}}}{r+h\bar{v}_{\text{platoon}}+L_{\text{com}}} B \\ \beta B_u & \mathrm{O}_{n\times 1} \\ \mathrm{O}_{3\times 1} & -\frac{L_{\text{gap}}}{r+h\bar{v}_{\text{platoon}}+L_{\text{com}}} B \end{pmatrix} \begin{pmatrix} u_0 \\ u_{\text{merging}} \end{pmatrix}, \tag{3.2.22}$$

where J_m is an $n \times n$ zero matrix with the exception that $J_m(m,\, m-1) = \beta$ and $J_m(m-1,\, m-1) = 1-\beta$. Note that substituting the expression for $\ddot{r}_m$ in (3.2.18) into (3.2.10) gives the same result.

In this case, the merging car is still assumed to arrive at the gap in the platoon at time $\tilde{t}_{\text{ramp}}$ and is hence modeled by the dynamics in (3.2.17) with $u_{0,\text{merging}} = u_{m-1}$ as is described in section 3.2.4.

3.2.6 The merge

The dynamics given in (3.2.17) and in (3.2.22) model respectively the merging car and the platoon from time t_{start} until time $\tilde{t}_{\text{ramp}}$. At time $\tilde{t}_{\text{ramp}}$, the standstill distance r_{merging} of the merging vehicle equals the default value r, but the actual distance between the merging vehicle and vehicle $m-1$ is not necessarily equal to the desired distance. In addition, at $\tilde{t}_{\text{ramp}}$ the merging vehicle might still be driving on the entrance ramp. Therefore, after time $\tilde{t}_{\text{ramp}}$ the dynamics in (3.2.22) continue to describe the platoon and (3.2.17) with $r_{\text{merging}}(t) = r$ is used to model the merging vehicle.

At some point, however, all vehicles should be modeled by (3.1.1). There are two possible moments to switch to this model. When $d_{\text{merging}} = r + h\bar{v}_{\text{platoon}}$, the merging vehicle drives at the desired distance behind vehicle $m-1$. The standstill distance r_m of vehicle m then equals the desired value of $r+L_{\text{gap}}$, but the distance between vehicle m and the merging vehicle might not yet be large enough. Nevertheless, at this moment it is possible to model all vehicles by (3.1.1). The merging vehicle is virtually inserted into the platoon, but does not yet merge and continues driving in a separate lane until it is able to merge. It is also possible that the merging vehicle is able to merge while $d_{\text{merging}} < r + h\bar{v}_{\text{platoon}}$. This is caused by how the model checks whether the merging vehicle is able to merge, which will be discussed below. In this case, the merging vehicle merges into the platoon, and the new platoon consisting of $n+1$ vehicles is modeled by (3.1.1). It might happen that $d_{\text{merging}} > r + h\bar{v}_{\text{platoon}}$, but then the first scenario applies since at some time d_{merging} must have been equal to $r + h\bar{v}_{\text{platoon}}$.

The actual merging takes place at time t_{merge}, the moment the merging car detects enough distance in front and behind. This distance is now denoted by d_{merge} and could, for example, be chosen to be the standstill distance r of the merging vehicle or the desired distance $d^d_{\text{merging}}(t) = r + hv_{\text{merging}}(t)$ between the merging vehicle and its neighbors, which consists of a constant term with the standstill distance r and a term for the time gap h that depends on the velocity $v_{\text{merging}}(t)$. The safest option would be to use at least the desired distance or an even larger distance, but then it might take too long before the vehicle can merge. On the other hand, using the standstill distance when vehicles are driving at a high velocity is risky. The solution is to let d_{merge} depend on how far the merging vehicle is from the end of the acceleration lane q_{end}. This distance can be defined as

$$\delta_{\text{end}}(t) \coloneqq q_{\text{end}} - q_{\text{merging}}(t)$$

and is assumed to be known in order to ensure that the vehicle merges before it reaches q_{end}.

When the merging vehicle is at q_{ramp}, the distance d_{merge} should be equal to the desired distance $d^d_{\text{merging}}(t) = r_{\text{merging}} + hv_{\text{merging}}(t)$ of the merging vehicle and when the merging vehicle is at q_{end}, it should be equal to the standstill distance r. By using a linear function, d_{merge} can be defined as

$$d_{\text{merge}}(\delta_{\text{end}}) = \frac{hv_{\text{merging}}}{q_{\text{end}} - q_{\text{ramp}}} \left(\delta_{\text{end}} - q_{\text{end}} + q_{\text{ramp}}\right) + r + hv_{\text{merging}}$$

for $t_{\text{ramp}} < t < t_{\text{merge}}$, and hence time t_{merge} can be defined as the time $t > t_{\text{ramp}}$ where

$$d_{\text{merging}} \coloneqq q_{m-1} - q_{\text{merging}} - L_{\text{merging}} \geq d_{\text{merge}}(\delta_{\text{end}})$$

and

$$d_m := q_{\mathrm{merging}} - q_m - L_m \geq d_{\mathrm{merge}}(\delta_{\mathrm{end}})$$

first hold. The merging vehicle merges into the platoon at time t_{merge} and continues driving on the highway as part of the platoon. Note that the actual lane change that is part of the merge is not explicitly modeled. Instead it is assumed that if the merging vehicle is ready to merge before it reaches q_{end}, the merge will be successful.

3.3 Merging at a fixed location

Section 3.2 provides a way to model the merge of a vehicle into a platoon on the highway: a vehicle m is selected, the platoon creates a gap between vehicles m and $m-1$, the merging vehicle aligns itself with the gap, and the vehicle merges into the platoon. This model, however, is based on the estimation $\tilde{t}_{\mathrm{ramp}}$ of the time where the merging vehicle reaches the beginning of the acceleration lane q_{ramp} and should merge into the platoon. It is not guaranteed that the merging vehicle is actually able to merge at this time or that the merging vehicle is even driving in the acceleration lane. Any disturbances could cause the actual time of the merge or the time that the merging vehicle arrives at q_{ramp} to deviate from the estimated time $\tilde{t}_{\mathrm{ramp}}$ with the possible result that the vehicle cannot merge into the platoon at all.

In this section, the model of section 3.2 for merging at a fixed time is adapted in order to obtain a model that takes into account that a vehicle is only able to merge while it is driving on the acceleration lane and hence should provide a more robust way to model the merge of a vehicle onto the highway. The new model will be largely the same as the model of section 3.2, but is designed such that the merging vehicle and the platoon both aim to complete the merge at a fixed location. As mentioned before, the merging vehicle aims to merge as soon as possible and hence the beginning of the acceleration lane q_{ramp} is chosen as the fixed merge location. If any problems arise that result in the merging vehicle not being able to merge at this time, the entire length of the acceleration lane can be used to resolve these issues.

The merge of an automated vehicle into a platoon that is driving on the highway is modeled as follows:

1. At time t_{start}, the estimate $\tilde{t}_{\mathrm{ramp}}$ is computed as given in (3.2.2) and a vehicle m is selected according to (3.2.3).
2. During the time interval $[t_{\mathrm{start}}, t_{\mathrm{ramp}}]$ the platoon is controlled as will be discussed below in section 3.3.1 such that it creates a gap between vehicles m and $m-1$ of the platoon. The platoon is modeled by (3.3.5) during this time.
3. During the time interval $[t_{\mathrm{start}}, t_{\mathrm{ramp}}]$ the merging vehicle is controlled such that it aligns itself with the gap in the platoon. This is done as is described in section 3.2.4 and hence the merging vehicle is modeled by (3.2.17) during this time.
4. At time t_{ramp} the merging vehicle is virtually inserted into the platoon, and hence during the time interval $[t_{\mathrm{ramp}}, t_{\mathrm{merge}}]$ the platoon, including the merging vehicle, is modeled by (3.2.10).
5. At time t_{merge} the merging vehicle merges into the platoon when it detects that there enough distance between itself and vehicles m and $m-1$. Whether the merging vehicle is able to merge is checked in the same way as in section 3.2.6.
6. After time t_{merge}, the new platoon that now consists of $n+1$ vehicles is modeled by (3.1.1).

To be complete, section 3.3.1 states all models used during the time interval $[t_{\mathrm{start}}, t_{\mathrm{ramp}}]$ and section 3.3.2 describes the model used during $[t_{\mathrm{ramp}}, t_{\mathrm{merge}}]$.

3.3.1 Arriving at the acceleration lane

As mentioned in the previous section, during the time interval $[t_{\mathrm{start}}, t_{\mathrm{ramp}}]$, the merging vehicle is modeled by the closed-loop dynamics of (3.2.17):

(closed-loop dynamics for the merging vehicle for $\mathbf{t}_{\text{start}} \leq \mathbf{t} \leq \mathbf{t}_{\text{ramp}}$)

$$\begin{aligned}\begin{pmatrix}\dot{x}_{\text{merging}}\\ \dot{u}_{\text{merging}}\\ \dot{R}_{\text{merging}}\end{pmatrix} &= \begin{pmatrix}A - Bk^T & O_{3\times 1} & [O_{3\times 2} \;\; B_r(:,1)]\\ \frac{1}{h}k^T & -\frac{1}{h} & O_{1\times 3}\\ O_{3\times 3} & O_{3\times 1} & I_{(1),3}\end{pmatrix}\begin{pmatrix}x_{\text{merging}}\\ u_{\text{merging}}\\ R_{\text{merging}}\end{pmatrix}\\ &\quad + \begin{pmatrix}O_{3\times 1} & B_r(:,2)\\ \frac{1}{h} & O_{1\times 1}\\ O_{3\times 1} & I_3(:,3)\end{pmatrix}\begin{pmatrix}u_{0,\,\text{merging}}\\ \dddot{r}_{\text{merging}}\end{pmatrix},\end{aligned} \tag{3.3.1}$$

where the input $u_{0,\,\text{merging}}$ is defined as

$$u_{0,\,\text{merging}} := u_{m-1},$$

since a copy of vehicle $m-1$ is used as the virtual reference vehicle of the merging car. The standstill distance r_{merging} of the merging vehicle is defined as is described in section 3.2.3 as a polynomial that satisfies the constraints

$$r_{\text{merging}}(t_{\text{start}}) = q_{m-1}(t_{\text{start}}) - q_{\text{start}} - L_{\text{merging}} - hv_{\text{merging}}(t_{\text{start}})$$

and

$$r_{\text{merging}}(\tilde{t}_{\text{ramp}}) = r,$$

and whose first two derivatives equal zero at t_{start} and $\tilde{t}_{\text{ramp}}$.

Next, the platoon should be controlled during $[t_{\text{start}},\, t_{\text{ramp}}]$ such that it creates a gap. When the merging vehicle arrives at the beginning of the acceleration lane, the platoon should have created a gap between vehicles m and $m-1$, which were chosen at time t_{start}. As the merging vehicle cannot merge before it reaches q_{ramp} at time t_{ramp}, the platoon can gradually increase the standstill distance between vehicles m and $m-1$ during the interval $[t_{\text{start}}, t_{\text{ramp}}]$. This could be modeled by (3.2.10) using a fixed standstill distance function $r_m(t)$ defined at time t_{start} according to section 3.2.3, but it is also possible to let the standstill distance of vehicle m depend on the position of the merging vehicle using a linear function, similar to what was done in section 3.2.5.

When the merging vehicle enters the entrance ramp, all standstill distances $r_i,\ i \in \mathcal{V}$, should still be equal to the default value r, but when the vehicle reaches the beginning of the acceleration lane, the standstill distance r_m of the chosen vehicle m should have been increased by an additional distance L_{gap}, such that the merging vehicle will be able to merge into the platoon between vehicles m and $m-1$. Ideally, L_{gap} would equal $r + hv_{\text{merging}} + L_{\text{merging}}$ such that when the distance between vehicles m and $m-1$ is increased by L_{gap} and the merging vehicle merges into the gap, it immediately drives at the correct distance from vehicle m and from vehicle $m-1$. To avoid complicated derivatives, however, it might be convenient to choose $L_{\text{gap}} = r + h\bar{v}_{\text{platoon}} + L_{\text{merging}}$ instead, which uses the constant platoon velocity $\bar{v}_{\text{platoon}}$. Since $\bar{v}_{\text{platoon}}$ is the desired velocity of all vehicles, in steady state all velocities will be equal to $\bar{v}_{\text{platoon}}$.

As it was assumed that for all vehicles the distance δ to the beginning of the acceleration lane q_{ramp} is known, when vehicle m is making a gap, the standstill distance r_m can be defined as a function of δ_{merging}. When the distance δ_{merging} is equal to the initial value $\delta_{\text{merging}}(t_{\text{start}}) = q_{\text{ramp}} - q_{\text{start}}$, the standstill distance r_m should be equal to the default distance r. When the distance δ_{merging} is zero, then $q_{\text{merging}} = q_{\text{ramp}}$ and hence r_m should be equal to $r + L_{\text{gap}}$. Standstill distance r_m should therefore satisfy

$$r_m(\delta_{\text{merging}} = q_{\text{ramp}} - q_{\text{start}}) = r$$

and

$$r_m(\delta_{\text{merging}} = 0) = r + L_{\text{gap}}.$$

This results in the following linear function:

$$r_m(\delta_{\text{merging}}) = \left(\frac{-L_{\text{gap}}}{q_{\text{ramp}} - q_{\text{start}}}\right)\delta_{\text{merging}} + r + L_{\text{gap}}, \tag{3.3.2}$$

defined for $q_{\text{ramp}} - q_{\text{start}} > \delta_{\text{merging}} > 0$ or for $t_{\text{start}} < t < t_{\text{ramp}}$, as t_{ramp} was defined as the time where the merging vehicle reaches q_{ramp} and $\delta_{\text{merging}}(t_{\text{ramp}}) = 0$.

Since the distance δ_{merging} was defined as $\delta_{\text{merging}}(t) := q_{\text{ramp}} - q_{\text{merging}}(t)$ and q_{ramp} is fixed, the vehicle dynamics of the merging vehicle (see also (2.3.1)),

$$\begin{pmatrix} \dot{q}_{\text{merging}} \\ \dot{v}_{\text{merging}} \\ \dot{a}_{\text{merging}} \end{pmatrix} = \begin{pmatrix} v_{\text{merging}} \\ a_{\text{merging}} \\ -\frac{1}{\tau} a_{\text{merging}} + \frac{1}{\tau} u_{\text{merging}} \end{pmatrix}, \tag{3.3.3}$$

can be used to find the derivatives of (3.3.2). As r and L_{gap} are constants, the first time derivative of (3.3.2) becomes

$$\dot{r}_m = \left(\frac{-L_{\text{gap}}}{q_{\text{ramp}} - q_{\text{start}}} \right) (-v_{\text{merging}}),$$

where derivative $\dot{r}_m$ is defined as $\dot{r}_m := \frac{dr_m}{d\delta_{\text{merging}}} \dot{\delta}_{\text{merging}}$ as r_m is now a function of δ_{merging} and δ_{merging} depends on time. The second and third time derivatives are then given by

$$\ddot{r}_m = \left(\frac{-L_{\text{gap}}}{q_{\text{ramp}} - q_{\text{start}}} \right) (-a_{\text{merging}})$$

and

$$\begin{aligned} \dddot{r}_m &= \left(\frac{-L_{\text{gap}}}{q_{\text{ramp}} - q_{\text{start}}} \right) \left(\frac{1}{\tau} a_{\text{merging}} - \frac{1}{\tau} u_{\text{merging}} \right) \\ &= -\frac{1}{\tau} \ddot{r}_m + \frac{1}{\tau} \left(\frac{L_{\text{gap}}}{q_{\text{ramp}} - q_{\text{start}}} \right) u_{\text{merging}}. \end{aligned} \tag{3.3.4}$$

Substituting (3.3.4) into the closed-loop platoon dynamics of (3.2.10) results in

(closed-loop platoon dynamics for $\mathbf{t}_{\text{start}} \leq \mathbf{t} \leq \mathbf{t}_{\text{ramp}}$)

$$\begin{aligned} \begin{pmatrix} \dot{X} \\ \dot{U} \\ \dot{R}_m \end{pmatrix} &= \begin{pmatrix} I_n \otimes A - \hat{\mathcal{L}} \otimes B k^T & \mathrm{O}_{3n \times n} & \mathrm{O}_{3n \times 3} \\ \hat{\mathcal{L}} \otimes \frac{1}{h} k^T & A_u & \mathrm{O}_{n \times 3} \\ \mathrm{O}_{3 \times 3n} & \mathrm{O}_{3 \times n} & A \end{pmatrix} \begin{pmatrix} X \\ U \\ R_m \end{pmatrix} \\ &+ \begin{pmatrix} \mathrm{O}_{3n \times 1} & J(:, m) \otimes \frac{-L_{\text{gap}}}{q_{\text{ramp}} - q_{\text{start}}} B \\ \beta B_u & \mathrm{O}_{n \times 1} \\ \mathrm{O}_{3 \times 1} & \frac{L_{\text{gap}}}{q_{\text{ramp}} - q_{\text{start}}} B \end{pmatrix} \begin{pmatrix} u_0 \\ u_{\text{merging}} \end{pmatrix}. \end{aligned} \tag{3.3.5}$$

In addition to u_0, also the desired acceleration of the merging vehicle u_{merging} is now required as an external input.

Note that, in (3.3.5), X and U do no longer depend on R_m. This can be seen easily by considering the error dynamics of (3.2.5) for vehicle m:

$$\dddot{e}_m = -\frac{1}{\tau} \ddot{e}_m + \frac{1}{\tau} \bar{u}_m + \beta \left[-\frac{1}{\tau} \ddot{r}_m - \dddot{r}_m \right]. \tag{3.3.6}$$

If $\dddot{r}_m$ is replaced by (3.3.4) in these error dynamics, the first term of (3.3.4) cancels against the $\ddot{r}_m$ in the last term of (3.3.6).

3.3.2 The merge

From time t_{start} until t_{ramp}, the merging vehicle is modeled by (3.3.1) and the platoon by (3.3.5) such that, ideally, the merging vehicle will be able to merge with the platoon at time t_{ramp}. However, most likely, the merging vehicle will not exactly be aligned with the gap at time t_{ramp}. Therefore, finally, the merging vehicle should be controlled such that it will be aligned with the gap and it should be checked whether it is safe to merge before the actual merge can take place at time t_{merge}.

While the merging vehicle might not yet be exactly aligned with the gap in the platoon at time t_{ramp}, it will be close enough to the gap that the merging vehicle can communicate to and act as part of the platoon. Therefore, the merging vehicle is added to the platoon, though they are still driving in separate lanes, and the dynamics in (3.2.10), which are

(closed-loop platoon dynamics for $\mathbf{t}_{\mathrm{ramp}} \leq \mathbf{t} \leq \mathbf{t}_{\mathrm{merge}}$)

$$\begin{pmatrix} \dot{X} \\ \dot{U} \\ \dot{R}_m \end{pmatrix} = \begin{pmatrix} I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T & \mathrm{O}_{3n\times n} & J(:,m) \otimes [\mathrm{O}_{3\times 2} \;\; B_r(:,1)] \\ \hat{\mathcal{L}} \otimes \frac{1}{h} k^T & A_u & \mathrm{O}_{n\times 3} \\ \mathrm{O}_{3\times 3n} & \mathrm{O}_{3\times n} & I_{(1),3} \end{pmatrix} \begin{pmatrix} X \\ U \\ R_m \end{pmatrix} + \begin{pmatrix} \mathrm{O}_{3n\times 1} & J(:,m) \otimes B_r(:,2) \\ \beta B_u & \mathrm{O}_{n\times 1} \\ \mathrm{O}_{3\times 1} & I_3(:,3) \end{pmatrix} \begin{pmatrix} u_0 \\ \dddot{r}_m \end{pmatrix}, \tag{3.3.7}$$

now cover both the merging vehicle and the platoon. This ensures that the merging vehicle will be exactly aligned with the gap.

While the standstill distance between the merging vehicle and vehicle $m-1$ is already equal to the default value r at time t_{ramp} (see section 3.3.1), this is not necessarily true for the standstill distance between vehicle m and the merging vehicle. Therefore, the standstill distance r_m is increased or decreased from the initial value $d_m(t_{\mathrm{ramp}}) - hv_m(t_{\mathrm{ramp}})$ at t_{ramp} to the default value r, where d_m is the distance between the merging vehicle and vehicle m and hence $d_m(t_{\mathrm{ramp}}) = q_{\mathrm{ramp}} - q_m(t_{\mathrm{ramp}}) - L_m$. This is done, again as in section 3.2.3, by finding a 5th-degree polynomial $r_m(t)$ that satisfies the constraints

$$r_m(t_{\mathrm{ramp}}) = q_{\mathrm{ramp}} - q_m(t_{\mathrm{ramp}}) - L_m - hv_m(t_{\mathrm{ramp}})$$

and

$$r_m(t_{\mathrm{ramp}} + t_{\mathrm{align}}) = r,$$

and for which the first two derivatives equal zero at $t_{\mathrm{ramp}} + t_{\mathrm{align}}$ and are continuous at t_{ramp}. Here, t_{align} is the amount of time given for the final adjustments. Its value can be small as t_{align} is mainly used to avoid discontinuities in $r_m(t)$ and abrupt changes in velocity for the vehicles.

The actual merging can then take place at time t_{merge}, the moment the merging car detects enough distance in front and behind. This distance is again denoted by d_{merge} and is again defined as the linear function

$$d_{\mathrm{merge}}(\delta_{\mathrm{end}}) = \frac{hv_{\mathrm{merging}}}{q_{\mathrm{end}} - q_{\mathrm{ramp}}} (\delta_{\mathrm{end}} - q_{\mathrm{end}} + q_{\mathrm{ramp}}) + r + hv_{\mathrm{merging}}$$

for $t_{\mathrm{ramp}} < t < t_{\mathrm{merge}}$, and hence time t_{merge} can be defined as the time $t > t_{\mathrm{ramp}}$ where

$$d_{\mathrm{merging}} \coloneqq q_{m-1} - q_{\mathrm{merging}} - L_{\mathrm{merging}} \geq d_{\mathrm{merge}}(\delta_{\mathrm{end}})$$

and

$$d_m \coloneqq q_{\mathrm{merging}} - q_m - L_m \geq d_{\mathrm{merge}}(\delta_{\mathrm{end}})$$

first hold. The merging vehicle merges into the platoon at time t_{merge} and continues driving on the highway as part of the platoon.

4. Methods

Both the models of section 3.2 and section 3.3 are implemented in MATLAB in order to run some simulations of a vehicle merging into a platoon on the highway. Each model is applied to the situation described by figure 3.1. At first, a platoon is driving on the highway with a constant velocity. Then at time t_{start} a vehicle enters the entrance ramp to the highway at q_{start}. The platoon and the merging vehicle are controlled such that the merge can take place at time t_{merge} before the merging vehicle reaches the end of the acceleration lane. The vehicle m in front of which the merging vehicle merges into the platoon is chosen as is described in (3.2.3). The merging vehicle merges at time t_{merge} when it detects enough distance between vehicles m and $m-1$. The model of section 3.3, which we will call the closed-loop model from now on, is given by (3.1.1), (3.3.1), (3.3.5), and (3.3.7). This model aims to let the merging vehicle merge into the platoon when it enters the acceleration lane at q_{ramp}. It was designed this way such that final adjustments can be made while driving the entire length of the acceleration lane. In the open-loop model, which is given in section 3.2, the merging vehicle was to merge at time $\tilde{t}_{\text{ramp}}$, which is the estimation of t_{ramp} given in (3.2.2). The dynamics in (3.2.17) and (3.2.22), however, only describe the situation during the time interval $[t_{\text{start}}, \tilde{t}_{\text{ramp}}]$. Outside of this interval the merging vehicle is part of the platoon, and hence (3.1.1) can be used. Since it is not guaranteed that the merging vehicle is driving in the acceleration lane at $\tilde{t}_{\text{ramp}}$, this model is called the open-loop model, even though it does contain some feedback elements.

For the simulations, we consider the merge of a vehicle onto a Dutch highway where the maximum speed is 120km/h. According to the guidelines for the design of Dutch highways, the acceleration lane should in this case have a length of 350m, and the minimum distance needed to accelerate is 125m based on an acceleration of 1m/s^2 and an initial velocity of 70km/m^2 [14, chapter 6]. Therefore, we choose $\bar{v}_{\text{platoon}} = 120$km/h $= 33.33$m/s, $v_{\text{merging}}(t_{\text{start}}) = 70$km/h $= 19.44$m/s, $a_{\text{ramp}} = 1$m/s^2, and a length of 450m for the entrance ramp and 350m for the acceleration lane. Starting the simulation at time $t_{\text{start}} = 0$s with $q_{\text{start}} = 0$m gives $q_{\text{ramp}} = 450$m and $q_{\text{end}} = 800$m. It is assumed that the platoon consists of 10 vehicles and is homogeneous, which means that all vehicles use the same values for the length, the drive-line dynamics time constant τ, and the time gap h. Also the default standstill distance r is the same for all vehicles. For the vehicle lengths, we use $L_i = 5$m, $i = 1, \ldots, n$, and $L_{\text{merging}} = 5$m. In [21], values of $\tau = 0.1$s, $r = 2$m, and $h = 0.6$s are used. For creating a gap in the platoon, we use $L_{\text{gap}} = L_{\text{merging}} + r + h\bar{v}_{\text{platoon}}$. In the open-loop model, L_{com} is implemented as the distance between the merging vehicle and vehicle $m-1$ at time t_{start}: $L_{\text{com}} = -(q_{m-1}(t_{\text{start}}) - q_{\text{start}} - L_{\text{merging}})$. In the closed-loop model, we use $t_{\text{align}} = 5$s. The virtual reference vehicle of the platoon was assumed to be uncontrolled, so $u_0(t) = 0$m/s^2, and for the merging vehicle we use $u_{0,\text{merging}}(t) = u_{m-1}(t)$. The values of α, β, and the controller gain vector $k = [k_1, k_2, k_3]^T$ are chosen according to section 2.9 such that $\alpha \in (0, 1]$, $\beta \in \left(\frac{1}{2}, 1\right]$, and

$$\begin{cases} k_1 > 0 \\ k_2 > \frac{k_1 \tau}{\min_i\{\lambda_i k_3 + 1\}} \\ k_3 > -\frac{1}{\max_i\{\lambda_i\}} \end{cases}$$

For k we use the same values as in [21], $k^T = [0.2, 1, 0]$, and we choose $\alpha = 0.8$ and $\beta = 0.8$. As mentioned before in section 2.6, we use $k_3 = 0$ for all results. This is possible since all eigenvalues λ_i of the matrix $\hat{\mathcal{L}}$ defined in (2.1.10) and (2.1.11) are positive and hence $-\frac{1}{\max_i\{\lambda_i\}}$ is negative. Using $k_3 = 0$ also simplifies the constraints for k_1 and k_2 to $k_1 > 0$ and $k_2 > k_1\tau$.

The values mentioned above are summarized in table 4.1 and are used in all results of chapter 5 unless it is mentioned otherwise.

Table 4.1: Default values of parameters.

	parameters			
	description	symbol	value	unit
platoon	size	n	10	-
	velocity	$\bar{v}_{\text{platoon}}$	120	km/h
vehicle	length	$L_i \,/\, L_{\text{merging}}$	5	m
	drive-line dynamics constant	τ	0.1	s
	standstill distance	r	2	m
	time gap	h	0.6	s
merging	initial velocity	$v_{\text{merging}}(t_{\text{start}})$	70	km/h
	acceleration	a_{ramp}	1	m/s^2
	gap size	L_{gap}	$L_{\text{merging}} + r + h\bar{v}_{\text{platoon}}$	m
	communication distance	L_{com}	$-(q_{m-1}(t_{\text{start}}) - q_{\text{start}} - L_{\text{merging}})$	m
weights	communication topology	α	0.8	-
	error	β	0.8	-
controller	gain vector	k	$[k_1, k_2, k_3]^T$	-
	gain values	k_1	0.2	-
		k_2	1	-
		k_3	0	-
highway	beginning of entrance ramp	q_{start}	0	m
	beginning of acceleration lane	q_{ramp}	450	m
	end of acceleration lane	q_{end}	800	m
simulation	start time	t_{start}	0	s
		t_{align}	5	s
input	platoon	$u_0(t)$	0	m/s^2
	merging vehicle	$u_{0,\text{merging}}(t)$	$u_{m-1}(t)$	m/s^2

5. Results and discussion

In this chapter, the models that were designed in chapter 3 are simulated and the results are reported and discussed. The open-loop and closed-loop models are implemented as described in chapter 4. The first simulations are performed with the default settings as given in chapter 4. These values are summarized in table 4.1. After that some of the parameters are changed and more simulations are done.

First of all, we consider the closed-loop model with the default parameters given in table 4.1. Some plots are given in figure 5.1 showing the error $e_i(t)$, the distance $d_i(t)$, the velocity $v_i(t)$, and the acceleration $a_i(t)$ of several vehicles. Recall that the distance of a vehicle i was defined as the distance between vehicle i and vehicle $i-1$ and the error as

$$e_i(t) = \beta\left[d_i(t) - r - hv_i(t)\right] + (1-\beta)\left[d_{i+1}(t) - r - hv_i(t)\right].$$

A platoon of size $n = 10$ was simulated, but figure 5.1 only shows the merging vehicle and vehicles $m-2$, $m-1$, m, and $m+1$ of the platoon. In this case, the merging vehicle chooses to merge in front of vehicle $m = 3$ and hence figure 5.1 only shows the first four vehicles of the platoon. In this simulation, the merging vehicle reaches the beginning of the acceleration lane q_{ramp} at time $t_{\mathrm{ramp}} = 16.06$s, merges into the platoon at time $t_{\mathrm{merge}} = 17.62$s, and reaches the end of the acceleration lane q_{end} at time $t_{\mathrm{end}} = 26.45$s. The plots in figure 5.1 show that the model is working as expected. The errors converge to zero for all vehicles, the distances all converge to the same value of $r + h\bar{v}_{\mathrm{platoon}} = 22$m, the velocities all converge to the platoon velocity $\bar{v}_{\mathrm{platoon}} = 33.33$m/s, and also the accelerations converge to zero. The distance of the merging vehicle to vehicle $m-1$ is initially negative because the merging vehicle starts in front of the platoon. A negative distance between two vehicles would normally mean that these vehicles crashed, but there are some exceptions. The distance of the merging vehicle is defined with respect to vehicle $m-1$ of the platoon and is allowed to be negative until time t_{merge} when it merges into the platoon. Also the distance of the first vehicle is allowed to be negative, since it drives behind a virtual car, and, between time t_{ramp} and t_{merge}, the distance of vehicle m to the merging vehicle could be negative. Note that during this time, figure 5.1 would not clearly show a crash between vehicles m and $m-1$ because the distance of vehicle m is here defined with respect to the merging vehicle which is still driving in the acceleration lane. For this reason, additional checks were implemented in MATLAB to detect any crashes. The distance of the merging car in figure 5.1 shows some overshoot at time t_{merge}. This means that when the merging vehicle merges into the platoon it drives closer to vehicle m than to vehicle $m-1$ and that its velocity is slightly higher than the platoon velocity. Large overshoots should be avoided, since this could cause vehicles to crash. This overshoot could be reduced by increasing the value of the controller gain vector k, but this would increase the peak in acceleration for the merging vehicle, and hence might not be desirable. Figure 5.1 also shows that the error of vehicle 2, 3, and the merging vehicle and the distance of vehicle 3 is discontinuous at time $t = 16.06$s. This is as expected, since the merging vehicle reaches q_{ramp} at this time. The merging vehicle is then virtually inserted into the platoon, meaning that it acts as part of the platoon but is still driving in the acceleration lane, and hence the error state of vehicle $m-1 = 2$, the merging vehicle, and vehicle $m = 3$ are redefined and are discontinuous at this point. Since the error of a vehicle i is defined based on the vehicle directly in front of and the vehicle directly behind vehicle i, nothing changes for the remaining vehicles in the platoon. The distance of vehicle m is discontinuous at time t_{ramp} for the same reason.

For the same simulation, the standstill distance r_i and its first two derivatives are given in figure 5.2 for the first four vehicles of the platoon and the merging vehicle. For vehicle $m = 3$, the standstill distance increases until time $t_{\mathrm{ramp}} = 16.06$s, where it is discontinuous since at that time the merging vehicle

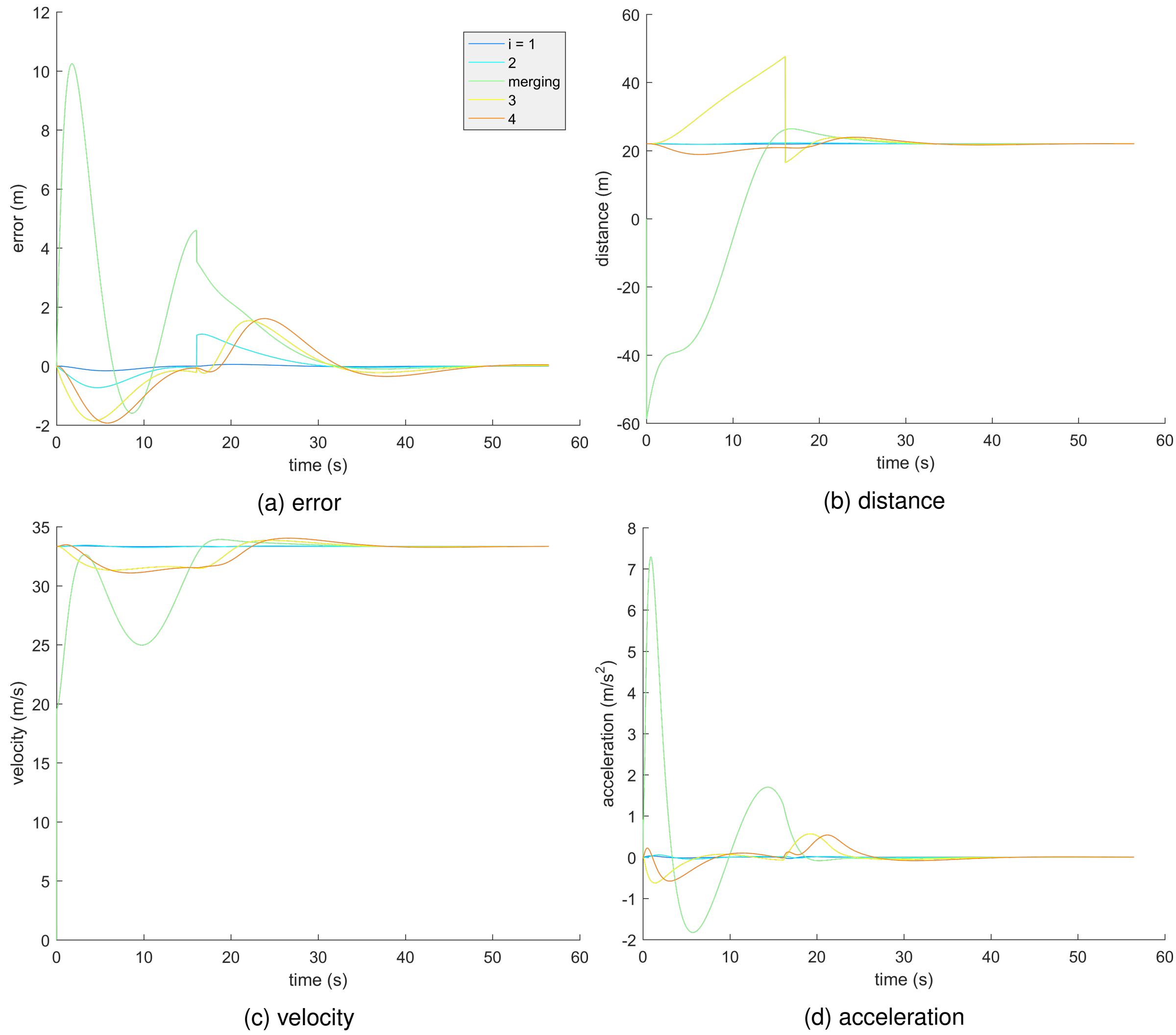


Figure 5.1: The error $e_i(t)$, distance $d_i(t)$, velocity $v_i(t)$, and acceleration $a_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for a simulation of the closed-loop model with the default settings. In this case, $t_{\text{ramp}} = 16.06$s, $t_{\text{merge}} = 17.62$s, and $t_{\text{end}} = 26.45$s.

starts acting as a part of the platoon, and then converges to the default value $r = 2$m. Its derivatives are continuous, which is consistent to how r_m was designed. For the merging vehicle, the standstill distance increases gradually to the default value $r = 2$m as expected. It is, however, discontinuous at t_{ramp}, though it is hard to see in figure 5.2, and also its derivatives are discontinuous. This is caused by the fact that $r_{\text{merging}}(t)$ was designed based on an estimation of t_{ramp} and in this case the merging vehicle reaches q_{ramp} slightly earlier than was estimated. This could easily be fixed by designing a new function $r_{\text{merging}}(t)$ at t_{ramp} and by adding $\dddot{r}_{\text{merging}}(t)$ as an input to the dynamics in (3.3.7), but this is not necessary since these discontinuities do not cause any problems and the dynamics in (3.3.7) do not contain $r_{\text{merging}}(t)$ at all. For the remaining vehicles, the standstill distance is constant and hence the derivatives equal zero.

Similarly, the plots of figure 5.3 show that the open-loop model is working as expected. Here, the merging vehicle arrives at q_{ramp} at time $t_{\text{ramp}} = 16.06$s, merges in front of vehicle $m = 3$ at time $t_{\text{merge}} = 16.96$s, and arrives at q_{end} at time $t_{\text{end}} = 26.45$s. Comparing these plots to those of the closed-loop model, we see that the merging vehicle acts very similar in the open-loop model, which is as expected since both models use the same dynamics for the merging vehicle until t_{ramp}. Similar to the closed-loop model, figure 5.3 shows discontinuities in the error and distance plots, but now at time t_{merge} instead of at time t_{ramp} since in the open-loop model the merging vehicle is not inserted into the platoon before it is able to merge. The merging vehicle is able to merge into the platoon closer to the beginning of the acceleration ramp q_{ramp}. With the closed-loop model, there was slightly more time between t_{ramp} and t_{merge}, but this was part of the model. By decreasing the value of t_{align} in the closed-loop model, the merging vehicle would merge into the platoon closer to q_{ramp}. A noticeable difference between the two models is also that the vehicles that will eventually be driving behind the merging vehicle in the platoon, so all vehicles $i \geq m = 3$, accelerate more aggressively in the open-loop model than in the closed-loop model. Figure 5.4 shows the acceleration of these vehicle and the merging vehicle for both models.

In designing the model, it was assumed that the platoon maintained a constant velocity. In reality, however, this is not guaranteed. The platoon could be accelerating or decelerating, and it is very likely that the platoon velocity fluctuates. Such disturbances in the platoon can be modeled by using a nonzero input signal $u_0(t)$ such that the virtual reference vehicle and therefore also the platoon will either accelerate or decelerate. We consider two cases: the virtual reference vehicle accelerates based on the input signal

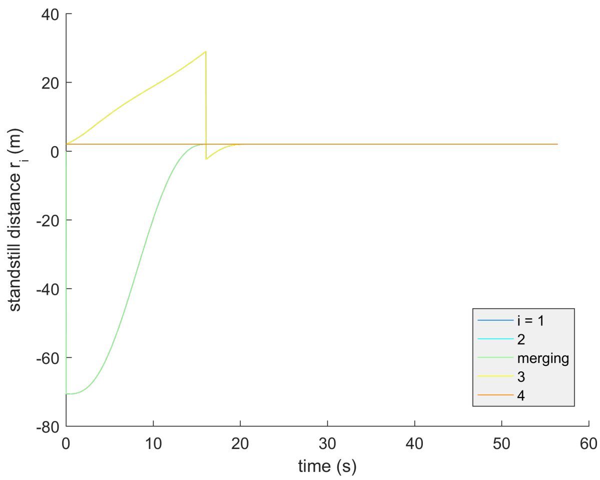
$$u_0(t) = \begin{cases} 1 & 5 \leq t \leq 10 \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

or it decelerates based on

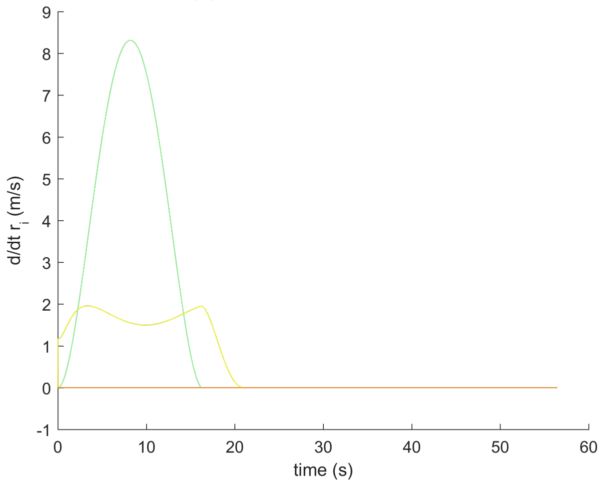
$$u_0(t) = \begin{cases} -1 & 5 \leq t \leq 10 \\ 0 & \text{otherwise} \end{cases}. \tag{5.2}$$

This means that during a period of five seconds the desired acceleration of the virtual reference vehicle equals plus or minus 1m/s^2 such that, in theory, its velocity increases to 38.33m/s or decreases to 28.33m/s. Some results of the closed-loop and the open-loop models are shown in figure 5.5 and figure 5.6. Apart from $u_0(t)$, no parameters were changed compared to the default settings (table 4.1).

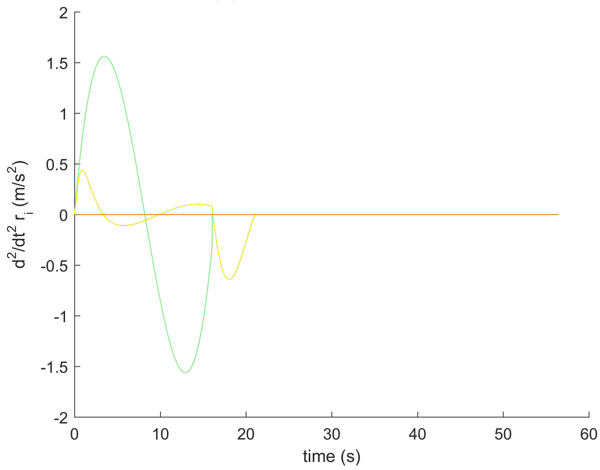
These results show that the merging vehicles of both the open-loop and closed-loop model are able to successfully merge into the platoon despite the fact that the platoon does not maintain a constant velocity. The errors in figures 5.5 and 5.6 converge to zero, all distances converge to the same value, all velocities converge to the same value, and the accelerations converge to zero. As expected, when using the input function of (5.1), the vehicles accelerate to a velocity of approximately 38.33m/s^2 (figure 5.5), while for the input of (5.2), the vehicles decelerate to a velocity of approximately 28.33m/s^2 (figure 5.6). And, since the desired distance between vehicles depends on their velocities, the distance between vehicles therefore increases in figure 5.5 and decreases in figure 5.6. Since the input $u_0(t)$ represents the desired acceleration of the virtual reference vehicle, and we use $\alpha = \beta = 0.8$ so that vehicles primarily react to the vehicle in front, the acceleration of the first vehicle in the platoon is expected to be able to follow the input function the best. Figures 5.5 and 5.6 indeed show that this is the case. As with the previous simulations with constant platoon velocity, the merging vehicles of the open-loop and closed-loop models again act very similar, and the error and distance plots show discontinuities at t_{merge} (open-loop) and at t_{ramp} (closed-loop).



(a) standstill distance

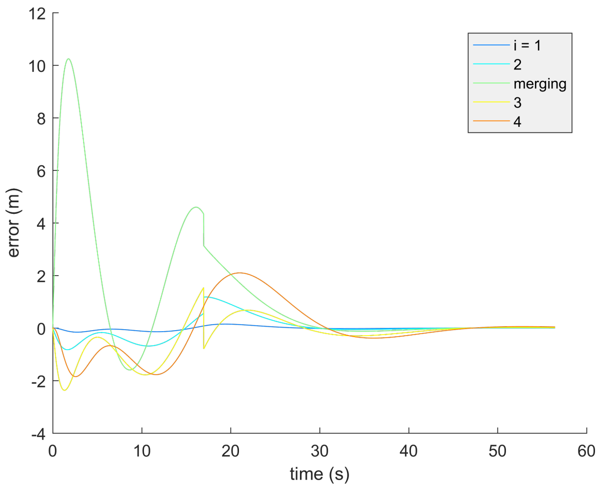


(b) first derivative

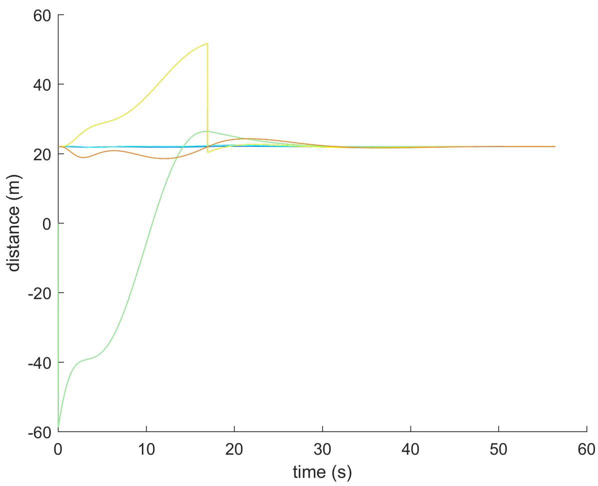


(c) second derivative

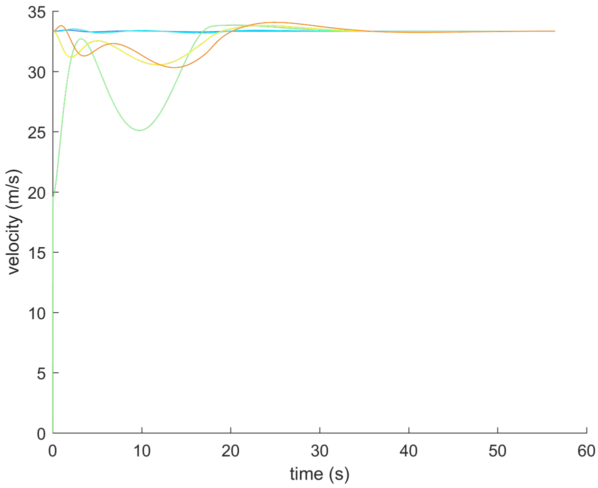
Figure 5.2: The standstill distance $r_i(t)$, the first derivative $\dot{r}_i(t)$, and the second derivative $\ddot{r}_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for a simulation of the closed-loop model with the default settings. In this case, $t_{\text{ramp}} = 16.06$s, $t_{\text{merge}} = 17.62$s, and $t_{\text{end}} = 26.45$s.



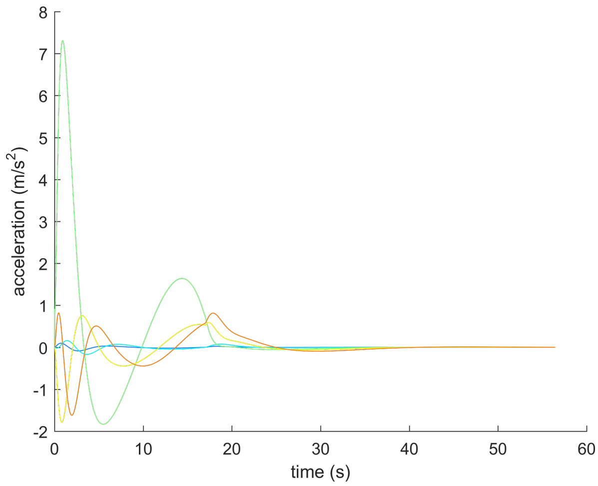
(a) error



(b) distance

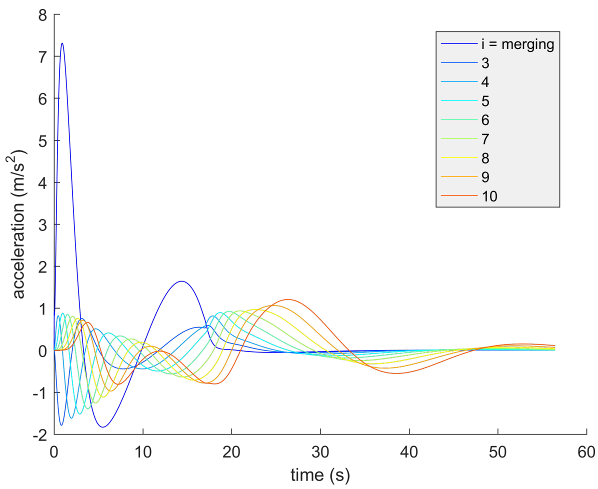


(c) velocity

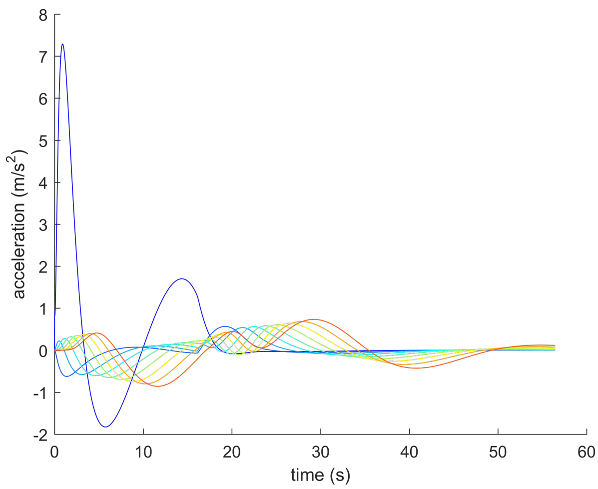


(d) acceleration

Figure 5.3: The error $e_i(t)$, distance $d_i(t)$, velocity $v_i(t)$, and acceleration $a_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for a simulation of the open-loop model with the default settings. In this case, $t_{\text{ramp}} = 16.06\text{s}$, $t_{\text{merge}} = 16.96\text{s}$, and $t_{\text{end}} = 26.45\text{s}$.



(a) open-loop



(b) closed-loop

Figure 5.4: The acceleration $a_i(t)$ of the merging vehicle and vehicles $(m =) 3, 4, \ldots, 10$ of the platoon for simulations of the open-loop and closed-loop models with the default settings.

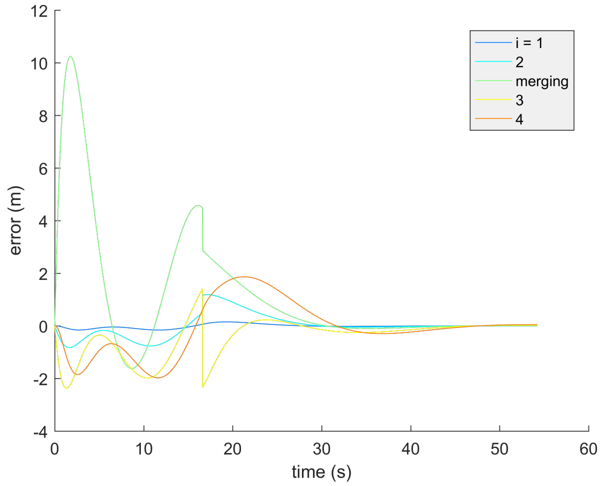
While the closed-loop model was designed to be able to handle such disturbances in the platoon velocity by taking into account the beginning and the end of the acceleration lane, this was not necessarily expected of the open-loop model. The model that we call the open-loop, however, is not fully open-loop but instead includes some feedback elements: the virtual reference vehicle of the merging vehicle is defined to be a copy of vehicle $m-1$, the distance between vehicles m and $m-1$ depends on the position of the merging vehicle relative to vehicle $m-1$, and when checking whether it is safe to merge the distance to the end of the acceleration lane is taken into account. A fully open-loop model would have more trouble dealing with these disturbances. This is demonstrated in figure 5.7. For this simulation, a simple model was implemented, where (3.2.10) was used for the platoon and r_m was defined as a 5th-degree polynomial with constraints (3.2.11) to (3.2.14) in order to increase the distance between vehicles m and $m-1$ from the default value r to $r+L_{\text{gap}}$. The merging vehicle was modeled by (3.2.17) with $r_{\text{merging}}(t) = r$ and $u_{0,\,\text{merging}}(t)$ given by (3.2.15). Figure 5.7 shows that when the platoon continues with constant velocity the merging vehicle accelerates to approximately the same velocity and drives close to vehicles m and $m-1$ at the estimated time of merge $\tilde{t}_{\text{ramp}}$. When the platoon accelerates, however, the velocity of the merging vehicle is too low compared to the platoon and hence it drives too far behind vehicle $m-1$ at time $\tilde{t}_{\text{ramp}}$. Conversely, when the platoon decelerates, the velocity of the merging vehicle is too high and at time $\tilde{t}_{\text{ramp}}$ it drives too close to vehicle $m-1$. Figure 5.7 was included mainly to demonstrate the problems caused by an open-loop model. Even though the open-loop model as it was designed in section 3.2 contains some feedback elements, compared to the closed-loop model of section 3.3 it is still an open-loop model. And while it does not clearly show in the plots of figures 5.5 and 5.6, for modeling the merge of a vehicle into a platoon on the highway using the closed-loop model seems to be the best choice out of the options that were considered and hence this model will be used in all remaining simulations. Moreover, figure 5.4 showed that the vehicles in the platoon accelerate more aggressively in the open-loop model than the vehicles in the closed-model. In practice, some vehicles might not be able to accelerate in this way plus a more gentle acceleration will most likely be preferred by the users.

Looking at the values of t_{ramp}, t_{merge}, and t_{end} for all previous simulations in table 5.1, we see that for both the open-loop and the closed-loop model the amount of time between t_{ramp} and t_{merge} increases when the platoon accelerates and decreases when the platoon decelerates. This can be explained by the fact that when the platoon accelerates it will reach q_{ramp} earlier than expected and hence has less time to form the gap, while it has more time when the platoon decelerates and reaches q_{ramp} later than expected. Also note that the value of L_{gap} was fixed based on the initial platoon velocity. When the platoon decelerates, distances between vehicles get smaller and hence also the distance needed to merge decreases, but since L_{gap} does not change, vehicles m and $m-1$ increase the distance more than is necessary and it becomes easier for the merging vehicle to merge. On the other hand, when the platoon accelerates, distances between the vehicles will get larger, which then means that there is more distance needed before the merging vehicle is able to merge. The value of L_{gap} remains the same and hence it will be more difficult for the merging vehicle to merge and the merge will take place further on the acceleration lane.

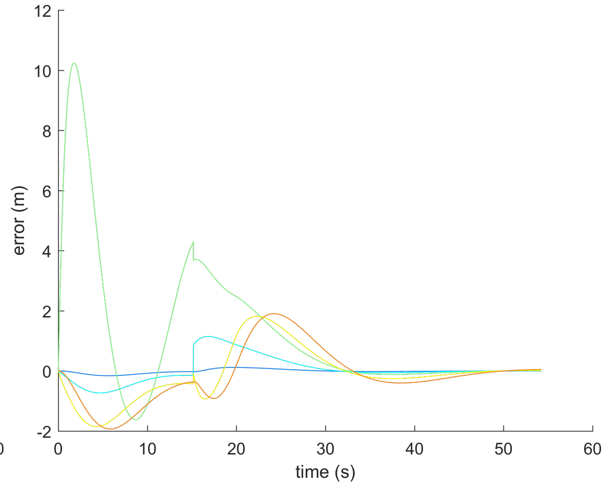
Table 5.1: Values of t_{ramp}, t_{merge}, and t_{end} for several simulations.

platoon velocity	model	t_{ramp} (s)	t_{merge} (s)	t_{end} (s)
constant	open-loop	16.06	16.96	26.45
	closed-loop	16.06	17.62	26.45
increasing	open-loop	15.16	16.61	24.24
	closed-loop	15.17	17.16	24.25
decreasing	open-loop	17.25	17.26	29.45
	closed-loop	17.27	18.29	29.45

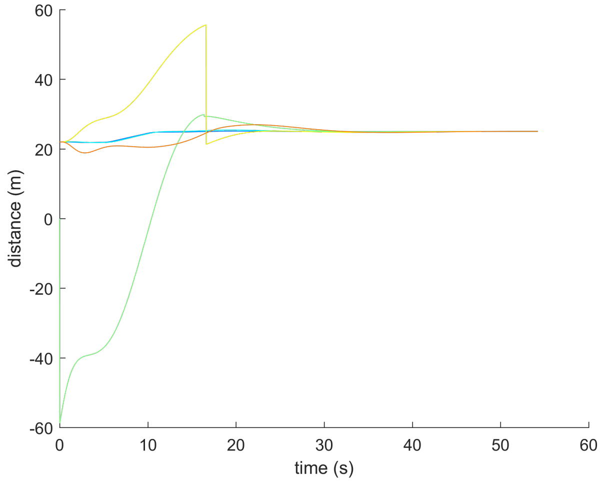
As can be seen in the previous plots, especially in figure 5.4, there is a slight delay between two consecutive vehicles. At time $t_{\text{start}} = 0$, vehicle $m = 3$ decelerates in order to increase the distance to vehicle 2. Vehicle 4 reacts quickly but it takes about a second before it starts decelerating. The response of vehicle 5 is also delayed compared to its predecessor, and this continues for all vehicles further towards the back of the platoon. In addition, the last vehicle of the platoon, vehicle 10, needs to



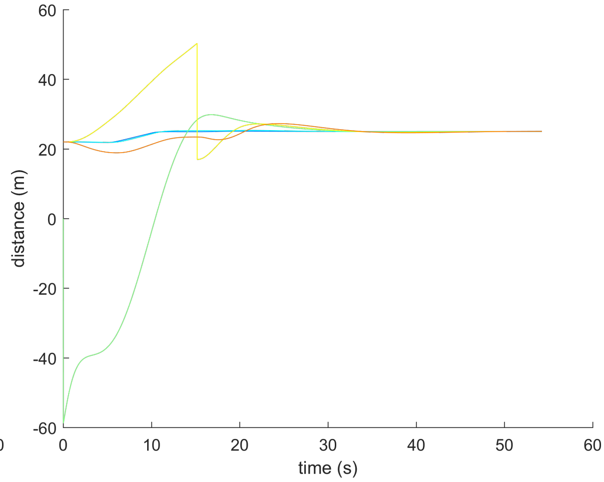
(a) error, open-loop



(b) error, closed-loop

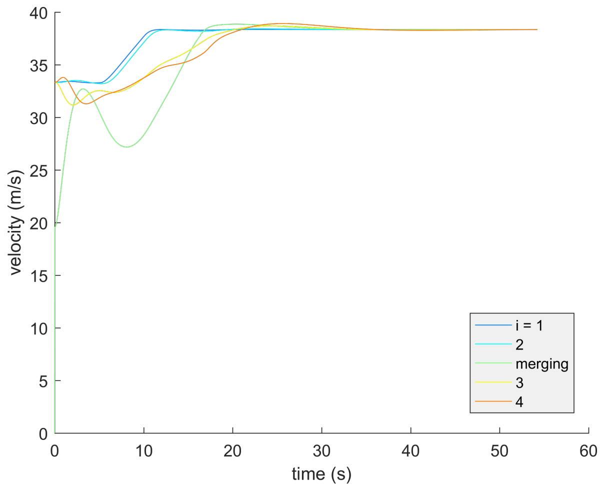


(c) distance, open-loop

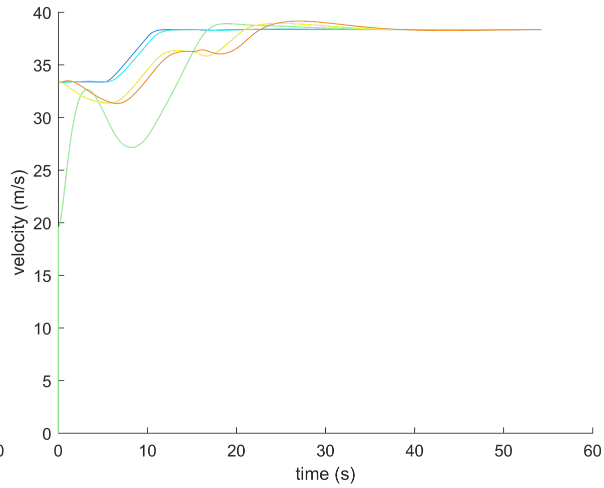


(d) distance, closed-loop

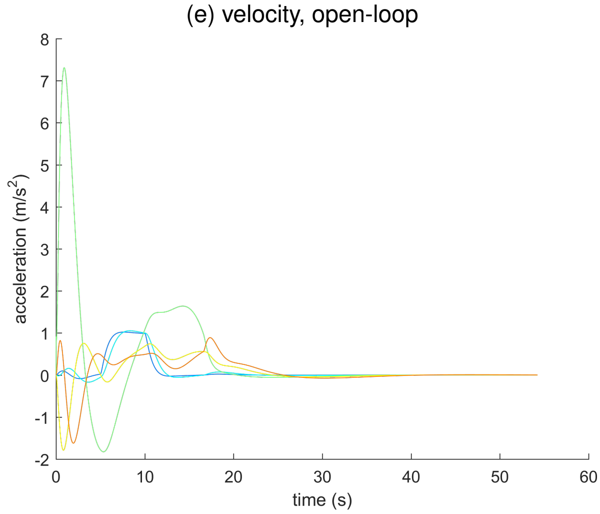
Figure 5.5: Results of simulations with an accelerating platoon. Plotted are the error $e_i(t)$, distance $d_i(t)$, velocity $v_i(t)$, and acceleration $a_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for a simulation of the open-loop model and a simulation of the closed-loop model. For both models the default settings were used except for the desired acceleration of the virtual reference vehicle $u_0(t)$, which is given by (5.1). In this case, for the open-loop model $t_{\text{ramp}} = 15.16$s, $t_{\text{merge}} = 16.61$s, and $t_{\text{end}} = 24.24$s, while for the closed-loop model $t_{\text{ramp}} = 15.17$s, $t_{\text{merge}} = 17.16$s, and $t_{\text{end}} = 24.25$s.



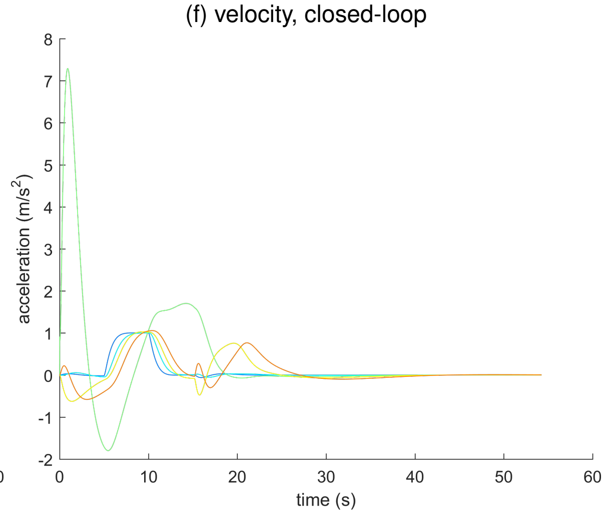
(e) velocity, open-loop



(f) velocity, closed-loop

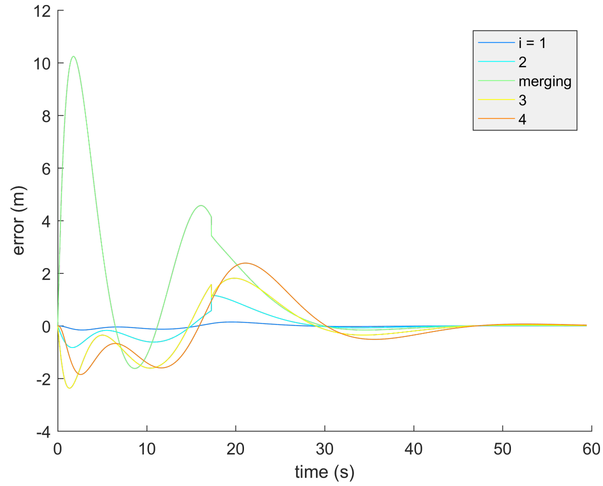


(g) acceleration, open-loop

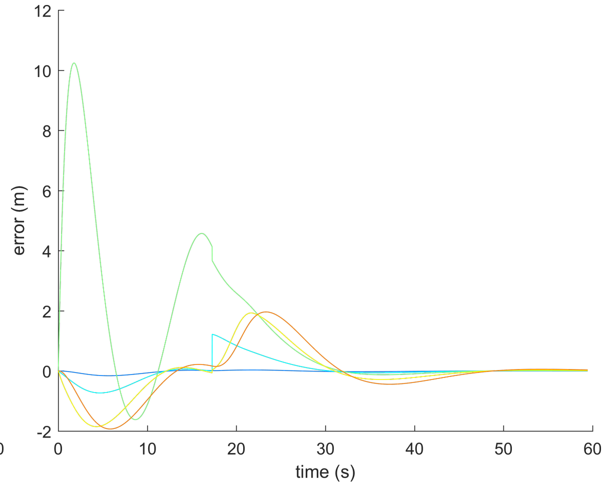


(h) acceleration, closed-loop

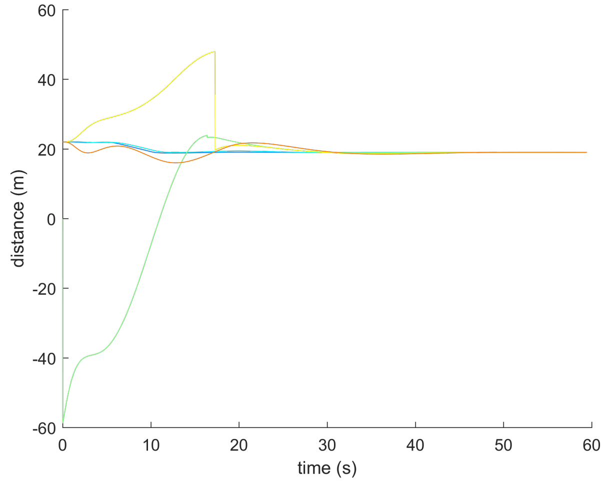
Figure 5.5 (continued): Results of simulations with an accelerating platoon. Plotted are the error $e_i(t)$, distance $d_i(t)$, velocity $v_i(t)$, and acceleration $a_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for a simulation of the open-loop model and a simulation of the closed-loop model. For both models the default settings were used except for the desired acceleration of the virtual reference vehicle $u_0(t)$, which is given by (5.1). In this case, for the open-loop model $t_{\mathrm{ramp}} = 15.16$s, $t_{\mathrm{merge}} = 16.61$s, and $t_{\mathrm{end}} = 24.24$s, while for the closed-loop model $t_{\mathrm{ramp}} = 15.17$s, $t_{\mathrm{merge}} = 17.16$s, and $t_{\mathrm{end}} = 24.25$s.



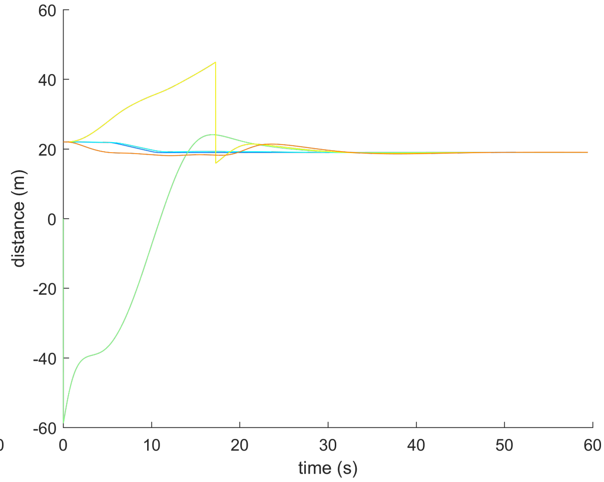
(a) error, open-loop



(b) error, closed-loop

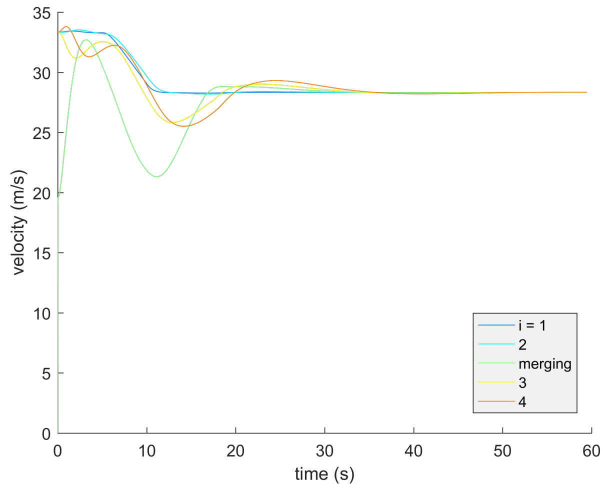


(c) distance, open-loop

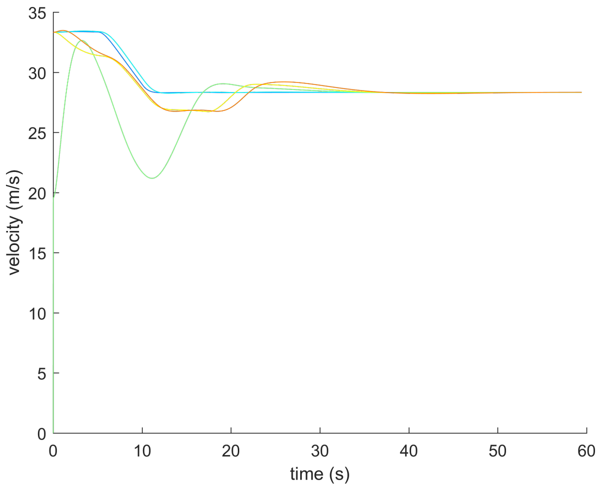


(d) distance, closed-loop

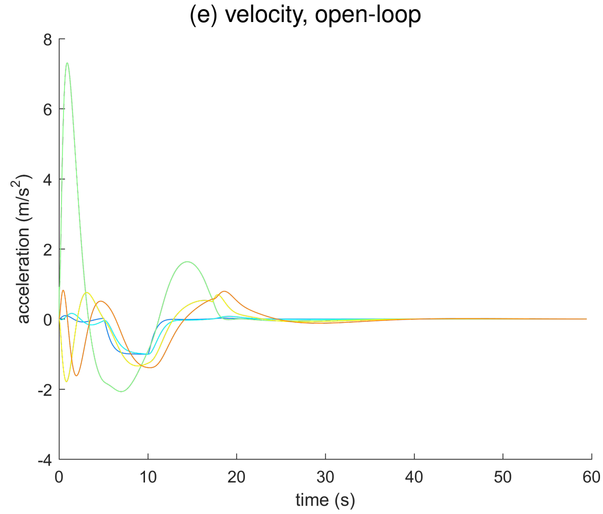
Figure 5.6: Results of simulations with a decelerating platoon. Plotted are the error $e_i(t)$, distance $d_i(t)$, velocity $v_i(t)$, and acceleration $a_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for a simulation of the open-loop model and a simulation of the closed-loop model. For both models the default settings were used except for the desired acceleration of the virtual reference vehicle $u_0(t)$, which is given by (5.2). In this case, for the open-loop model $t_{\text{ramp}} = 17.25$s, $t_{\text{merge}} = 17.26$s, and $t_{\text{end}} = 29.45$s, while for the closed-loop model $t_{\text{ramp}} = 17.27$s, $t_{\text{merge}} = 18.29$s, and $t_{\text{end}} = 29.45$s.



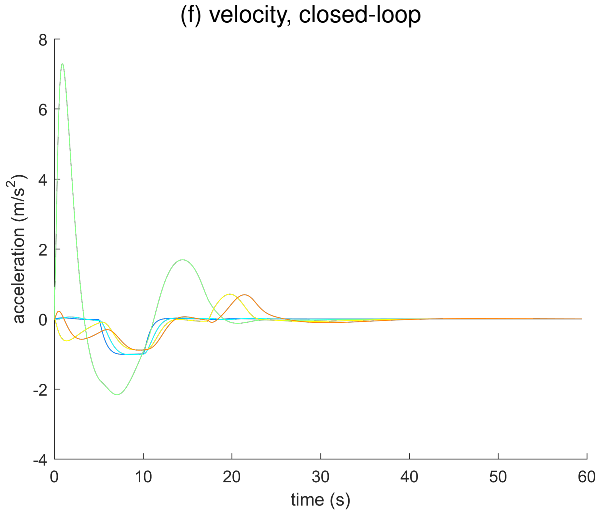
(e) velocity, open-loop



(f) velocity, closed-loop

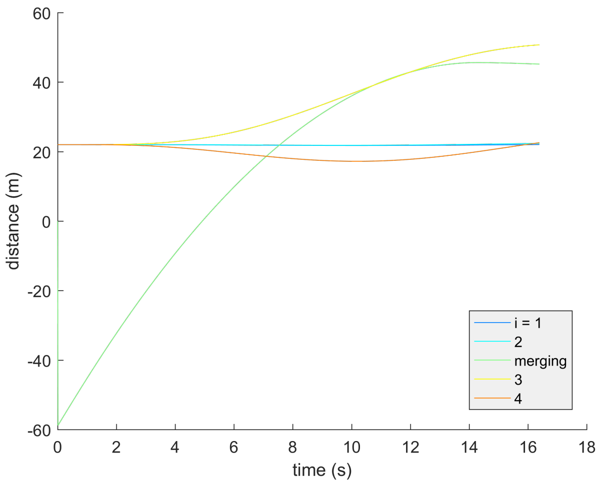


(g) acceleration, open-loop

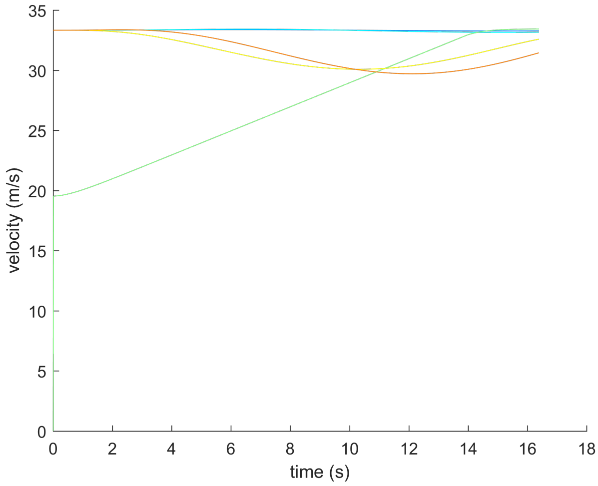


(h) acceleration, closed-loop

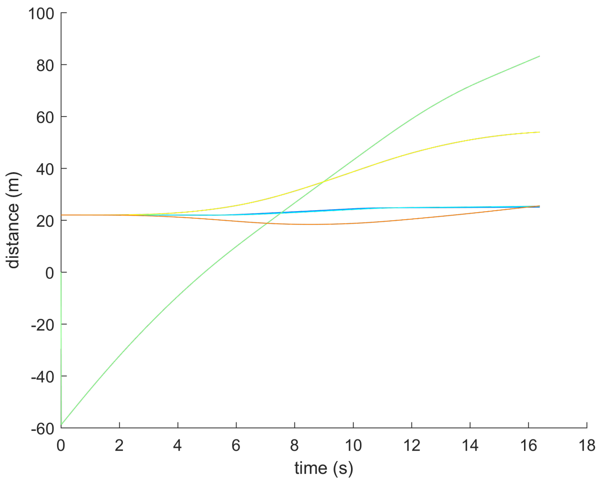
Figure 5.6 (continued): Results of simulations with a decelerating platoon. Plotted are the error $e_i(t)$, distance $d_i(t)$, velocity $v_i(t)$, and acceleration $a_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for a simulation of the open-loop model and a simulation of the closed-loop model. For both models the default settings were used except for the desired acceleration of the virtual reference vehicle $u_0(t)$, which is given by (5.2). In this case, for the open-loop model $t_{\text{ramp}} = 17.25$s, $t_{\text{merge}} = 17.26$s, and $t_{\text{end}} = 29.45$s, while for the closed-loop model $t_{\text{ramp}} = 17.27$s, $t_{\text{merge}} = 18.29$s, and $t_{\text{end}} = 29.45$s.



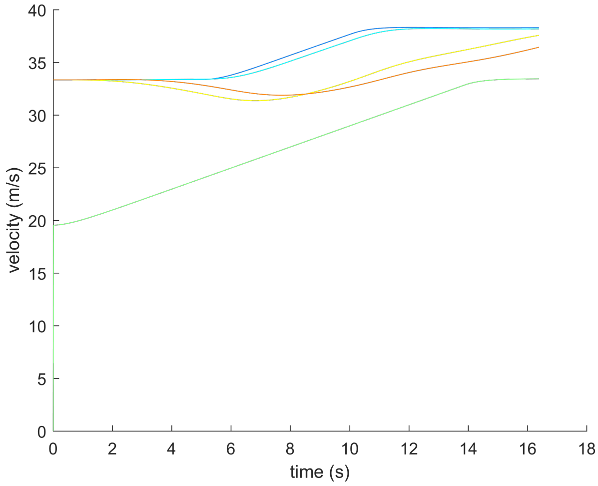
(a) constant platoon velocity, distance



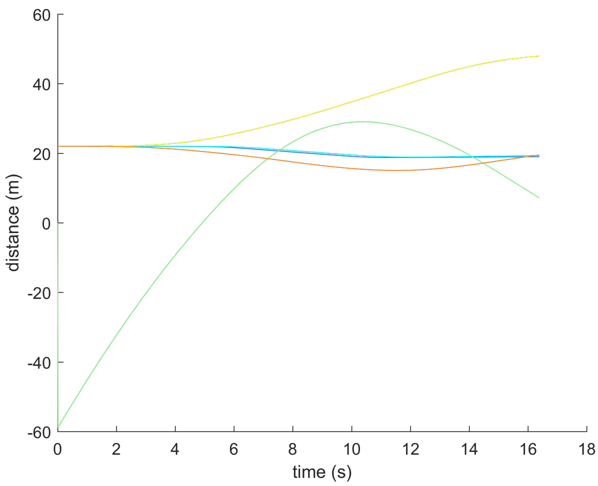
(b) constant platoon velocity, velocity



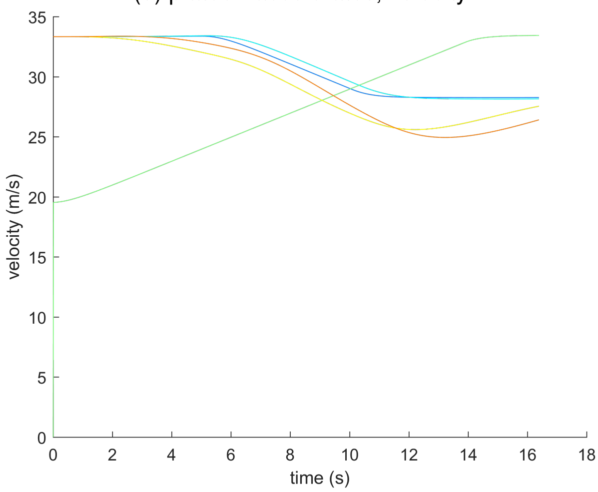
(c) platoon accelerates, distance



(d) platoon accelerates, velocity



(e) platoon decelerates, distance



(f) platoon decelerates, velocity

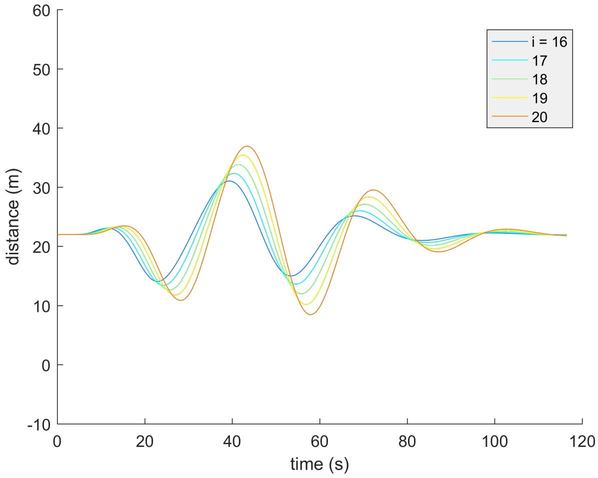
Figure 5.7: Results of simulations of a model that is fully open-loop. Simulations are performed for a platoon with constant velocity, an accelerating platoon, and a decelerating platoon. Plotted are the distance $d_i(t)$, and velocity $v_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon until time $\tilde{t}_{\text{ramp}} = 16.39\text{s}$. For each simulation the default settings were used except for the desired acceleration of the virtual reference vehicle $u_0(t)$, which is given by $u_0(t) = 0$, by (5.1), or by (5.2).

accelerate and decelerate more than vehicles that are closer to the front of the platoon. In this case, because the platoon only consists of 10 vehicles, the total delay for the last vehicle of the platoon is still relatively small, its acceleration is small compared to the merging vehicle, and hence the platoon does not have any problems in reaching steady state. For larger platoons, however, the vehicles in the back of the platoon might react too late or require accelerations that are too extreme. Consider for example a platoon consisting of 20 vehicles and a platoon consisting of 25 vehicles, which are both modeled by the closed-loop model. Figure 5.8 shows the distance, velocity, and acceleration of the last five vehicles of each platoon. For the platoon of 20 vehicles, the distances, velocities, and accelerations of the last five vehicles oscillate, but remain close enough to the steady-state values in order to eventually converge. When the platoon consists of 25 vehicles, however, the last vehicles need such a high acceleration that the distance between some vehicles becomes negative, meaning that in reality these vehicles would have crashed. For the closed-loop models with the default settings of table 4.1 platoons have a maximum size of 23 vehicles, though in practice smaller platoons would be used since with 23 vehicles the distances between vehicles can become very small or the required accelerations might be too large.

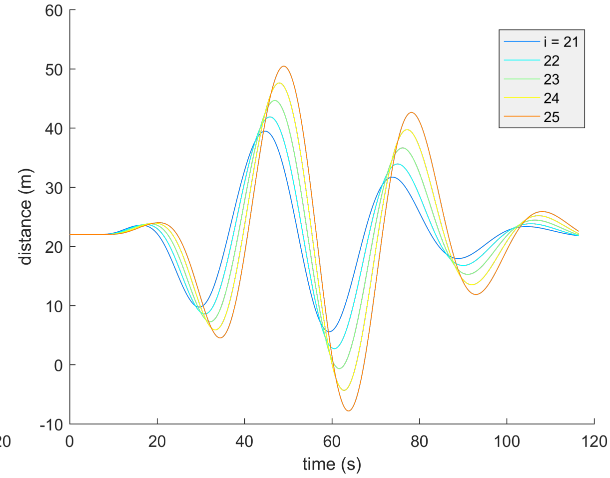
In section 2.9, it was proven that the model used for a steady-state platoon is asymptotically stable for $\alpha > 0$, $\beta > \frac{1}{2}$, and

$$\begin{cases} k_1 > 0 \\ k_2 > \frac{k_1 \tau}{\min_i\{\lambda_i k_3 + 1\}} \\ k_3 > -\frac{1}{\max_i\{\lambda_i\}} \end{cases}$$

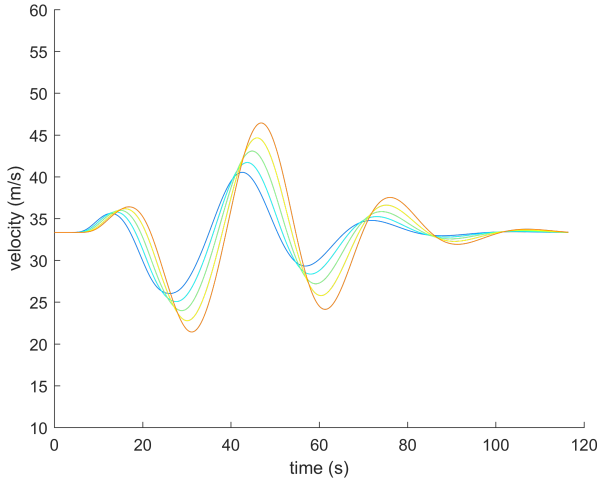
Changing these values leads to some interesting results of which some are plotted in figures 5.9 to 5.11. If, for example, k_1 is set to zero, as is done in figures 5.9a and 5.9b, the distances still converge but not to the same value, while the velocities do converge to the platoon velocity. Since k_1 is associated to the errors $e_i(t)$ and hence to the distances $d_i(t)$, while k_2 applies to $\dot{e}_i(t)$ and hence to the velocities $v_i(t)$, only the velocities are controlled and not the distances between vehicles. Figures 5.9a and 5.9b show exactly this. In practice, it might be useful to set $k_1 = 0$ for the first vehicle of the platoon, since this vehicle follows a virtual reference vehicle and hence the distance does not need to equal the desired distance. Using a different value for k_1, k_2, or k_3 that does not satisfy the corresponding constraint results in an unstable system. Figures 5.9c and 5.9d show the distances and velocities when $k_2 = 0$. Increasing the values of k_1 or k_2 mostly results in a larger acceleration for the merging vehicle as is shown in figure 5.10. For $\alpha = \beta = 1$, it is expected that only vehicle m and the vehicles that drive behind vehicle m decelerate in order to create space for the merging vehicle and that vehicles that drive in front of vehicle m do not react at all. This is exactly what happens for the closed-loop model with $\alpha = \beta = 1$ in figure 5.11.



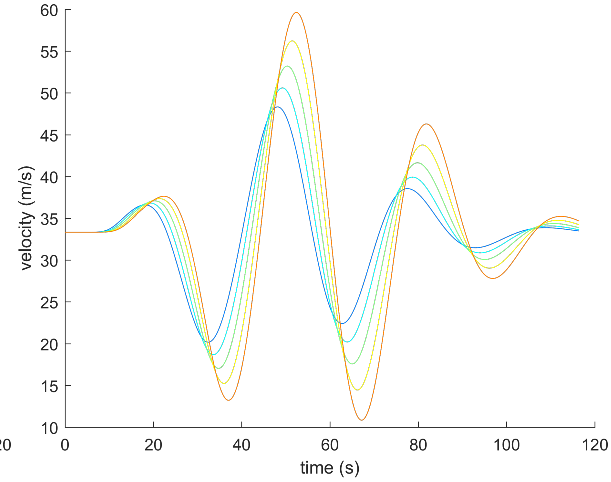
(a) distance, 20 vehicles



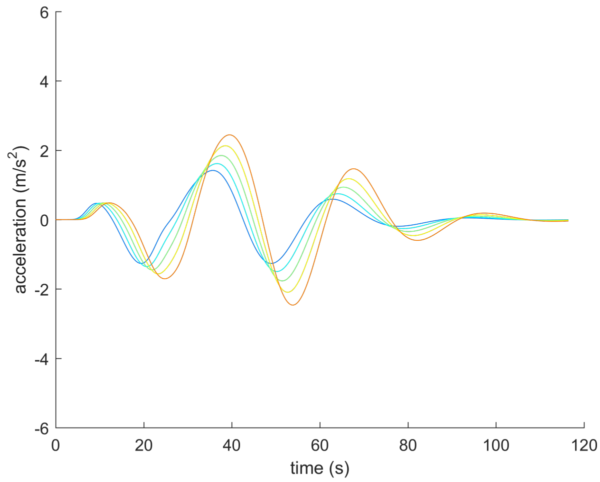
(b) distance, 25 vehicles



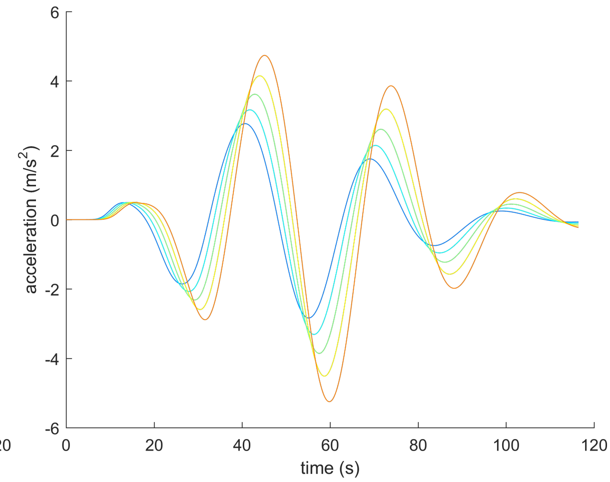
(c) velocity, 20 vehicles



(d) velocity, 25 vehicles

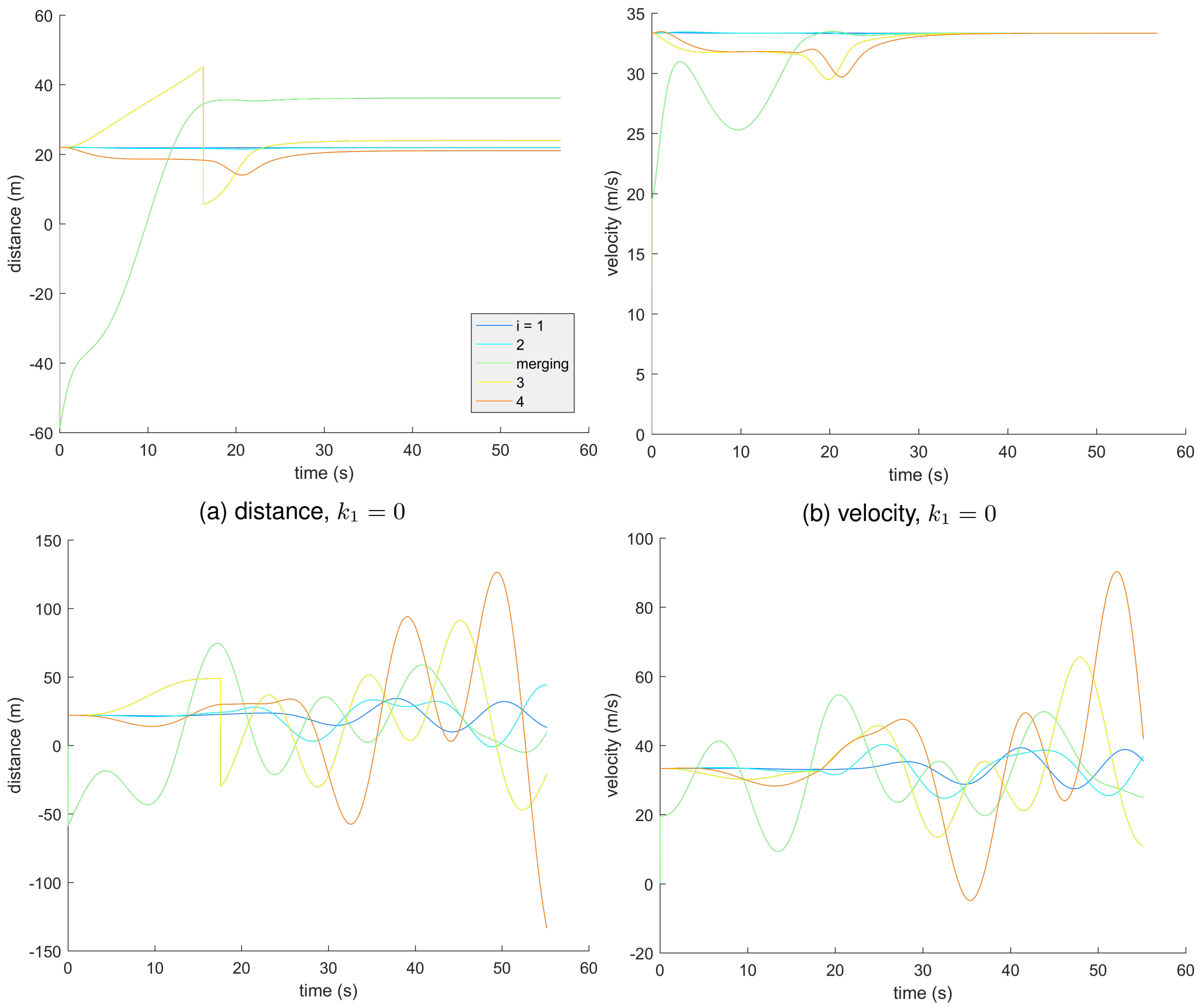


(e) acceleration, 20 vehicles



(f) acceleration, 25 vehicles

Figure 5.8: Results of simulations of platoons consisting of 20 and 25 vehicles. The plots show the distance $d_i(t)$, velocity $v_i(t)$, and acceleration $a_i(t)$ of the last 5 vehicles of the platoons for simulations of the closed-loop models with the default settings.



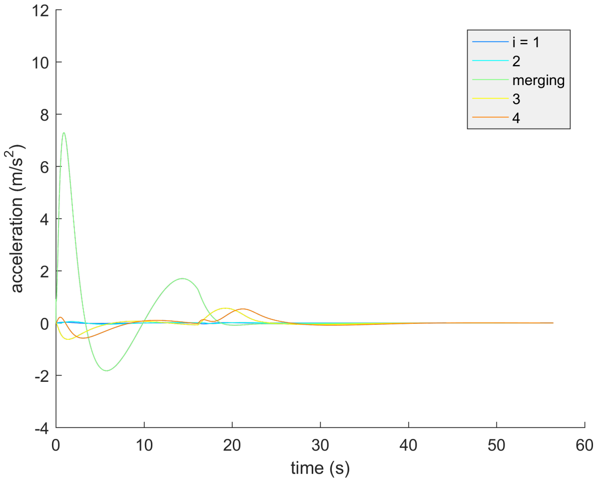
(a) distance, $k_1 = 0$

(b) velocity, $k_1 = 0$

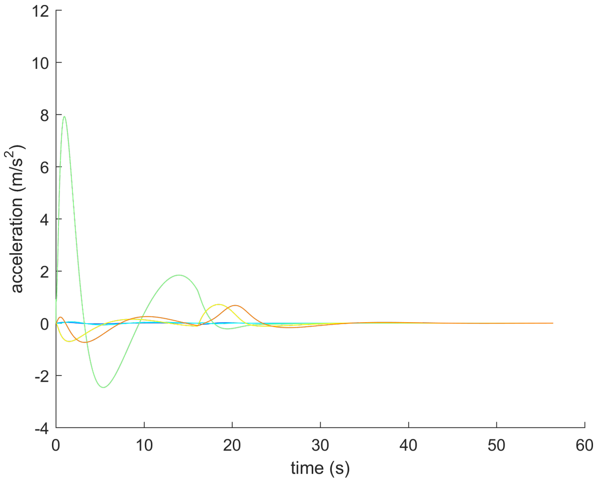
(c) distance, $k_2 = 0$

(d) velocity, $k_2 = 0$

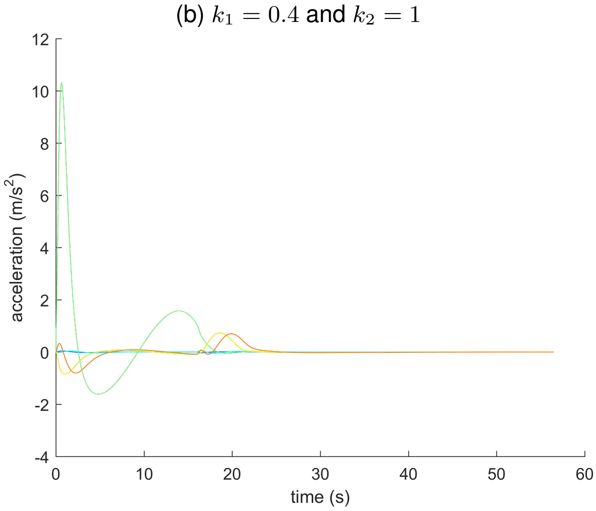
Figure 5.9: Results of simulations where either $k_1 = 0$ or $k_2 = 0$. The plots show the distance $d_i(t)$ and velocity $v_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for simulations of the closed-loop model with the default settings.



(a) default: $k_1 = 0.2$ and $k_2 = 1$

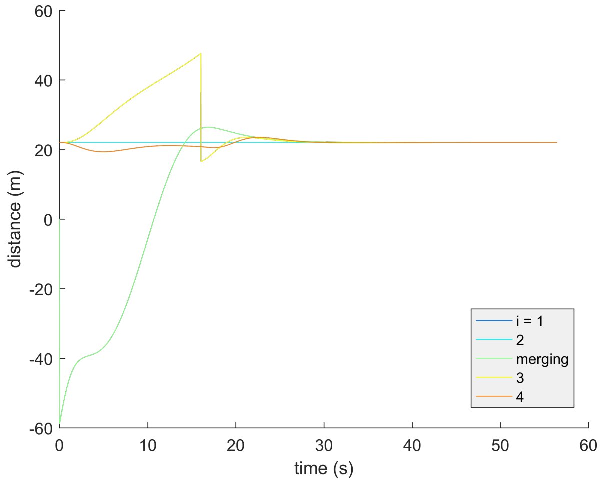


(b) $k_1 = 0.4$ and $k_2 = 1$

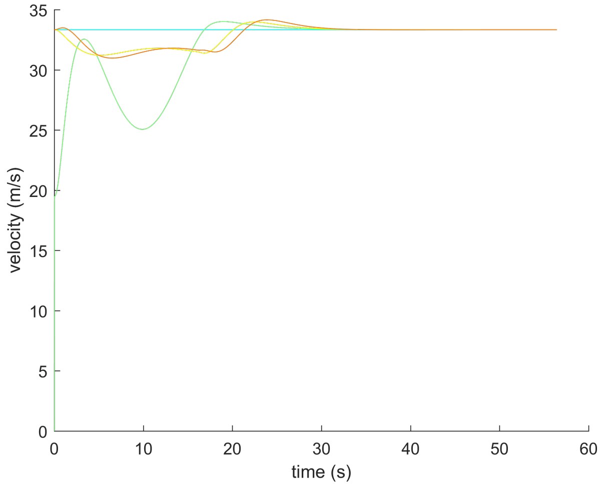


(c) $k_1 = 0.2$ and $k_2 = 2$

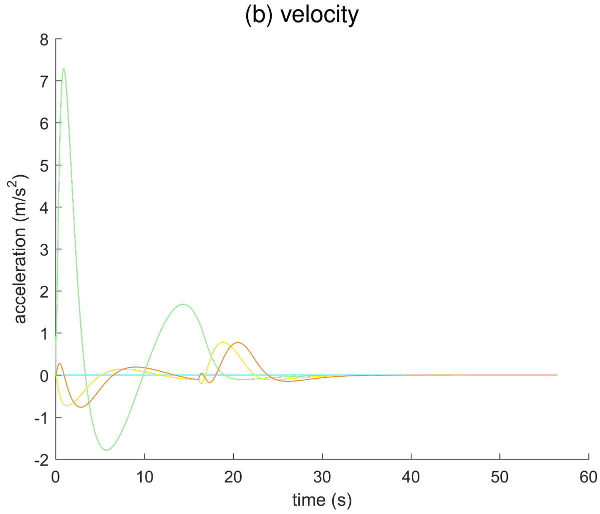
Figure 5.10: Results of simulations where either k_1 or k_2 is increased. The plots show the acceleration $a_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for simulations of the closed-loop model with the default settings.



(a) distance



(b) velocity



(c) acceleration

Figure 5.11: Results of a simulation where $\alpha = \beta = 1$. The plots show distance $d_i(t)$, velocity $v_i(t)$, and acceleration $a_i(t)$ of the merging vehicle and vehicles 1, 2, $m = 3$, and 4 of the platoon for a simulation of the closed-loop model with the default settings.

6. Conclusions

Overall, the results in chapter 5 show that with the methods proposed in chapter 3 we are able to model a platoon of automated vehicles and to control the vehicles such that a separate automated vehicle entering the highway merges into the platoon that is already driving on the highway.

Nevertheless, the models could be further improved in future research. The results showed that the difference between the open-loop model from section 3.2 and the closed-loop model from section 3.3 is small. The closed-loop model was expected to have better performance than the open-loop model since it takes into account the beginning and end of the acceleration lane. The closed-loop model, however, still lets the merging vehicle position itself behind vehicle $m-1$ of the platoon based on the estimated time of merge. Instead, this could also be done based on the distance between the platoon and the beginning of the acceleration lane.

Furthermore, the model might be improved by designing a controller for virtual reference vehicle of the platoon. In that case, the platoon would be able to accelerate or decelerate slightly in order to let a vehicle merge into the platoon. It would also be possible to control only the velocity of the first car of the platoon, since it is not necessary that this vehicle drives at the desired distance behind the virtual reference vehicle. A different strategy for selecting the vehicle m could improve the model. If, for example, the merging vehicle is not able to accelerate fast enough, it might be desirable to let vehicle m accelerate such that it closes the gap to vehicle $m-1$ and the merging vehicle can instead merge between vehicles $m+1$ and m.

For a more realistic model, the methods used could be applied to a heterogeneous platoon that takes into account the differences between vehicles. The velocities and accelerations of vehicles could be restricted. Right now, because the vehicles can achieve any required acceleration, the merge can almost always be completed successfully. The models could be extended such that multiple vehicles are able to merge into the platoon. Some delays or noise could be added to the used signals, for example, by using a nonzero value for the drive-line dynamics time delay ϕ.

Bibliography

[1] Z. Abdollahi, R. A. Biroon, and P. Pisu. Cooperative adaptive cruise control: PDE modeling and stability analysis. In *2018 Annual American Control Conference (ACC)*, pages 5554–5559. IEEE, 2018.

[2] S. Gong and L. Du. Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transportation Research Part B: Methodological*, 116:25–61, 2018.

[3] S. Gong, J. Shen, and L. Du. Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon. *Transportation Research Part B: Methodological*, 94:314–334, 2016.

[4] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 1990.

[5] E. S. Kazerooni and J. Ploeg. Interaction protocols for cooperative merging and lane reduction scenarios. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 1964–1970. IEEE, 2015.

[6] G. J. Naus, R. P. Vugts, J. Ploeg, M. J. van de Molengraft, and M. Steinbuch. String-stable CACC design and experimental validation: A frequency-domain approach. *IEEE Transactions on vehicular technology*, 59(9):4268–4279, 2010.

[7] C. Nowakowski, S. E. Shladover, X.-Y. Lu, D. Thompson, and A. Kailas. Cooperative adaptive cruise control (CACC) for truck platooning: Operational concept alternatives. 2015. URL `https://escholarship.org/uc/item/7jf9n5wm`.

[8] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 260–265. IEEE, 2011.

[9] J. Ploeg, D. P. Shukla, N. van de Wouw, and H. Nijmeijer. Controller synthesis for string stability of vehicle platoons. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):854–865, 2014.

[10] J. Ploeg, N. Van De Wouw, and H. Nijmeijer. Lp string stability of cascaded systems: Application to vehicle platooning. *IEEE Transactions on Control Systems Technology*, 22(2):786–793, 2014.

[11] R. Rajamani and C. Zhu. Semi-autonomous adaptive cruise control systems. *IEEE Transactions on Vehicular Technology*, 51(5):1186–1192, 2002.

[12] W. Ren and R. W. Beard. *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

[13] W. Ren, K. Moore, and Y. Chen. High-order consensus algorithms in cooperative vehicle systems. In *Networking, Sensing and Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on*, pages 457–462. IEEE, 2006.

[14] Rijkswaterstaat. *Richtlijn Ontwerp Autosnelwegen 2017: ROA2017*. 2017. URL `http://publicaties.minienm.nl/documenten/roa-2017-rws-gpo`.

[15] S. Sheikholeslam and C. A. Desoer. Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: A system level study. *IEEE Transactions on vehicular technology*, 42(4):546–554, 1993.

[16] S. S. Stankovic, M. J. Stanojevic, and D. D. Siljak. Decentralized overlapping control of a platoon of vehicles. *IEEE Transactions on Control Systems Technology*, 8(5):816–832, 2000.

[17] X. Sun, R. Horowitz, and C.-W. Tan. An efficient lane change maneuver for platoons of vehicles in an automated highway system. In *ASME 2003 International Mechanical Engineering Congress and Exposition, Dynamic Systems and Control, Volumes 1 and 2*, pages 355–362. American Society of Mechanical Engineers, 2003.

[18] J. W. Wang, Y. Yu, and C. Y. Sun. A PDE-based approach to formation control design for a large vehicular platoon. In *Chinese Automation Congress (CAC), 2015*, pages 1129–1134. IEEE, 2015.

[19] Z. Wang, G. Wu, and M. J. Barth. Developing a distributed consensus-based cooperative adaptive cruise control system for heterogeneous vehicles with predecessor following topology. *Journal of Advanced Transportation*, 2017, Article ID 1023654, 2017.

[20] J. C. Willems and J. W. Polderman. *Introduction to mathematical systems theory: a behavioral approach*, volume 26. Springer Science & Business Media, 2013.

[21] J. C. Zegers. Distributed consensus control in vehicle platooning. Master's thesis, Dept. Mech. Eng. Dyn. Control, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2015.

[22] J. C. Zegers, E. Semsar-Kazerooni, J. Ploeg, N. van de Wouw, and H. Nijmeijer. Consensus-based bi-directional CACC for vehicular platooning. In *American Control Conference (ACC), 2016*, pages 2578–2584. IEEE, 2016.

[23] J. C. Zegers, E. Semsar-Kazerooni, J. Ploeg, N. van de Wouw, and H. Nijmeijer. Consensus control for vehicular platooning with velocity constraints. *IEEE Transactions on Control Systems Technology*, 26(5):1592–1605, 2018.

[24] Y. Zheng, S. E. Li, J. Wang, L. Y. Wang, and K. Li. Influence of information flow topology on closed-loop stability of vehicle platoon with rigid formation. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2094–2100. IEEE, 2014.

Appendices

A. Proof of closed-loop stability

In this appendix, the stability of the closed-loop system in (2.8.7) with $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$ defined by (2.1.10) and (2.1.11) is proven as was mentioned in section 2.9. The main results are given in theorems A.5 to A.7 and come together in corollary A.8, for which we will need the following definitions and lemma's.

Definition A.1 *The matrix $M = [m_{ij}] \in \mathbb{R}^{n\times n}$ is* diagonally dominant *if*

$$|m_{ii}| \geq \sum_{j\neq i} |m_{ij}| \quad \forall i = 1,\, \ldots, n.$$

Lemma A.2 [4, page 174] *Let $M = [m_{ij}] \in \mathbb{R}^{n\times n}$ be a tridiagonal matrix. If $m_{i,i+1}m_{i+1,i} \geq 0$ for all $i = 1,\, \ldots, n-1$, then all eigenvalues of M are real.*

Lemma A.3 (Gershgorin disc theorem) [4, theorem 6.1.1]. *Let $M = [m_{ij}] \in \mathbb{R}^{n\times n}$. The eigenvalues of M are located in the union of the n* Gershgorin discs

$$\bigcup_{i=1}^{n} \left\{ \lambda \in \mathbb{C} \;\middle|\; |\lambda - m_{ii}| \leq \sum_{j\neq i} |m_{ij}| \right\}.$$

Lemma A.4 (Lyapunov equation) [20, section 7.4] *Let $M \in \mathbb{R}^{n\times n}$. The system $\dot{X} = M\dot{X}$ is asymptotically stable if and only if there are positive definite matrices P and Q that satisfy the* Lyapunov equation

$$M^TP + PM = -Q.$$

Now we can continue with the main theorems. First of all, theorem A.5 provides bounds for the controller gain values $k^T := [k_1,\, k_2,\, k_3]$ such that the system $\dot{X} = (I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T)X$ is asymptotically stable given that $\hat{\mathcal{L}}$ has positive real eigenvalues. Theorem A.6 then shows that the matrix $\hat{\mathcal{L}}$ defined by (2.1.10) and (2.1.11) does indeed have positive real eigenvalues provided that $\alpha \in (0, 1]$. Next, it is proven in theorem A.7 that the matrix A_u is stable for $\beta \in \left(\frac{1}{2}, 1\right]$. And, finally, corollary A.8 combines these results into one statement that describes the stability of the closed-loop dynamics of (2.8.7).

Theorem A.5 [21, 24] *Consider the system $\dot{X} = (I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T)X$, where the communication topology is given by $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$. Let λ_i, $i = 1, \ldots, n$, be the eigenvalues of $\hat{\mathcal{L}}$. If all eigenvalues λ_i are positive and real, the system is asymptotically stable if and only if the values of $k^T := [k_1,\, k_2,\, k_3]$ satisfy*

$$\begin{cases} k_1 > 0 \\ k_2 > \frac{k_1\tau}{\min_i\{\lambda_i k_3 + 1\}} \\ k_3 > -\frac{1}{\max_i\{\lambda_i\}} \end{cases}$$

Theorem A.6 *The communication topology given by $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$, where the Laplacian matrix $\mathcal{L}$ is defined in (2.1.10) and the pinning matrix $\mathcal{P}$ in (2.1.11), has positive real eigenvalues for $\alpha \in (0, 1]$.*

Proof.

Using the Laplacian matrix $\mathcal{L}$ in (2.1.10) and the pinning matrix $\mathcal{P}$ in (2.1.11) results in

$$\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P} = \begin{pmatrix} 1 & -(1-\alpha) & 0 & \dots & 0 \\ -\alpha & 1 & -(1-\alpha) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -\alpha & 1 & -(1-\alpha) \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix}.$$

Consider the following cases.

- For $\alpha = 1$, matrix $\hat{\mathcal{L}}$ is lower triangular and hence the eigenvalues of $\hat{\mathcal{L}}$ are on the diagonal and are all equal to one.
- For $\alpha \in (0, 1)$, all non-zero off-diagonal elements are negative, which means that lemma A.2 applies and all eigenvalues of $\hat{\mathcal{L}}$ are real. Since $\hat{\mathcal{L}}$ is diagonally dominant, it follows from the Gershgorin disc theorem (lemma A.3) that the eigenvalues are nonnegative. To show that zero is not an eigenvalue of $\hat{\mathcal{L}}$, we try to find a nonzero vector $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ such that $\hat{\mathcal{L}}x = 0$. Let $\hat{\mathcal{L}}(i, :)$ denote the i-th row of matrix $\hat{\mathcal{L}}$. For the last row of $\hat{\mathcal{L}}$ we have that $\hat{\mathcal{L}}(n, :)x = -x_{n-1} + x_n$ and hence x_{n-1} should be equal to x_n. And for row $n-1$ we then have that $\hat{\mathcal{L}}(n-1, :)x = -\alpha x_{n-2} + x_{n-1} - (1-\alpha)x_n = -\alpha x_{n-2} + \alpha x_n$, which only equals zero when $x_{n-2} = x_n$. Similarly, from rows $n-2, \dots, 2$ it follows that $x_{n-3}, \dots, x_1$ should all be equal to x_n. The eigenvector x should thus be of the form $x = x_n \cdot [1, \dots, 1]^T$ for some $x_n \in \mathbb{R}$, but then $\hat{\mathcal{L}}x = [\alpha x_n, 0, \dots, 0]^T$ and this only equals zero when $x_n = 0$. Therefore, zero cannot be an eigenvalue of $\hat{\mathcal{L}}$, so all eigenvalues of $\hat{\mathcal{L}}$ are positive and real.

□

Theorem A.7 *All eigenvalues of the matrix*

$$A_u := -\frac{1}{h} \begin{pmatrix} 2\beta-1 & 1-\beta & 0 & \dots & \dots & 0 \\ -\beta & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1-\beta & 0 \\ \vdots & & \ddots & -\beta & 2\beta-1 & 1-\beta \\ 0 & \dots & \dots & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{n \times n}$$

are located in the open left half-plane for $\beta \in \left(\frac{1}{2}, 1\right]$.

Proof.

Let $P \in \mathbb{R}^{n \times n}$ be a positive matrix defined as

$$P := \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ 0 & \dots & 0 & \beta \end{pmatrix}$$

which yields

$$PA_u = -\frac{1}{h} \begin{pmatrix} 2\beta-1 & 1-\beta & 0 & \dots & \dots & 0 \\ -\beta & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1-\beta & 0 \\ \vdots & & \ddots & -\beta & 2\beta-1 & 1-\beta \\ 0 & \dots & \dots & 0 & -\beta & \beta \end{pmatrix}$$

and consider the matrix $Q = [q_{ij}] \in \mathbb{R}^{n\times n}$, which is given by

$$Q := -\left(A_u^T P + PA_u\right) = \frac{1}{h}\begin{pmatrix} 4\beta-2 & 1-2\beta & 0 & \dots & \dots & 0 \\ 1-2\beta & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1-2\beta & 0 \\ \vdots & & \ddots & 1-2\beta & 4\beta-2 & 1-2\beta \\ 0 & \dots & \dots & 0 & 1-2\beta & 2\beta \end{pmatrix}.$$

Clearly, this matrix Q has positive diagonal entries for $\beta > \frac{1}{2}$, and for the rows then holds that

$$\begin{aligned} |q_{11}| &= \frac{|4\beta-2|}{h} = \frac{4\beta-2}{h} = \frac{2(2\beta-1)}{h} > \frac{2\beta-1}{h} = \frac{|1-2\beta|}{h} = \sum_{j\neq 1}|q_{1j}| \\ |q_{ii}| &= \frac{|4\beta-2|}{h} = \frac{4\beta-2}{h} = \frac{2(2\beta-1)}{h} = \frac{2(2\beta-1)}{h} = \frac{2|1-2\beta|}{h} = \sum_{j\neq i}|q_{ij}| \quad \text{for } i = 2,\dots,n-1 \\ |q_{nn}| &= \frac{|2\beta|}{h} = \frac{2\beta}{h} > \frac{2\beta-1}{h} = \frac{|1-2\beta|}{h} = \sum_{j\neq n}|q_{nj}| \end{aligned}$$

and hence, by definition A.1, the matrix Q is diagonally dominant for $\beta > \frac{1}{2}$. Furthermore, since matrix Q is symmetric, all its eigenvalues are real. From lemma A.3, the Gershgorin disc theorem, it follows that for $\beta > \frac{1}{2}$ all eigenvalues of Q are nonnegative. To prove that all eigenvalues are positive, we will show that zero cannot be an eigenvalue of Q. Let $Q(i,:)$ denote the i-th row of matrix Q. Suppose that zero is an eigenvalue of matrix Q, then there is a nonzero vector $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ such that $Qx = 0$. Looking at the first row of Q, the product

$$Q(1,:)x = (4\beta-2)x_1 + (1-2\beta)x_2$$

equals zero only when $x_2 = 2x_1$. For the second row, since now $x_2 = 2x_1$, we have that

$$\begin{aligned} Q(2,:)x &= (1-2\beta)x_1 + (4\beta-2)x_2 + (1-2\beta)x_3 \\ &= -(2\beta-1)x_1 + 4(2\beta-1)x_1 - (2\beta-1)x_3 \\ &= 3(2\beta-1)x_1 - (2\beta-1)x_3, \end{aligned}$$

which equals zero only when $x_3 = 3x_1$. In general, for $i = 2, \dots, n-1$, $Q(i,:)x = 0$ holds only when $x_i = ix_1$, since

$$\begin{aligned} Q(i,:)x &= (1-2\beta)x_{i-1} + (4\beta-2)x_i + (1-2\beta)x_{i+1} \\ &= -(2\beta-1)(i-1)x_1 + 2(2\beta-1)(i)x_1 - (2\beta-1)(i+1)x_1 \\ &= -[(i-1) - 2i + (i+1)](2\beta-1)x_1 \\ &= 0. \end{aligned}$$

The vector x thus becomes $x = [x_1, 2x_2, 3x_3, \dots, nx_n]^T$. For the last row, however, we have

$$\begin{aligned} Q(n,:)x &= (1-2\beta)x_{n-1} + (2\beta)x_n \\ &= (1-2\beta)(n-1)x_1 + (2\beta)(n)x_1 \\ &= [n - 2\beta n + 2\beta - 1 + 2\beta n]x_1 \\ &= [n + 2\beta - 1]x_1, \end{aligned}$$

which, since $n > 0$ and $2\beta - 1 > 0$, only equals zero when $x_1 = 0$. The only solution of $Qx = 0$ is thus $x = 0$, which contradicts that x is nonzero. Therefore, matrix Q is nonsingular and zero is not an eigenvalue of Q. All eigenvalues of Q are positive and real.

Finally, since $A_u^T P + PA_u = -Q$ and both P and Q are positive matrices, it follows from lemma A.4 that the matrix A_u is stable. □

Corollary A.8 *The closed-loop dynamics of (2.8.7) with $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$ defined by the Laplacian matrix $\mathcal{L}$ in (2.1.10) and the pinning matrix $\mathcal{P}$ in (2.1.11) are asymptotically stable if $\alpha \in (0, 1]$, $\beta \in \left(\frac{1}{2}, 1\right]$, and the values of $k^T := [k_1,\, k_2,\, k_3]$ satisfy*

$$\begin{cases} k_1 > 0 \\ k_2 > \frac{k_1 \tau}{\min_i\{\lambda_i k_3 + 1\}} \\ k_3 > -\frac{1}{\max_i\{\lambda_i\}} \end{cases}$$

Proof.

Since the system matrix of the closed-loop dynamics of (2.8.7) is lower block triangular, we only need to consider the diagonal blocks. The complete system is asymptotically stable when the eigenvalues of the diagonal blocks, $(I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T)$ and A_u, are located in the left open half-plane. From theorem A.7 it follows that the matrix A_u is stable for $\beta \in \left(\frac{1}{2}, 1\right]$. Theorem A.6 shows that $\hat{\mathcal{L}}$ has positive real eigenvalues for $\alpha \in (0, 1]$. Therefore, theorem A.5 applies and the eigenvalues of $(I_n \otimes A - \hat{\mathcal{L}} \otimes Bk^T)$ have negative real parts for $\alpha \in (0, 1]$ and

$$\begin{cases} k_1 > 0 \\ k_2 > \frac{k_1 \tau}{\min_i\{\lambda_i k_3 + 1\}} \\ k_3 > -\frac{1}{\max_i\{\lambda_i\}} \end{cases}$$

□

Remark A.9 *Note that corollary A.8 specifically uses the communication topology $\hat{\mathcal{L}}$ that is defined by the Laplacian matrix $\mathcal{L}$ in (2.1.10) and the pinning matrix $\mathcal{P}$ in (2.1.11). Theorem A.5, however, only requires that all eigenvalues of $\hat{\mathcal{L}}$ are positive and real. This means that when α, β, and k are defined according to the constraints in corollary A.8, the dynamics of (2.8.7) are stable for any Laplacian matrix $\mathcal{L}$ and any pinning matrix $\mathcal{P}$ such that $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$ has positive real eigenvalues.*

By definition, the Laplacian matrix $\mathcal{L}$, and therefore also $\hat{\mathcal{L}}$, is diagonally dominant and has positive diagonal entries. From the Gershgorin disc theorem (lemma A.3) it follows that all eigenvalues of $\hat{\mathcal{L}}$ have nonnegative real parts. For all eigenvalues to be located in the open right half-plane the matrix $\hat{\mathcal{L}}$ has to be nonsingular. In [24] it is shown that $\hat{\mathcal{L}}$ is nonsingular when the graph corresponding to the communication topology defined by $\hat{\mathcal{L}}$ contains a directed spanning tree rooted at the virtual reference vehicle, which means that there exists a directed path between the virtual reference vehicle and every vehicle in the platoon. If the matrix $\hat{\mathcal{L}}$ is also triangular or symmetric, or if lemma A.2 applies, the eigenvalues are guaranteed to be real. Clearly, the graph that corresponds to the matrix $\hat{\mathcal{L}} = \mathcal{L} + \mathcal{P}$ defined in (2.1.10) and (2.1.11) contains a directed spanning tree whenever $\alpha > 0$ as is demonstrated by figure 2.3.