EINDHOVEN UNIVERSITY of TECHNOLOGY
Department of Mechanical Engineering
Manufacturing Networks

MASTER'S THESIS

Optimizing Traffic Light Control

A control theory approach

420689

by Stijn Fleuren

Eindhoven, August 14th, 2012

Supervisor:	Prof. dr. ir. I.J.B.F Adan
Coaches:	Dr. ir. A.A.J. Lefeber
	Dr. ir. M.A.A. Boon

MASTER PROJECT (INDIVIDUAL)

EINDHOVEN UNIVERSITY OF TECHNOLOGY July 2012
Department of Mechanical Engineering
Systems Engineering Group

Student	S.T.G. Fleuren
Supervisor	Prof.dr.ir I.J.B.F. Adan
Advisors	Dr.ir. A.A.J. Lefeber Dr.ir. M.A.A. Boon
Start	December 2011
Finish	August 2012
Title	Optimizing traffic light control - A control theory approach

Subject
With an increasing number of road users the efficiency of traffic lights gets more and more important. Traffic light schedules, defining when each traffic light is green, amber and red, could have a great effect on the circulation of traffic at intersections and on the circulation of traffic through cities. Another way to increase the circulation of traffic is by changing the road structure. Obviously, changing traffic light schedules is relatively cheap and easy to implement in comparison to changing road structure. Moreover, changing road structure might not be possible due to a lack of space.

Assignment
The main goal is to develop traffic light schedules that improve the efficiency of traffic lights at intersections, i.e. improve the circulation of traffic. These traffic light schedules should satisfy a number of constraints; minimum green times, maximum green times, maximum queue lengths, conflicting traffic lights may not be green at the same moment and for conflicting traffic lights clearance times are given (a traffic light may only switch to green whenever each of the conflicting traffic lights is red for a certain amount of time). In this project we will distinguish three problems. The first problem is *trajectory optimization*, which is deriving optimal periodic behavior of the intersection. This optimal periodic behavior can be derived using a hybrid fluid model to model an intersection. The second problem is *regulation*, which is finding a policy. A policy is a set of rules when to change the color of the traffic lights. When we deviate from the optimal periodic behavior, the policy should make sure that we will again return to this optimal periodic behavior. The third problem is accessing the *performance* of the policy in a *stochastic setting*.

Prof.dr.ir I.J.B.F. Adan Dr.ir. A.A.J. Lefeber Dr.ir. M.A.A. Boon

Systems Engineering Department of Mechanical Engineering

Summary

With an increasing number of road users the efficiency of traffic lights gets more and more important. Traffic light schedules could have a great effect on the circulation of traffic at intersections and on the circulation of traffic. To derive efficient traffic light control more research is needed.

Most of the research on this topic is devoted to fixed cycle traffic light control. For fixed cycle traffic light control the durations of the green periods as well as the durations of the red periods are not affected by traffic. This type of traffic light control can be efficient when a lot of traffic arrives at the traffic lights. However, when little traffic arrives, road users might have to wait in front of a traffic light for no apparent reason. For example, at night a road user might have to wait while there is no other traffic near this intersection.

Another type of traffic light control is vehicle-actuated traffic light control. In contrast to fixed cycle traffic light control, for vehicle-actuated traffic light control the durations of the green periods as well as the duration of the red periods are affected by arriving traffic. For vehicle-actuated traffic light control, via detectors information is gathered about the queue lengths at the intersection. This information is used to regulate the duration of the green and red periods.

In practice, for safety reasons restrictions on the duration of a green period are given: minimum green times and maximum green times. Most research devoted to vehicle-actuated traffic light control either does not regard these restrictions on green times or is restricted to one traffic light being green at a time. Furthermore, one of the most studied vehicle actuated traffic light controls is the exhaustive policy which switches a traffic light to red when its queue is emptied.

In this thesis we derive a vehicle-actuated traffic light control that does regard restrictions on green time duration, that is not restricted to one traffic light being green at a time and that is not restricted to switching a traffic light to red whenever its queue is cleared.

We discuss three problems in this thesis. The first problem is *trajectory optimization*, which is finding the desired behavior of the intersection. This desired behavior is derived by modeling the intersection with a hybrid fluid model. This hybrid fluid model traffic assumes deterministic arrivals and deterministic departures. We derive the desired behavior of the intersection, minimizing the average weighted queue length at the intersection. Since we assume deterministic arrivals and deterministic departures to derive the desired behavior of the intersection, in practice the intersection deviates from it due to for example stochastic arrivals.

The second problem is *regulation*. The problem of regulation is to find a policy (a feedback control), which is a set of rules (as function of the queue lengths at the traffic lights) that defines when to take what control actions (for example when to change the color of a traffic light). This policy should make sure that the intersection returns to the desired behavior whenever the intersection deviates from this desired behavior.

The third problem is to address the quality of the proposed policy in a stochastic environment. To this end, we use a stochastic model for the intersection. This stochastic model assumes Poisson arrivals.

In this thesis we consider relatively small intersections. However, in the future we will try to extend to larger intersections.

Summary (Dutch)

Door het toenemen van het aantal weggebruikers wordt de efficiëntie van verskeerslichten steeds belangrijker. Het aanpassen van verskeerslichtregelingen kan een groot effect hebben op de doorstroming van verkeer. Om efficiënte verskeerslichtregelingen te verkrijgen is onderzoek nodig.

Het grootste deel van het onderzoek is gedaan naar verskeerslichten met vaste groen- en roodtijden. Verkeerslichten met deze vaste afstellingen zijn niet afhankelijk van het aankomend verkeer. Dit type verskeerslichtregeling kan goed werken wanneer er veel verkeer aankomt bij de verskeerslichten. Wanneer weinig verkeer aankomt bij een kruispunt kan het zo zijn dat je moet wachten zonder duidelijke reden. Bijvoorbeeld wanneer 's nachts een auto aankomt bij een leeg kruispunt kan het zijn dat deze auto toch moet wachten voor een rood verskeerslicht.

Een ander type verskeerslichtregeling is de voertuigafhankelijke regeling. In tegenstelling tot een verskeerslicht met vaste groen- en roodtijden is de voertuigafhankelijke regeling wel afhankelijk van aankomend verkeer. Via meetlussen in de weg wordt informatie verkregen over het wachtend verkeer op een kruispunt. Deze informatie wordt gebruikt om groen- en roodtijden te bepalen.

In de praktijk worden er boven- en ondergrenzen gesteld op groentijden. Het merendeel van het onderzoek naar de voertuigafhankelijke regeling houdt geen rekening met deze grenzen op groentijden. Verder wordt er vaak aangenomen dat er hoogstens één verskeerslicht tegelijkertijd groen is. Een veel onderzochte voertuigafhankelijke regeling is de regeling waarbij een verskeerslicht rood wordt zodra er geen verkeer meer staat te wachten voor dit verskeerslicht.

In dit verslag beschouwen we een voertuigafhankelijke verskeerslichtregeling die wel rekening houdt met de boven- en ondergrenzen op groentijden, waarbij meerdere (niet conflicterende) verskeerslichten tegelijkertijd groen kunnen zijn en waarbij een verskeerslicht niet per sé rood wordt als er geen verkeer staat te wachten voor dit verskeerslicht.

Er worden drie problemen behandeld. Het eerste probleem wordt het *trajectory optimization* probleem genoemd. Voor het *trajectory optimization* probleem wordt het gewenste gedrag van een kruispunt afgeleid door het kruispunt te modeleren met een hybride vloeistof model. Dit hybride vloeistof model gaat uit van deterministische aankomsten en deterministische vertrekken. Tijdens het *trajectory optimization* probleem wordt gezocht naar het gedrag van het kruispunt dat de gewogen wachrijlengte aan het kruispunt minimaliseert. Om het gewenste gedrag van het kruispunt af te leiden wordt aangenomen dat de aankomsten en vertrekken deterministisch zijn. In de praktijk zijn de aankomsten en vertrekken stochastisch en zal het kruispunt van dit gewenste gedrag afwijken.

Het tweede probleem is *regulation*. Voor dit probleem wordt naar een feedback policy (als functie van de wachtrijlengtes bij de verskeerslichten) gezocht. De feedback policy bestaat uit regels die bepalen wanneer welke acties ondernomen moeten worden (bijvoorbeeld wanneer de kleur van een verskeerslicht moet veranderen). Deze regels moeten ervoor zorgen dat wanneer het kruispunt afwijkt van het gewenste gedrag (verkregen via het *trajectory optimization* probleem), het kruispunt weer terug gaat naar dit gewenste gedrag.

Verder worden deze regels getest in een stochastische omgeving. Via simulatie worden resultaten verkregen. Voor deze simulatie wordt het kruispunt gemodelleerd met een stochastisch model. Dit stochastische model neemt aan dat de aankomsten bij het kruispunt Poisson verdeeld zijn.

In dit verslag beschouwen we relatief kleine kruispunten. In de toekomst wordt geprobeerd om dit werk uit te breiden naar grotere kruispunten.

Contents

Chapter 1

Introduction

Traffic lights are signalling devices that control the access of competing traffic flows to an intersection. The main purpose of traffic lights is to improve safety and decrease discomfort of road users. The traffic lights origin, can be traced back to semaphores and lights used in regulating train traffic. The first traffic light was already in use before automobiles dominated the roads and traffic consisted of pedestrians, buggies and wagons. This traffic light was a rotating gas lantern with red and green lights and it was installed in 1868 outside the British Houses of Parliament in Londen. The gas lantern was very similar to the railway signals of the time and it was invented by railway engineer J.P. Knight. With the rise of the automobiles, the roads got more occupied. Hence, the traffic light got more useful and more practical. In 1920 the first four-way three color traffic light was installed in Detroid.

In the 1920s traffic lights were also introduced in the urban streets of almost every sizable Dutch city. Each of these cities developed its own traffic light system. In the 1930s this variety of systems gave way for the three-color system that would become the international standard.

With an increasing number of road users the efficiency of traffic lights gets more and more important. Traffic light schedules could have a great effect on the circulation of traffic at intersections and on the circulation of traffic. Another way to increase the circulation of traffic is by changing the road structure. Obviously, changing traffic light schedules is relatively cheap and easy to implement in comparison to changing road structure. Besides, changing road structure is not always possible. For example, because of limited space in cities, adding an extra lane to decrease traffic congestion is often out of the question.

The main goal of this thesis is to make a first step in developing vehicle-automated traffic light controls that improve the efficiency of traffic lights at intersections, i.e. improve the circulation of traffic. More specifically, our goal is to minimize the average weighted queue length for relatively small intersections. Minimizing the average queue length is equivalent to minimizing the average delay of road users at an intersection. In this thesis we define the delay as the additional travel time experienced by a driver, cyclist or pedestrian, which is the same definition as can be found in [14]. To minimize the average weighted queue length at an intersection, we distinguish two main roots of control theory cf. [24]: *trajectory optimization* and *regulation*.

In Section 1.1 we introduce some of the definitions and introduce some of the notation used in this thesis. We give a summary of the introduced definitions and the introduced notation in Section 1.2 and Section 1.3. In Section 1.4 we give the problem definition and in Section 1.5 we give an overview of this thesis.

1.1 Example: Intersection With 32 Traffic Lights

In this section we introduce some of the definitions and some of the notation used in this thesis. The definitions and notation introduced in this section are summarized in Section 1.2 and Section 1.3.

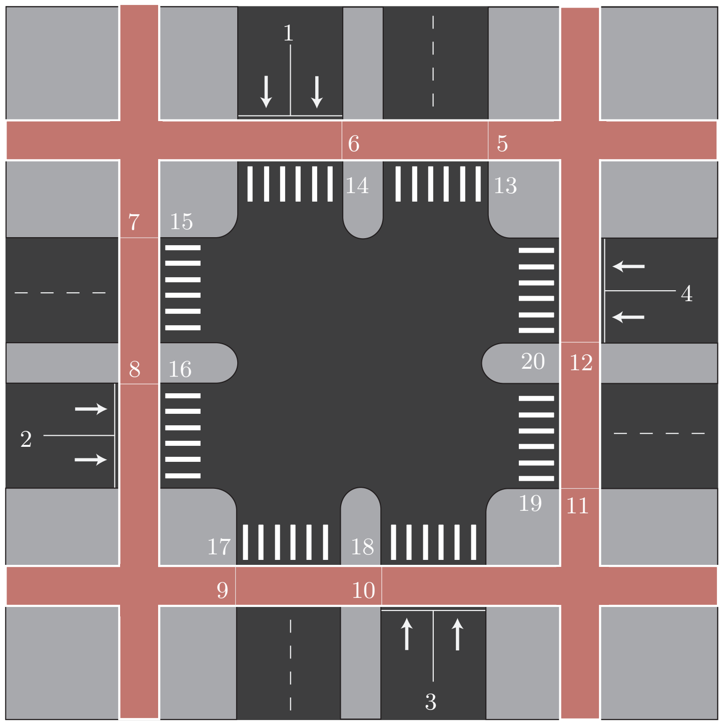


Figure 1.1: An example of an intersection controlled with traffic lights.

Vehicle lanes, bicycle lanes and pedestrian crossings In Figure 1.2 we can see an example of an intersection. This intersection consists of 8 vehicle lanes, 8 bicycle lanes and 8 (two-way) pedestrian crossings. A vehicle lane is defined as the part of the road leading to the intersection that is designed for a single line of vehicles. A bicycle lane is a portion of the roadway that has been designated by striping, signing, and pavement markings for the exclusive use of bicyclists. In this example each vehicle lane and each bicycle lane is equipped with one traffic light. We define a pedestrian crossing as a designated place where pedestrians can cross a street and where vehicles must stop to let them cross. Generally, at a pedestrian crossing pedestrians can walk in two directions (two-way pedestrian crossing). Each direction is equipped with a traffic light. Hence, this intersection has a total of 32 traffic lights.

The intersection depicted in Figure 1.2 is not very realistic due to its lack of vehicle lanes for turning traffic. However, we have chosen this example because it is one of the more complex intersections that we could control using the results from this thesis.

Signals and approaches We can partition each of the traffic lights in exactly one set, each set corresponding to one signal. A signal is a set of one or more traffic lights, which switch to green simultaneously and switch to red simultaneously. A Signals is designated to either vehicles, cyclists

Signal	Type of traffic	number of traffic lights in each signal
1,...,4	Vehicles going straight ahead	2
5,12	Cyclists (in this example cyclists are not allowed to not go clockwise)	1
13,...,20	Pedestrians.	2

Table 1.1: partitioning the 32 traffic lights in Figure 1.2 into 20 signals

or pedestrians. Two traffic lights may only be partitioned in the same signal designated to vehicles or cyclists whenever the traffic arriving at these traffic lights originates from the same direction. Signals are numbered $1, ..., N$ and we use $\mathcal{N} = \{1, 2, \ldots, N\}$ to refer to the set of all signals at an intersection. With approach $i \in \mathcal{N}$ we refer to the roads that lead to the traffic lights in signal i. In Figure 1.2 and Table 1.1 we can see how we have numbered the approaches (and thus how we have numbered the signals) respectively how we have partitioned the 32 traffic lights in 20 signals. Thus, in this example $N = 20$. Another division of traffic lights in signals is also possible. The division of traffic lights in signals is up to the engineers that derive traffic light schedules for an intersection. The traffic lights that are partitioned into the same signal switch simultaneously. Therefore, these traffic lights have exactly the same traffic light schedule. For this example, by partitioning traffic lights into signals there are 20 different traffic light schedules (one for every signal) instead of 32 different traffic light schedules.

In each direction there are two lanes for vehicles. Because the traffic from these two adjacent lanes originates from the same direction and goes in the same direction it is logical to switch the corresponding two traffic lights to green simultaneously and to red simultaneously. Hence, we partition these traffic lights in the same signal. Moreover, for this example we partition the two traffic lights of a pedestrian crossing in the same signal.

Signal state In most countries, the state of each of these signals can be either green, amber or red. We define the signal state as the visual state of the traffic lights that are element of the same signal.

However, in this thesis we do not consider the amber (orange) signal state because depending on the assumptions this amber signal state could be modeled as a red signal state, a green signal state or a combination of those two. When assuming that traffic still departs when its signal is amber, the amber signal state can be modeled as a green signal state. When assuming that traffic does not depart when its signal is amber, the amber signal state can be modeled as a red signal state. When assuming that during the first part of the amber signal state traffic departs and during the second part traffic does not depart, we can model the first part as a green signal state and the second part as a red signal state.

We use $m_i(t)$ for the signal state of signal $i \in \mathcal{N}$ at time $t \in \mathbb{R}^+$. The signal state $m_i(t)$ is equal to ① or ❶ whenever the signal state is green or red respectively. When the signal state of signal $i \in \mathcal{N}$ is red, we often use the short version: signal i is red. When the signal state of signal $i \in \mathcal{N}$ is green we often use one of the short versions: signal i is served or signal i is green.

Green and red period We define a green (red) period as the interval during which the signal state is green (red), i.e. the interval between the moment that the signal is switched to green (red) and the moment that the signal is switched to red (green). During the green period of signal i, traffic waiting at the corresponding approach is allowed to cross the common crossing area. On the other hand, during the red period of signal i, traffic from the corresponding approach is not allowed to cross the common crossing area. The duration of a green (red) period is called a green (red) time. We use g_i^k, $k = 1, 2, \ldots$ for the kth green time of signal i (starting from $t = 0$) and we use r_i^k for the red time of signal i that comes between g_i^k and g_i^{k+1}. A traffic light schedule is a specified sequence of red and green periods for a traffic light.

In practice, signals generally have constraints on the length of the green period, i.e. a green time may not exceed the maximum green time and a green time must exceed the minimum green time. We denote the maximum green time and the minimum green time of signal $i \in \mathcal{N}$ with g_i^{max} respectively g_i^{min}, where $g_i^{max} > 0$ and $g_i^{min} \geq 0$. Whenever a green period is extremely short or extremely long (and as a result a red period of another signal is extremely long), road users could get irritated which results in more red negation, i.e. in more people ignoring a red light. Further, whenever a green period is extremely short or extremely long, road users might think the traffic lights are malfunctioning. These lower and upper boundaries on green times should guarantee that the intersection is believable and should limit the irritation of road users.

Arrival rate We assume that at all of the traffic lights, traffic arrives. How much traffic arrives is denoted with the arrival rate. The arrival rate is the mean number of vehicles, cyclists or pedestrians arriving at a signal or traffic light per second. In practice this arrival rate varies. For example more traffic arrives during rush hour. However we assume that the arrival rate at signal $i \in \mathcal{N}$ is constant. The arrival rate at signal $i \in \mathcal{N}$ is denoted with λ_i and it is often obtained by counting the number of vehicles, cyclists or pedestrians arriving at a traffic light. We assume $\lambda_i > 0$.

Maximum departure rate The maximum departure rate is the highest possible rate at which traffic from a traffic light or signal could cross the intersection in vehicles per second, cyclists per second or pedestrians per second. We use μ_i for the maximum departure rate of signal $i \in \mathcal{N}$. In practice this maximum departure rate is not constant because at the beginning of a green time there is a startup effect; people do not respond instantaneously and the traffic needs some time to accelerate. Hence, in the beginning the maximum departure rate increases. After a certain amount of time the maximum departure rate does not change anymore. The transient part, where the maximum departure rate increases, is called the startup effect in the departure rate. The maximum departure rate μ_i can be easily computed, based on the design of the intersection. The maximum departure rate satisfies $\mu_i > 0$. We assume $\mu_i > \lambda_i$. Further, we use ρ_i for the ratio between the arrival rate and the maximum departure rate of signal i, i.e. $\rho_i = \frac{\lambda_i}{\mu_i}$. $i \in \mathcal{N}$ which is equal to $\frac{\lambda_i}{\mu_i}$.

Queues The traffic waiting at a traffic light forms a queue. A queue is defined as the vehicles, cyclists and pedestrians at an approach that are waiting to cross the intersection. With queue $i \in \mathcal{N}$ we refer to the vehicles, cyclists and pedestrians that are waiting at approach i. We use $x_i(t)$ to refer to the queue length of queue $i \in \mathcal{N}$. Since there is limited space for traffic to wait at a traffic light, there are maximum queue lengths. The maximum queue length is the maximum amount of vehicles, cyclists or pedestrian that could be waiting in front of a traffic light or signal. These maximum queue lengths follow from the design of the intersection. The maximum queue length of queue $i \in \mathcal{N}$ is denoted with $x_i^{max}(t)$.

Slow mode During a slow mode a signal is green and the corresponding queue is empty. During a slow mode arriving traffic experiences no delay. We use $g_i^{\lambda,k}$ for the length of the slow mode at signal i during the kth green period of signal i. We use $g_i^{\mu,k}$ for the length of the interval during the kth green period of signal i during which the queue of signal i is not empty, i.e. $g_i^{\mu,k} = g_i^k - g_i^{\lambda,k}$.

Conflicting signals For safety reasons, not all signals at an intersection that is controlled with traffic lights may be green simultaneously (if this was the case we would not need traffic lights). Two signals are conflicting when the traffic from these approaches cannot safely cross the common crossing area simultaneously. In Table 1.2 we present the conflict matrix for the intersection in Figure 1.2. In a conflict matrix we can see which signals are conflicting (denoted with an 'x') and which are not conflicting.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		x		x		x			x					x			x			
2	x		x					x			x					x			x	
3		x		x	x					x			x					x		
4	x		x				x					x			x					x
5			x																	
6	x																			
7				x																
8		x																		
9	x																			
10			x																	
11		x																		
12				x																
13			x																	
14	x																			
15				x																
16		x																		
17	x																			
18			x																	
19		x																		
20				x																

Table 1.2: Conflict matrix of the intersection in Figure 1.2.

Some of the vehicles, cyclists and pedestrians already start to accelerate when they expect their signal to switch to green. When the order in which these signals are served changes, these expectations may be wrong and could result in unsafe situations. Hence, in practice, often signals are served in a fixed order. To define the order in which we serve these signals, we use signal groups. A signal group is a group of signals that do not conflict with each other. The order in which we serve these signal groups is always the same.

When we have more than 2 signal groups, we have to determine the order in which to serve the signals. For example in the case of 3 signal groups (group 1, group 2 and group 3) we can choose to serve the signals in the following orders:

order 1 serve the signals in group 1 $\rightarrow$ serve the signals in group 2 $\rightarrow$ serve the signals in group 3 $\rightarrow$ serve the signals in group 1 $\rightarrow$ serve the signals in group 2 $\rightarrow$ serve the signals in group 3 $\rightarrow \dots$

order 2 serve the signals in group 1 $\rightarrow$ serve the signals in group 3 $\rightarrow$ serve the signals in group 2 $\rightarrow$ serve the signals in group 1 $\rightarrow$ serve the signals in group 3 $\rightarrow$ serve the signals in group 2 $\rightarrow \dots$

One way to find signal groups from a conflict matrix is by solving a graph coloring problem. The vertices of the graph coloring problem represent the signals. Two vertices are connected (with an edge) whenever the corresponding signals are conflicting. We can find the minimum number of signal groups needed, by coloring the vertices with a minimum number of colors such that two connected vertices do not have the same color.

For the intersection in Figure 1.2 this graph coloring problem results in two signal groups. One of the groups contains the signals $1, 3, 7, 8, 11, 12, 15, 16, 19$ and 20. The other group contains the signals $2, 4, 5, 6, 9, 10, 13, 14, 17$ and 18.

Determining the signal groups and determining in which order to serve these signal groups is not in the scope of this thesis. In this thesis we assume that the signal groups are given. Furthermore, we only consider intersections with 2 signal groups.

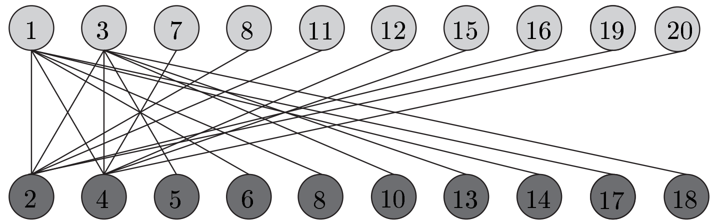


Figure 1.2: Graph coloring for the intersection in Figure 1.2.

Setup times A setup time is a fixed minimum time between the end of the green period of a signal and the beginning of the green period of a conflicting signal. A setup time is a safety measure that limits the hinder for traffic crossing the intersection. The setup time between serving signal $i \in \mathcal{N}$ and serving signal $j \in \mathcal{N}$ is denoted with $\sigma_{i,j}$. During this setup time, signal i and signal j are both red. In practice, setup times can be negative. Whenever $\sigma_{i,j} < 0$ this means that signal j may switch to green (a maximum of) $|\sigma_{i,j}|$ seconds before signal i has switched to red. For the intersection in Figure 1.2, possibly signal 3 may switch to green before signal 5 has switched to red because it takes some time for the vehicles from signal 3 to reach the part of the road that cyclists from signal 5 also use. Furthermore, we should also note that in general $\sigma_{i,j}$ and $\sigma_{j,i}$ are not equal. for the intersection in Figure 1.2 it probably holds that $\sigma_{1,4} < \sigma_{4,1}$ because it takes longer for vehicles from signal 4 (than for vehicles from signal 1) to arrive at the part of the intersection that both the vehicles from signal 1 and the vehicles from signal 4 use. Further, we use $\sigma_{i,j,i} = \sigma_{i,j} + \sigma_{j,i}$.

In this thesis we restrict ourselves to non-negative setup times.

1.2 Terminology

Below we list some of the terminology used in this thesis.

Approach	=	Approach i refers to the roads that lead to the traffic lights in signal i, $i = 1, ..., N$.
Bicycle lane	=	Portion of the roadway that has been designated by striping, signing, and pavement markings for the exclusive use of bicyclists.
Conflicting signals	=	Two signals are conflicting when the traffic from the corresponding approaches cannot safely cross the common crossing area simultaneously.
Delay	=	Additional travel time experienced by a driver, cyclist or pedestrian with respect to the travel time of a driver, cyclist or pedestrian that arrives at the same traffic light during a green period while the queue is empty.
Green period	=	Interval during which the signal state is green, i.e. the interval between the moment that the signal is switched to green and the moment that the signal is switched to red. During the green period of signal i, traffic from the queue of the corresponding approach can cross the common crossing area.
Green time	=	Duration of a green period.
Intersection	=	Set of approaches and a common crossing area.
Pedestrian crossing	=	A designated place where pedestrians can cross a street and where vehicles must stop to let them cross.
Queue	=	The vehicles, cyclists and pedestrians at an approach that are waiting to cross the intersection. With queue i we refer to the vehicles, cyclists and pedestrians that are waiting at approach i.
Red period	=	Interval during which the signal state is red, i.e. the interval between the moment that the signal is switched to red and the moment that the signal is switched to green.
Red time	=	Duration of a red period.
Setup time	=	Fixed minimum time between the end of the green period of a signal and the beginning of the green period of a conflicting signal.
Signal	=	Set of one or more traffic lights which switch simultaneously to green and simultaneously to red. A Signals is designated to either vehicles, cyclists or pedestrians. Two traffic lights may only be partitioned in the same signal designated to vehicles or cyclists whenever the traffic arriving at these traffic lights originates from the same direction. Signals are numbered $1, ..., N$.
Signal group	=	A group of signals that do not conflict with each other.
Signal state	=	The visual state, i.e. green or red, of the traffic lights that are element of the same signal. We do not consider the amber state. Vehicles, cyclists and pedestrians are assumed to depart only when the signal state of their signal is green.
Switch	=	Change in the state of a signal.
Slow mode	=	Interval during which the signal state is green and the corresponding queue is empty. The traffic that arrives during a slow mode experiences no delay.
Vehicle lane	=	Part of the road leading to the intersection that is designed for a single line of vehicles.

1.3 Notations

In this section we list some of the notation used in this thesis.

λ_i	=	Arrival rate at signal i, i.e. the mean number of vehicles, cyclists or pedestrians arriving at approach i per second.
μ_i	=	Maximum departure rate of signal i, i.e. the maximum rate at which traffic from signal i could cross the intersection in vehicles per second, cyclists per second or pedestrians per second.
$\mathcal{N}$	=	The set of all signals at an intersection, i.e. $\mathcal{N} = \{1, 2, \ldots, N\}$
ρ_i	=	Ratio between the arrival rate and the maximum departure rate of signal i, i.e. $\rho_i = \frac{\lambda_i}{\mu_i}$.
$\sigma_{i,j}$	=	Setup time between serving signal i and serving signal j. During this setup time signal i and signal j are both red, i.e. during this setup, traffic from signals i and j may not cross the intersection.
$\sigma_{i,j,i}$	=	$\sigma_{i,j} + \sigma_{j,i}$.
$m_i(t)$	=	Signal state of signal i at time $t \in \mathbb{R}^+$. The signal state $m_i(t)$ is equal to ⓘ or ❶ whenever the signal state is green or red respectively.
g_i^k	=	Length of the kth green period period of signal i.
$g_i^{\lambda,k}$	=	Length of the slow mode at signal i during the kth green period of signal i.
$g_i^{\mu,k}$	=	Length of the interval during the kth green period of signal i during which the queue of signal i is not empty, i.e. $g_i^{\mu,k} = g_i^k - g_i^{\lambda,k}$.
r_i^k	=	Length of the red period of signal i that comes between the kth green period of signal i and the $k+1$th green period of signal i.
g_i^{max}	=	Upper boundary on the length of the green period of signal i, i.e. the maximum green time.
g_i^{min}	=	Lower boundary on the length of the green period of signal i, i.e. the minimum green time.
$x_i(t)$	=	Amount of traffic in queue i at time $t \in \mathbb{R}^+$, i.e. the queue length of queue i at time t.
x_i^{max}	=	Maximum queue length of queue i. The maximum queue length of queue i is the maximum amount of traffic that could be waiting at approach i.

1.4 Problem Description

In this thesis we discuss three problems.

Problem 1: trajectory optimization The first problem is trajectory optimization. The problem of trajectory optimization is the process of designing a trajectory that minimizes or maximizes some measure of performance within prescribed constraint boundaries. For small intersections we want to find optimal trajectories, minimizing the average weighted queue length at the intersection. These optimal trajectories can be seen as the desired behavior of the intersection. We assume that we can control the signal state and the departure rate of traffic at each of the signals of an intersection.

An example of a trajectory optimization problem is the problem of finding an optimal flight trajectory of an airplane from Vancouver to Cape town. Before the pilot even starts flying he could already derive some trajectory he wants to follow. For example the fastest trajectory. Finding this trajectory is the trajectory optimization problem.

For an intersection a trajectory consists out of the evolution (as function of time) of the following variables for each of the signals $i \in \mathcal{N}$:

- $x_i(t)$: the queue lengths of signal i as a function of time.
- $m_i(t)$: the signal state, also called the mode, of signal i as a function of time.
- $d_i(t)$: the departure rate of both signal i as a function of time.

To solve the trajectory optimization problem for an intersection, we model the intersection with a hybrid fluid model (see Section 3.2). This model assumes deterministic arrivals and deterministic departures. In practice there are stochastic effects. However, this deterministic model is more suitable for optimization purposes.

Problem 2: regulation To obtain the desired trajectory we assume deterministic arrivals and deterministic departures. However, due to stochastic effects we may deviate from the desired trajectory. The second problem is regulation, which is finding a policy (a feedback control). A policy is the set of rules that defines when to take what control actions (for example when to change the color of a traffic light). When we deviate from the desired trajectory, the policy should make sure that we again return to this desired trajectory.

An example of a regulation problem is when an airplane wants to follow some trajectory (for example the fastest trajectory). This desired trajectory follows from the trajectory optimization problem. However, due to factors like chaotic airflow, the airplane cannot follow this desired trajectory exactly. Whenever the airplane deviates from this desired trajectory, the pilot can observe this and he could correct the airplane in the right direction by steering, accelerating or decelerating. In this way the airplane returns to the desired trajectory whenever it deviates from it. What actions the pilot should take (and when it should take these actions) is the problem of regulation.

Problem 3: Performance in a stochastic setting The third problem is accessing the performance of the policy in a stochastic setting via simulation.

1.5 Outline of this thesis

First we give an overview of the literature about traffic light control in Chapter 2. Subsequently, we introduce two models, i.e. the stochastic model (SM) and the hybrid fluid model (HFM), in Chapter 3. The hybrid fluid model is used in the largest part of this thesis.

In chapters 4, 5 and 6 we consider a simple intersection of two signals. Subsequently, we consider a more general intersection (with two signal groups) in chapters 7, 8 and 9.

In Chapter 4 and Chapter 7 we address the first problem: trajectory optimization. First we prove that we can always find an optimal trajectories satisfying some properties. Using these properties we can pose an optimization problem. This optimization problem can be solved analytically for some intersections.

In chapters 5 and 8 we consider the second problem; regulation. In these chapters we propose a policy (a feedback control) and we prove that this policy works as desired; the policy makes sure that we return to the desired periodic optimal trajectory whenever we deviate from it. This policy does not need to control the departure rates. We only need to control the signal states.

In chapters 6 and 9 we address the quality of the proposed policy in a stochastic environment.

Finally, in Chapter 10 we summarize the most important conclusions of this thesis and we give some recommendations for future research.

Chapter 2

Literature

There are two primary types of traffic light control: fixed cycle traffic light control and vehicle-actuated traffic light control. For fixed cycle traffic light control the duration of the green period as well as the duration of the red period are not affected by traffic, i.e. the green time of a traffic light is always the same and the red time of a traffic light is always the same. For a vehicle-actuated traffic light control, via detectors information is gathered about the queue lengths at the intersection. This information is used to regulate the red and green times. We give an overview of the literature about both types of traffic light control.

Fixed Cycle Traffic Light Control There has been a broad effort to obtain exact expressions and good approximations for the queue length and the delay of vehicles at intersections with traffic lights. The delay is often used as an optimization and evaluation criterion for traffic light control. However, it is not easy to determine the delay.

In the effort to obtain good approximations for the delay, the fixed-cycle traffic light (FCTL) queue is one of the best-studied models in traffic engineering. For this model, the traffic light alternates between green and red periods of effective duration g and r and the vehicles that arrive at a traffic light form a queue. It is assumed that during the green periods traffic can depart at equal time intervals.

For the FCTL queue the duration of the green period as well as the duration of the red period are not affected by traffic, i.e. the green time of a traffic light is always the same and the red time of a traffic light is always the same. The majority of the research devoted to the FCTL queue is based on the simplifying assumption that traffic arrives according to a Poisson process. The most famous result is that from Webster [28]. It gives the mean delay of a vehicle in closed form, which is partly based on theoretical grounds and partly based on simulation.

Other expressions for the mean delay, assuming that traffic arrives according to a (Compound) Poisson process, can be found in Darroch [8], McNeill [18] and Webster and Cobe [29]. In McNeill [18] an exact expression for the mean delay was given up to one unknown: the mean queue length at the end of a green period. We denote this unknown with $\mathbb{E}X_g$. Later, in [8], Darroch found an exact expression for $\mathbb{E}X_g$. In [23], Ohno gives a detailed description of a computational algorithm for several characteristics such as the average delay, the average queue length and the probability of clearing the queue. This computational algorithm calculates the (rather complicated) expression for $\mathbb{E}X_g$ given in Darroch [8] in an exact manner. Further, Ohno gives an overview of new approximate expressions and existing approximate expressions in comparison with the exact values of the average delay. In Heidemann [13] analytical results on statistical distributions of queue lengths and delays at traffic lights are derived. To obtain these probability distributions Poisson arrivals are assumed.

There is also some research devoted to more general types of arrival processes. In van Leeuwaarden [16], a probability generating function is given for the queue length distribution at the end of a green period and a Laplace-Stieltjes transform is given for the delay distribution. To obtain this probability generating function and Laplace-Stieltjes transform, the queue is modeled in discrete-time and it is assumed that the number of vehicles that arrive per time slot follows some discrete distribution (the Poisson distribution and Compound Poisson distribution are also discrete distributions). Also in Van den Broek et al. [6], a more general discrete distribution is considered and several bounds and approximations are presented for the average delay. Further, in [6] a new approximation is given, based on the heavy traffic limit and a scaling argument. In Miller [19] and Newell [20] approximations for $\mathbb{E}X_g$ are derived using fairly general arguments.

More recently, a probabilistic queuing model is used in Viti and Van Zuylen [26], assuming any temporal distribution of the arrivals. It can explain the dynamic and stochastic behavior of queues at fixed-time controlled intersections and allows one to capture the temporal behavior of queues, as well as the uncertainty of a prediction.

We have shown that there are several exact expressions and approximations available for the delay. These expressions can be used to find the optimal fixed cycle traffic light control. In Webster [28], a technique is proposed, that uses Webster's famous delay formula, to find fixed cycle control schemes for an isolated intersection. In Van den Broek [5], a mixed integer program is given for finding the optimal control and an algorithm is proposed to solve this mixed integer program. In Fouladvand and M. Nematollahi [11] the analytical solutions were found for a fixed-time controlled intersection of two one-way streets and a fixed-time controlled intersection of a two-way street with a one-way street. To find this analytical solution, constant arrival rates and constant departure rates are assumed. Further, no setup times, no constraints on green times and no maximum queue lengths are considered. An algorithm for designing traffic light schedules is proposed in Riedel et. al [25]. The model of an intersection is derived by considering a small intersection. Using a combination of dynamic programming and branch and bound, a control algorithm is developed.

Further, there is also some research devoted to networks of intersections. In Brockfeld et al. [4] the goal is to minimize travel times for a city network: a square lattice of intersections. To this end, the network is modeled with a cellar automata model. For synchronized traffic lights, one finds strong oscillations in the global flow as function of the cycle time. Further, green wave and random switching strategies are tested. In Alfa and Neuts [1], the arrival process is modeled using a discrete-time Markovian Process. This model takes into account the bunching of traffic, i.e. forming of platoons, and the correlations between inter arrival times. They conclude that ignoring the correlation in the arrival process results in the underestimation of performance measures such as the mean queue length, especially at high traffic intensities.

Vehicle-actuated Traffic Light Control For a vehicle-actuated traffic light control, via detectors information is gathered about the queue lengths at the intersection. This information is used to regulate the red and green times.

One of the most studied vehicle-actuated traffic light controls is the exhaustive policy that switches a traffic light to red when the queue is cleared. One of the first efforts in analyzing vehicle-actuated traffic light control is done in Darroch [9]. In Darroch [9] Poisson arrivals are assumed to analyze the exhaustive policy for an intersection of two one-way streets. In Newell [21] this exhaustive policy is analyzed using fluid and diffusion queueing approximations for an intersection of two one-way streets. In [21] arriving traffic is assumed to be a stationary stochastic process with an arrival rate just slightly below that which saturates the intersection. Newell concludes that the vehicle-actuated traffic light control has a high efficiency compared to the fixed cycle traffic light control.

In Daganzo [7] and Boon [2] polling models with more general arrivals and departure processes are used to model and analyze intersections. These polling models are either restricted to serving one flow of traffic at a time or restricted to the exhaustive policy.

More recently an intersection of two intersecting traffic flows is considered in Wang and Yin [27]. Wan and Yin analyze green extensions; after a queue is cleared, arriving vehicles activate a green period extension during a period called the critical gap. When no vehicles arrive during the critical gap, the signal is switched to red. Wang finds that the optimal critical gap is generally not zero, which indicates that the exhaustive policy even in heavy traffic is not optimal.

Some research is devoted to intersection where multiple signals are served simultaneously. Haijema and Van der Wal [12] consider an intersection with a number of signals. The set of all signals is partitioned into signal groups. The problem concerning when to switch (and which signal group to serve next) is modeled as a Markovian decision process in discrete time. In [22] the analysis of a vehicle-actuated intersection from Newell [21] is extended to an intersection of two two-way streets (four-way intersection). They conclude that the high efficiency of a vehicle-actuated traffic light control, as found in Newell [21], does not necessarily hold for the case of two-way streets. Further, in Lämmer and Helbing [15] a self-organization approach to traffic light control is proposed. This self-organization approach is inspired by self-organized oscillations of pedestrian flows at bottlenecks. The control strategy is a combination of two complementary control regimes, an optimizing regime and a stabilizing regime.

Chapter 3

Models

In this chapter we introduce two models for an intersection. Both models are used in this thesis. The first model is a stochastic model (SM) and the second model is a hybrid fluid model (HFM). We model an intersection with a hybrid fluid model for the trajectory optimization problem and for the regulation problem. The stochastic model (SM) be used to obtain simulation results. For the trajectory optimization problem we assume that we can control the departure rate of traffic at each signal. However, it turns out that the policy that we propose does not need to control the departure rate. Thus, for the stochastic model we do not assume that we can control the departure rate, we always let traffic depart at the highest possible rate.

We show that we could model mixed arrival flows with a hybrid fluid model. Further, we show how to model two-way pedestrian crossings for the hybrid fluid model. We do not model mixed arrival flows and two-way pedestrian crossings in our stochastic model.

3.1 Stochastic Model

We describe the behavior of an intersection with a stochastic queueing model. For this stochastic queueing model we assume a Poisson arrival process and a deterministic departure process. We model each traffic light (also those in the same signal) separately. Each of these traffic lights has one queue, an arrival process, and a departure process. We assume that the arrival processes and the departure processes of the different traffic lights are independent. However, in practice this might not be the case. When for example a vehicle arrives at signal $1, 2, 3$ or 4 in Figure 1.2, the driver decides at which lane to wait and the arrival and departure processes of these traffic lights are not independent.

In this section we use slightly different notation than in the rest of this thesis. In this section we use:

- λ_i for the arrival rate at traffic light $i = 1, ..., N_{tl}$, where N_{tl} is the number of traffic lights at the intersection. In the rest of this thesis λ_i refers to the arrival rate at a signal and not at a traffic light.
- μ_i for the maximum departure rate at traffic light $i = 1, ..., N_{tl}$. In the rest of this thesis μ_i refers to the maximum departure rate of a signal and not at a traffic light.
- $\sigma_{i,j}$ for the setup time from traffic light $i = 1, ..., N_{tl}$ to (conflicting) traffic light $j = 1, ..., N_{tl}$. This setup time is the fixed minimum time between the end the green period of traffic light i and the beginning of the green period of conflicting traffic light j. Normally $\sigma_{i,j}$ refers to the setup time between two signals and not the setup time between two traffic lights. $\sigma_{i,j}$ is equal to the setup time from the signal that traffic light $i = 1, ..., N_{tl}$ is element of to the signal that traffic light

$j = 1, ..., N_{tl}$ is element of. Two traffic lights are conflicting when the corresponding signals of these two traffic lights are conflicting.

3.1.1 Arrival Processes and Departure Processes

We assume that the inter-arrival times at traffic light $i = 1, ..., N_{tl}$ are exponentially distributed with mean $\frac{1}{\lambda_i}$ which means that we consider an isolated intersection. We define the arrival time as the time at which a vehicle, cyclist or pedestrian would have crossed the stop line if its traffic light was green and no traffic was waiting at that traffic light. Note that this assumption of exponential inter-arrival times is not valid for a sequence of intersections because vehicles arrive in so called platoons. Platoons occur especially when the distance between two connected intersections is small. A platoon is a group of vehicles, cyclists or pedestrians traveling together. When platoons arise the arrival rate fluctuates and the inter-arrival times are not independent.

Each traffic light has a separate departure process. The departure process is assumed to be deterministic. Whenever there is traffic waiting in front of traffic light $i = 1, ..., N_{tl}$ at the beginning of a green period then a departure process is started. We register a departure at the moment that a vehicle, cyclist or pedestrian has entirely crossed the stop line which occurs $\frac{1}{\mu_i}$ seconds after the start of this departure process. When a departure is registered the next vehicle, cyclist or pedestrian (if present) can start its departure process. This inter departure time $\frac{1}{\mu_i}$ is assumed to be constant. In Figure 3.1 we show the departures during a green period whenever the queue is not empty during the whole green period. Since at each traffic light at most one departure process is active at a time, this may not be the best way to model the departures of cyclists and pedestrians because in practice cyclists and pedestrians can depart with more than one at a time.

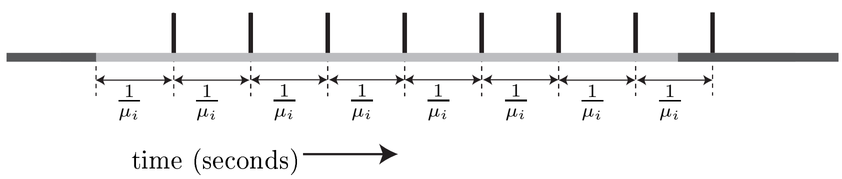


Figure 3.1: Departures during a green period of traffic light $i = 1, ..., N_{tl}$. The color of the time line is light gray and dark gray whenever the traffic light is green respectively red. A departure is visualized with a bold black vertical line.

The departure time is defined as the time at which a vehicle, cyclist or pedestrian actually crosses the stop line. Hence, the delay of a vehicle, cyclist or pedestrian is equal to the difference between its departure time and its arrival time. Whenever a vehicle, cyclist or pedestrian arrives when the queue is empty and the corresponding traffic light is green, this vehicle, cyclist or pedestrian could depart immediately. In this case its departure time is equal to its arrival time and it experiences a delay of zero seconds. Whenever the queue is emptied during a green period it stays empty during this green period.

3.1.2 Queue

Vehicles that have to wait in front of a traffic light form a queue. We model the queue with a FIFO (First-In-First-Out) buffer. The queue length is a non-negative integer. We assume that traffic arrives at the queue at the arrival time; the queue length increases with one at the moment of an arrival time. Furthermore, the queue length decreases with one at the moment of a departure.

There is no difference between a vehicle, cyclist or pedestrian that arrives when the maximum queue length is reached (or exceeded) and a vehicle, cyclist or pedestrian that arrives when the maximum queue length is not reached.

3.1.3 Modeling Startup Effect in the Maximum Departure Rate

We assume in our stochastic model that the inter departure time $\frac{1}{\mu_i}$ at traffic light $i = 1, ..., N_{tl}$ is constant. However, at the beginning of a green time there is a startup effect; people do not respond instantaneously and traffic needs some time to accelerate. Hence, the inter departure time is not constant; it decreases and after a certain amount of time the inter departure time does not change anymore. The transient part, where the inter departure time decreases, is called the startup effect. We assume that inter departure times are deterministic (also during the startup effect) and we assume the the startup effect always has the same duration at a traffic light.

We can model this startup effect by adapting the duration of the setups. Assume for example that the startup effect at traffic light $i = 1, .., N_{tl}$ takes 5.0 seconds. In these 5.0 seconds, 2 vehicles could depart. Hereafter, every 2 seconds a vehicle could depart (hence, $\frac{1}{\mu_i} = 2$)(see Figure 3.2). Assuming that the startup effect takes less than the minimum green time we can get the same number of departures during a green period by taking the departure rate equal to zero vehicles per second in the first second and hereafter equal to 0.5 vehicles per second. Traffic cannot depart during the red period and therefore this maximum departure rate equal to zero vehicles per second can be realized by increasing the red period of this traffic light with one second. Increasing the red period of traffic light i with one second can be realized by increasing the setup time $\sigma_{j,i}$ with 1 second, for all traffic lights $j = 1, ..., N_{tl}$ that are conflicting with traffic light i.

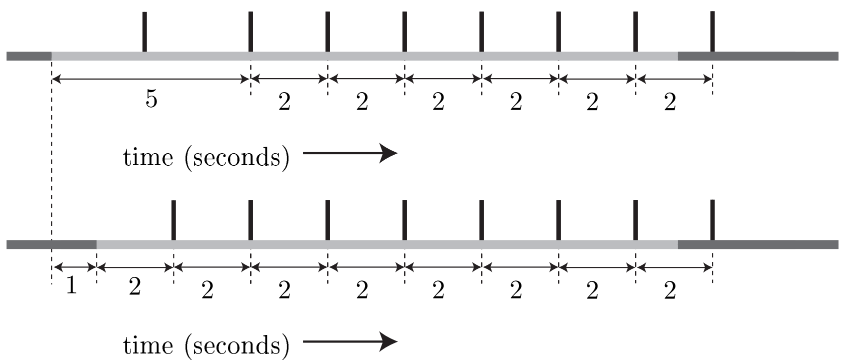


Figure 3.2: Departures during a green period of a traffic light. The color of the time line is light gray and dark gray whenever the traffic light is green respectively red. A departure is visualized with a bold black vertical line. The upper time line shows the departures with startup effect. The lower time line shows the departures without startup effect, but with larger setup times.

More general, when $t_{startup}$ is the duration (in seconds) of the startup effect at signal $i = 1, .., N_{tl}$ and D is the number of vehicles, cyclists or pedestrians that could depart during this startup effect at signal i then we model this startup effect by increasing the setups $\sigma_{j,i}$ with $t_{startup} - \frac{D}{\mu_i}$ seconds for all traffic lights j that are conflicting with traffic light i. During the green period, every $\frac{1}{\mu_i}$ seconds one vehicle, cyclist or pedestrian departs.

3.2 Hybrid Fluid Model

With the hybrid fluid model we can approximate the behavior of the stochastic model. We use the hybrid fluid model for the trajectory optimization problem because it is more suitable for optimization purposes. This makes it relatively easy (in comparison to using the stochastic model) to solve the trajectory optimization problem.

For the hybrid fluid model, traffic is modeled as a fluid. First we consider only one type of traffic arriving at a signal (for example only cars or only pedestrians) and we show how we model queues, arrivals and departures for the hybrid fluid model. Subsequently, we illustrate how to deal with mixed flows, i.e. signals and traffic lights where different types of traffic arrive. We also show how to model a two-way pedestrian crossing and how to incorporate a startup effect in the maximum departure rate. For the hybrid fluid model we model each signal with one queue, one arrival process and one departure process. The arrival process at signal $i \in \mathcal{N}$ is defined by a constant arrival rate λ_i and the departure process is defined by a constant maximum departure rate μ_i.

3.2.1 Modeling Queues

Because we model traffic as a fluid, the queue length is a non-negative real value. For example, the queue length at a signal could be equal to 0.75 cars.

Because there is limited waiting space at traffic lights (approaches are finite) we are given maximum queue lengths. The maximum queue length of queue $i \in \mathcal{N}$ is the maximum amount of traffic that could be waiting at approach i. These maximum queue lengths follow from the design of the intersection. The maximum queue length of queue $i \in \mathcal{N}$ is denoted with $x_i^{max}(t)$. Whenever a signal consists of M traffic lights and the maximum queue lengths of these traffic lights are $x_{i,j}^{max}$, $j = 1, ..., M$, we can determine the maximum queue length of signal i using:

$$x_i^{max} = \sum_{j=1}^{M} x_{i,j}^{max}. \tag{3.1}$$

Recall that in the stochastic model each traffic light has a separate buffer. Thus, the maximum queue length of traffic light j (in signal i) that is used in the stochastic model is equal to $x_{i,j}$.

For each signal we model its queue with one buffer. Thus, all the traffic arriving at signal i is stored in one buffer.

The total time that a vehicle, cyclist or pedestrian spends at the intersection, consists of:

1 A travel time to the back of the queue (whenever the queue is not empty) or to the traffic light (whenever the queue is empty). For the hybrid fluid model we assume that traffic arrives instantaneously, i.e. the travel time to the back of the queue or to the traffic light (depending on whether the queue is empty) is assumed to be equal to zero.

2 The time between the moment of arrival (at the back of the queue or at the traffic light, depending on whether the queue is empty) and the moment of departure (the moment at which this vehicle, cyclist of pedestrian crosses the stop line). During a slow mode the time between the moment of arrival and the moment of departure is equal to zero seconds.

3 A travel time (starting at the moment of departure) to leave the intersection. For a hybrid fluid model, we assume traffic departs instantaneously, i.e. the travel time to leave the intersection is assumed to be equal to zero seconds.

Hence, during a slow mode arriving traffic spends zero seconds at the intersection. The delay at the intersection is defined as the additional travel time experienced by a driver, cyclist or pedestrian at the intersection. For the hybrid fluid model the average delay at the intersection is equal to the average duration of part 2, i.e. the average time between the moment of arrival (at the back of the queue or at the traffic light depending on whether the queue is empty) and the moment of departure.

3.2.2 Modeling Arrivals

At each of the approaches, traffic (vehicles, cyclists or pedestrians) arrives with a certain intensity. We use λ_i to denote the arrival rate at signal $i \in \mathcal{N}$, i.e. the mean number of vehicles, cyclists or pedestrians arriving at approach i per second. For the hybrid fluid model we assume that the amount of traffic that arrives at signal i during an interval of size T is equal to exactly $\lambda_i T$. Thus, we assume that the arrival rate is constant over time (think of it as a constant flow of water into the buffer).

The arrival rate is often obtained by counting the number of vehicles, cyclists or pedestrians arriving at a traffic light. When signal $i \in \mathcal{N}$ consists out of more than one traffic light, the arrival rate of signal i is equal to the sum of the arrival rates of all traffic lights in this signal. When signal i consists out of M traffic lights and $\lambda_{i,j}$, $j = 1, ..., M$ are the arrival rates at these traffic lights then the arrival rate at signal i is given by:

$$\lambda_i = \sum_{j=1}^{M} \lambda_{i,j}. \tag{3.2}$$

Recall that in the stochastic model each traffic light has a separate arrival process. Thus, the arrival rate of traffic light j (in signal $i \in \mathcal{N}$) that is used in the stochastic model is equal to $\lambda_{i,j}$.

3.2.3 Modeling Departures

During the green period of a signal, traffic can depart. For the hybrid fluid model, departures can be seen as a flow of water pouring out of a buffer.

Consider the case where signal $i \in \mathcal{N}$ consists out of M traffic lights and $\lambda_{i,j}$, $j = 1, ..., M$ and $\mu_{i,j}$ are the arrival rates respectively the maximum departure rates at these traffic lights. The traffic waiting at these M traffic lights is stored in one buffer. For the hybrid fluid model, a fraction $\frac{\lambda_{i,j}}{\lambda_i}$ of the arrivals at signal i, actually arrive at traffic light j and a fraction $\frac{\lambda_{i,j}}{\lambda_i}$ of the traffic waiting at signal i is traffic that is actually waiting at traffic light j. For the hybrid fluid model, a fraction $\frac{\lambda_{i,j}}{\lambda_i}$ of the traffic that departs at this signal is traffic that departs at traffic light j. Hence, we can calculate the maximum departure rate at signal i using:

$$\mu_i = \sum_{j=1}^{M} \frac{\lambda_{i,j}}{\lambda_i} \mu_{i,j}. \tag{3.3}$$

Recall that in the stochastic model each traffic light has a separate departure process. Thus, the maximum departure rate of traffic light j (in signal $i \in \mathcal{N}$) that is used in the stochastic model is equal to $\mu_{i,j}$.

3.2.4 Modeling Mixed Arrival Flows

In this section we show how to calculate the arrival rate, maximum departure rate and maximum queue length of a traffic light with mixed traffic flows. From the arrival rates, maximum departure rates and maximum queue lengths of the traffic lights in a signal we can again calculate the arrival rate, departure rate and maximum queue length of a signal using (3.1), (3.2) and (3.3).

At a traffic light M different types of traffic arrive. We use:

- λ_i, $i = 1, ..., M$ is the arrival rate of type i traffic, i.e. how many type i units arrive per second.
- μ_i, $i = 1, ..., M$ is the maximum departure rate of type i traffic, i.e. the maximum amount of type i units that can depart per second.
- x_i^{max} is the maximum queue length in type i units, when the queue only consists out of type i traffic.

Note that in this section the index i refers to a type of traffic. In the rest of this thesis this index refers to a signal.

In this section we give the equations for the arrival rate, the maximum departure rate and the maximum queue length of a traffic light when we are given λ_i, $i = 1, ..., M$, μ_i, $i = 1, ..., M$ and x_i^{max}, $i = 1, ..., M$. These equations are explained using an example.

For traffic of type i, $i = 1, ..., M$, we can calculate the dimensionless arrival rate $\tilde{\lambda}_i$, the dimensionless maximum departure rate $\tilde{\mu}_i$, $i = 1, ..., M$ and the maximum queue length $\tilde{x}_i^{max}$ in seconds via:

$$\begin{aligned}
\tilde{\lambda}_i &= \frac{\lambda_i}{\mu_i}, \\
\tilde{\mu}_i &= \frac{\mu_i}{\mu_i} = 1, \\
\tilde{x}_i^{max} &= \frac{x_i^{max}}{\mu_i}.
\end{aligned}$$

When considering the hybrid fluid model, the dimensionless arrival rate $\tilde{\lambda}$ at the traffic light, the dimensionless maximum departure rate $\tilde{\mu}$ at the traffic light and the maximum queue length $\tilde{x}^{max}$ at the traffic light can be calculated via:

$$\begin{aligned}
\tilde{\lambda} &= \sum_{i=1}^{M} \tilde{\lambda}_i, \\
\tilde{\mu} &= 1, \\
\tilde{x}^{max} &= \sum_{i=1}^{M} \frac{\tilde{\lambda}_i}{\tilde{\lambda}} \tilde{x}_i^{max}.
\end{aligned}$$

Both $\tilde{x}^{max}$ and $\tilde{x}_i^{max}$ are real valued numbers (see also example below).

Example 3.2.1 *Consider the case where trucks, tractors and cars arrive at the same traffic light.*

The arrival rate of trucks, tractors and cars are respectively $\lambda_{truck} = 0.05$ trucks per second, $\lambda_{tractor} = 0.01$ tractors per second and $\lambda_{car} = 0.3$ cars per second. The maximum departure rates of trucks, tractors and cars are respectively $\mu_{truck} = 0.2$ truck per second, $\mu_{tractor} = 0.3$ tractors per second and $\mu_{car} = 0.5$ cars per second. Whenever the queue only consists out of trucks, the maximum queue length is $x_{truck}^{max} = 4.4$ trucks. Whenever the queue only consists out of tractors, the maximum queue length is $x_{tractor}^{max} = 7.5$ tractors and whenever the queue only consists out of cars, the maximum queue length is $x_{truck}^{max} = 12$ trucks.

The dimensionless arrival rates of trucks, tractors and cars are equal to respectively $\tilde{\lambda}_{truck} = 0.25$, $\tilde{\lambda}_{tractor} = 1/30$ and $\tilde{\lambda}_{car} = 0.6$. We can interpret these arrival rates as the seconds of work that arrive every second. For example, every second λ_{trucks} trucks arrive. When using the maximum departure rate (μ_{truck}) this amount of trucks departs in $\tilde{\lambda}_{truck} = 0.25$ seconds. Thus, every second, 0.25 seconds of work of the type 'trucks' arrives. The total arrival rate at the traffic light is $\tilde{\lambda} = 1/4+1/30+4/5 \approx 0.8833$ (seconds of work that arrive every second).

It is trivial to say that when we let traffic of type i depart at the maximum departure rate, every second one second of work departs. Thus, $\tilde{\mu}_i = 1$. As a result, independent of the type of traffic that departs, one second of work depart every second when working at the maximum departure rate. Hence, it holds that $\tilde{\mu} = 1$.

The maximum queue lengths of trucks, tractors and cars are equal to respectively $\tilde{x}^{max}_{truck} = 22$ seconds (of work), $\tilde{x}^{max}_{tractor} = 25$ seconds (of work) and $\tilde{x}^{max}_{car} = 24$ seconds (of work).

A fraction $\frac{\tilde{\lambda}_i}{\tilde{\lambda}}$ of the arriving work is of the type $i = truck, tractor, car$. Thus when the maximum queue length of the traffic light is reached, this queue contains $\frac{\tilde{\lambda}_i}{\tilde{\lambda}}\tilde{x}^{max}_i$ seconds (of work) of type i. Hence, the maximum queue length of the traffic light in seconds is equal to:

$$\sum_{i=truck,tractor,car} \frac{\tilde{\lambda}_i}{\tilde{\lambda}} \tilde{x}^{max}_i \approx 23.5 \text{ seconds.}$$

Whenever we have calculated all the arrival rates, departure rates and maximum queue lengths of the traffic lights in a signal, we can calculate the arrival rate, maximum departure rate and maximum queue length of this signal via (3.1), (3.2) and (3.3). In these equations all arrival rates, maximum departure rates and maximum queue lengths of the different traffic lights must be expressed in the same unit. For example, all arrival rates and maximum departure rates in their dimensionless form and all maximum queue lengths in seconds (of work).

3.2.5 Modeling Two-Way Pedestrian Crossings

At pedestrian crossings pedestrians generally walk in two directions (two-way pedestrian crossing). In this case at each side of the pedestrian crossing, a traffic light is positioned. In practice these two traffic lights switch to green at the same time and switch to red at the same time. Hence, these traffic lights can be partitioned in the same signal. The arrival rate at this signal is simply equal to the sum of the arrival rates at both sides of the pedestrian crossing. Further, because the pedestrian crossing has to be 'shared' by the pedestrians that walk the pedestrian crossing in both direction, it is logical to assume that the maximum departure rate μ_i at this signal (the number of pedestrians that can cross the pedestrian crossing per second) is constant.

For example, at a pedestrian crossing the maximum departure rate is equal to μ_i. This means that when at a moment the departure rate at one side is equal to $d(t) \leq \mu_i$ pedestrians per second, at that moment maximally $\mu_i - d(t)$ pedestrians can cross the pedestrian crossing from the other side.

Usually a large number of pedestrians can cross the pedestrian crossing simultaneously. Hence, pedestrian crossings usually have a large maximum departure rate. However, because pedestrians move relatively slow (in comparison to vehicles and cyclists) it takes a while before a pedestrian has crossed the pedestrian crossing. Therefore, a conflicting signal may only switch to green whenever the signal of a pedestrian crossing has been red for a relatively long time (large setup times).

Furthermore, it is fair to assume that the maximum queue lengths are infinite at a pedestrian crossing because pedestrians always find a spot to wait in front of the traffic light.

3.2.6 Modeling Startup Effect in the Maximum Departure Rate

We assume in our hybrid fluid model that the maximum departure rate is constant during a green period. However, at the beginning of a green time there is a startup effect; people do not respond instantaneously and traffic needs some time to accelerate. Hence, the maximum departure rate is not constant; it increases and after a certain amount of time the maximum departure rate does not change anymore. The transient part, where the maximum departure rate increases, is called the startup effect.

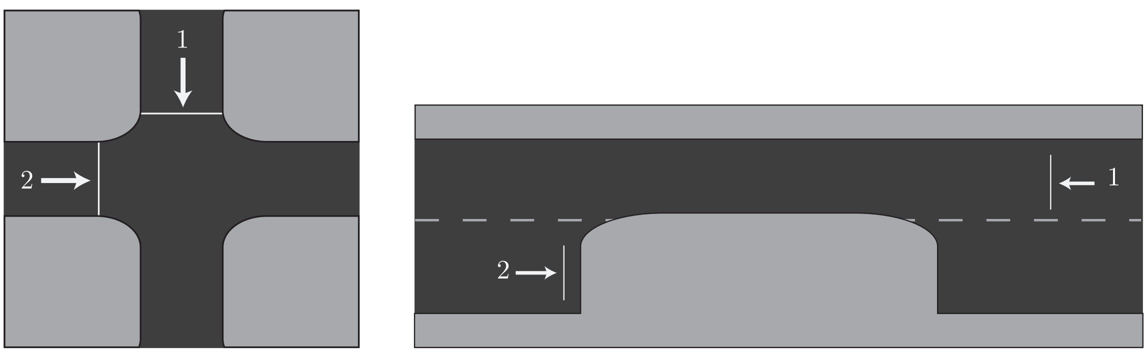
Just like for the stochastic model, we can adjust the duration of the setups to model this startup effect. When $t_{startup}$ is the duration (in seconds) of the startup effect at signal i and D is the number of vehicles, cyclists or pedestrians that depart at signal i during this startup effect then we model this startup effect by increasing the setup times $\sigma_{j,i}$ with $t_{startup} - \frac{D}{\mu_i}$ seconds for all signals j that are conflicting with signal i. During the green period, the maximum departure rate is constant and equal to μ_i. We assume that the startup effect always (every green period) has the same duration at a signal i and we assume that the startup effect takes less than the minimum green time.

In this section we introduced two models; a stochastic model and a hybrid fluid model. For both models we have shown how to model queues, arrivals and departures. Furthermore, we showed how to model a startup effect in the maximum departure rate. For the hybrid fluid model we showed that we could also model mixed arrival flows and two-way pedestrian crossings. In the next chapter we solve the trajectory optimization problem for a simple intersection with two (conflicting) signals. In that chapter we model the intersection with the hybrid fluid model. The stochastic model is used for simulation.

Chapter 4

Trajectory Optimization: A Simple Intersection of Two Signals

In this chapter we consider the trajectory optimization problem for a simple intersection of two conflicting signals. First we explain the trajectory optimization problem more explicitly in Section 4.1 and Section 4.2. Subsequently, in Section 4.3 we prove that we can always find an optimal trajectory satisfying some properties. Using these properties an optimization problem (that we solve analytically) is proposed in Section 4.8. In Figure 4.1 we present two example of an intersection with two conflicting signals: the intersection of two one-way streets a two-way street with a roadblock.



(a) An intersection of two one-way streets.

(b) Two-way street with a roadblock. Traffic lights controls which traffic flow may pass the roadblock.

Figure 4.1: Examples of two conflicting signals.

4.1 Problem Description

The problem of trajectory optimization is the process of designing a trajectory that minimizes or maximizes some measure of performance within prescribed constraint boundaries. A trajectory is a solution of a mathematical model. Just like in [24] we consider a mathematical model as an exclusion law. A mathematical model expresses the opinion that some things can happen, are possible, while other cannot, are declared impossible. These exclusion laws of a mathematical model can be expressed in the form of equations. These equations are called behavioral equations. The outcomes that a mathematical model allows, and are declared possible, are called the behavior of the mathematical model, i.e. the behavior is the solution set of the behavioral equations. A solution of the behavioral equations is called a trajectory. To solve the trajectory optimization problem we model the simple intersection with the hybrid fluid model proposed in Section 3.2. The behavioral equations of the hybrid fluid model for this simple intersection are presented in Section 4.2. A solution of these behavioral equations is called a trajectory and consists out of the evolution (as function of time) of the following variables:

- $x_i(t)$, $i = 1, 2$: the queue lengths of both signals as a function of time.
- $m_i(t)$, $i = 1, 2$: the signal state, also called the mode, of both signals as a function of time.
- $d_i(t)$, $i = 1, 2$: the departure rate of both signals as a function of time.

For the simple intersection of only two conflicting signals, we want to find a trajectory minimizing the average weighted queue length:

$$J = \limsup_{t\to\infty} \frac{1}{t} \int_0^t [f_1(x_1(s)) + f_2(x_2(s))]ds, \tag{4.1}$$

where $f_i : \mathbb{R}_+ \to \mathbb{R}_+$ is a weight function. Weight function f_i relates the queue lengths at signal i to costs. We assume that the functions f_i are strictly increasing, i.e. larger queue lengths result in higher costs. In Section 4.8 we use more specific weight functions f_i: the linear weight functions $f_1(x_1(t)) = w_1x_1(t)$ and $f_2(x_2(t)) = w_2x_2(t)$. In Section 4.1.1 we show that minimizing the linear weight function, where $w_1 = w_2$ is equivalent to minimizing the average delay of an arbitrary road user at this intersection.

4.1.1 Average Delay of A Road User At the Intersection

In this section we show that minimizing the linear weight function where $w = w_1 = w_2$, is equivalent to minimizing the average delay of an arbitrary road user at this intersection. In this section we assume that each arrival rate λ_i, $i = 1, 2$, is given in number of vehicles per second, number of cyclists per second or number of pedestrians per second and that each queue length x_i, $i = 1, 2$, is given in number of cars, number of cyclists or number of pedestrians. When different types of traffic arrive at a signal it does not hold that minimizing the linear weight function where $w = w_1 = w_2$, is equal to minimizing the average delay of an arbitrary road user at this intersection.

When $f_i(x_i) = wx_i$, $i = 1, 2$ we can write (4.1) as follows:

$$J = w(\overline{x}_1 + \overline{x}_2), \tag{4.2}$$

where

$$\overline{x}_1 = \limsup_{t\to\infty} \frac{1}{t} \int_0^t x_1(s)ds,$$
$$\overline{x}_2 = \limsup_{t\to\infty} \frac{1}{t} \int_0^t x_2(s)ds.$$

Note that $\overline{x}_i$ is the average queue length at queue $i = 1, 2$ including the road user that is departing. From Little's Law we know that:

$$\delta_i = \frac{\overline{x}_i}{\lambda_i}, i = 1, 2,$$

where, δ_i is the average delay of a road user at signal $i = 1, 2$. Hence, we can rewrite (4.2) to:

$$J = w\lambda\delta, \tag{4.3}$$

where

$$\delta = \left(\delta_1 \frac{\lambda_1}{\lambda} + \delta_2 \frac{\lambda_2}{\lambda}\right), \tag{4.4}$$
$$\lambda = \lambda_1 + \lambda_2.$$

A fraction $\frac{\lambda_1}{\lambda}$ of the road users arrives at signal 1 and a fraction $\frac{\lambda_2}{\lambda}$ of the road users arrive at signal 2. Hence, $\delta = \left(\delta_1 \frac{\lambda_1}{\lambda} + \delta_2 \frac{\lambda_2}{\lambda}\right)$ is the average delay of an arbitrary road user at the intersection. Note, that the optimal trajectory does not change when multiplying the objective function (cost function) with $\frac{1}{w\lambda} > 0$. Hence, minimizing the linear weight function, where $w = w_1 = w_2$ results in the same optimal trajectory as minimizing the average delay of an arbitrary road user.

4.2 Behavioral Equations of the Hybrid Fluid Model

In this section we give the behavioral equations of the hybrid fluid model. First, we introduce the variables that we use in these behavioral equations in Section 4.2.1. In Section 4.2.2 we give the behavioral equations of the hybrid fluid model.

4.2.1 Manifest Variables and Latent Variables

In this section we give the manifest variables and the latent variables that we use in the behavioral equations. The manifest variables are the variables that we are interested in. A trajectory consists out of the evolution (as function of time) of these manifest variables. The latent variables are used so that we can give the behavioral equations in a compact and readable form.

We use the following manifest variables:

- $x_i(t) \in \mathbb{R}^+$, $i = 1, 2$: the queue length of queue i as a function of time. The function $x_i(t)$, $i = 1, 2$ is right-continuous.
- $m_i(t) \in \{ⓘ, ❶\}$, $i = 1, 2$: the signal state of signal i as a function of time. The function $m_i(t)$, $i = 1, 2$ is right-continuous.
- $d_i(t) \in \mathbb{R}^+$, $i = 1, 2$: the departure rate at signal i as a function of time. The function $d_i(t)$, $i = 1, 2$ is measurable.

Further, we use the following latent variables:

- $L_\tau^i(t) \in \mathbb{R}^+$, $i = 1, 2$: the time that has elapsed since the last change in the signal state of signal i. The function $L_\tau^i(t)$, $i = 1, 2$ is right-continuous.

4.2.2 Behavioral Equations

In this section we give the behavioral equations of a simple intersection with two signals.

The change in the queue length is equal to the net inflow (arrival rate minus departure rate):

$$\dot{x}_i(t) = \lambda_i - d_i(t), \quad i = 1, 2. \tag{4.5a}$$

The latent variable $L^i_\tau(t)$, $i = 1, 2$ denotes a time. Hence, its derivative with respect to time is equal to one:

$$\dot{L}^i_\tau(t) = 1, \quad i = 1, 2. \tag{4.5b}$$

The time that has elapsed since the last change in the signal state is set to zero when the signal state changes:

$$L^i_\tau(t) = 0 \quad \text{if } m_i(t^-) \neq m_i(t), \quad i = 1, 2, \tag{4.5c}$$

where

$$m_i(t^-) = \lim_{y \uparrow t} m_i(y)$$

When the signal state of signal $i = 1, 2$ changes, it holds that $m_i(t^-)$ (the left limit) is not equal to $m_i(t)$ (the right limit) (see Figure 4.2).

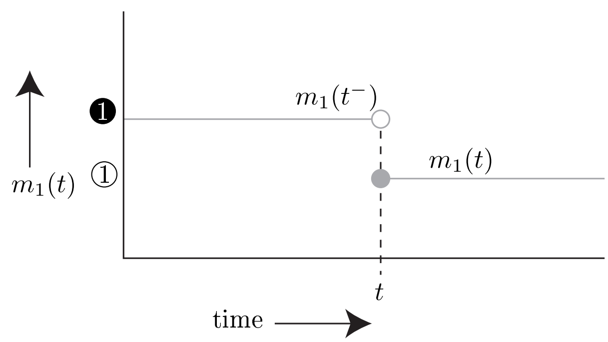


Figure 4.2: At time t signal 1 switches from red to green.

Whenever a signal is red, traffic from the corresponding queue cannot cross the intersection:

$$d_i(t) = 0 \quad \text{if } m_i(t) = \text{❶}, \quad i = 1, 2, \quad \forall t \in \mathbb{R}^+. \tag{4.5d}$$

When there is no traffic waiting at queue $i = 1, 2$, traffic can depart at a rate that is smaller than or equal to the arrival rate λ_i (otherwise it would result in a negative queue length $x_i(t)$):

$$d_i(t) \leq \lambda_i \quad \text{if } x_i(t) = 0, \quad i = 1, 2, \quad \forall t \in \mathbb{R}^+. \tag{4.5e}$$

Traffic cannot depart at a rate that exceeds the maximum departure rate:

$$d_i(t) \leq \mu_i, \quad i = 1, 2, \quad \forall t \in \mathbb{R}^+. \tag{4.5f}$$

Since signal 1 and signal 2 are conflicting, both signals cannot be green at the same time:

$$m_1(t) = \text{❶} \quad \text{if } m_2(t) = \text{②}, \tag{4.5g}$$

$$m_2(t) = \text{❷} \quad \text{if } m_1(t) = \text{①}. \tag{4.5h}$$

Signal 1 cannot switch to green whenever signal 2 switched to red less than $\sigma_{2,1}$ seconds ago. In the same way, signal 2 cannot switch to green whenever signal 1 switched to red less than $\sigma_{1,2}$ seconds ago:

$$m_1(t) = \text{❶} \quad \text{if } m_2(t) = \text{❷} \wedge L_\tau^2(t) < \sigma_{2,1}, \tag{4.5i}$$

$$m_2(t) = \text{❷} \quad \text{if } m_1(t) = \text{❶} \wedge L_\tau^1(t) < \sigma_{1,2}. \tag{4.5j}$$

The maximum queue length cannot be exceeded.

$$x_i(t) \leq x_i^{max}, \quad i = 1, 2. \tag{4.5k}$$

The duration of a green period must be at least the minimum green time and cannot exceed the maximum green time:

$$m_i(t) = \text{ⓘ} \quad \text{if } T_\tau^i(t) < g_i^{min} \wedge m_i(t^-) = \text{ⓘ}, \quad i = 1, 2, \tag{4.5l}$$

$$m_i(t) = \text{🛈} \quad \text{if } T_\tau^i(t) \geq g_i^{max} \wedge m_i(t^-) = \text{ⓘ}, \quad i = 1, 2. \tag{4.5m}$$

A solution of these behavioral equations (the manifest variables as function of time) is called a trajectory. Note that we allow every initial condition as long as it satisfies (4.5).

4.2.3 Assumptions

In this section we give the assumptions used in this chapter. We assume that the arrival rate and the maximum departure rate of a signal is positive:

$$\lambda_1, \lambda_2, \mu_1, \mu_2 > 0. \tag{4.6a}$$

We assume that the setup times are both non-negative and that one of the setup times is strictly positive:

$$\sigma_{1,2}, \sigma_{2,1} \geq 0, \tag{4.6b}$$

$$\sigma_{1,2} + \sigma_{2,1} > 0. \tag{4.6c}$$

We assume that the minimum green times are non-negative:

$$g_1^{min}, g_2^{min} \geq 0. \tag{4.6d}$$

We assume that the average green times converge. Thus, we assume that the following limits exist:

$$\bar{g}_1 = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_1^k}{M}, \tag{4.6e}$$

$$\bar{g}_2 = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_2^k}{M}. \tag{4.6f}$$

$$\tag{4.6g}$$

Further, we assume that the maximum green times and the maximum queue lengths satisfy the following inequalities:

$$\min\{g_1^{max}, \frac{x_2^{max}}{\lambda_2} - \sigma_{1,2,1}\} \geq \frac{\rho_1\sigma_{1,2,1}}{1-\rho_1-\rho_2}, \tag{4.6h}$$

$$\min\{g_2^{max}, \frac{x_1^{max}}{\lambda_1} - \sigma_{1,2,1}\} \geq \frac{\rho_2\sigma_{1,2,1}}{1-\rho_1-\rho_2}, \tag{4.6i}$$

$$\min\{g_1^{max}, \frac{x_2^{max}}{\lambda_2} - \sigma_{1,2,1}\} \geq g_1^{min}, \tag{4.6j}$$

$$\min\{g_2^{max}, \frac{x_1^{max}}{\lambda_1} - \sigma_{1,2,1}\} \geq \frac{\rho_2}{1-\rho_2}(\sigma_{1,2,1} + g_1^{min}), \tag{4.6k}$$

$$\min\{g_2^{max}, \frac{x_1^{max}}{\lambda_1} - \sigma_{1,2,1}\} \geq g_2^{min}, \tag{4.6l}$$

$$\min\{g_1^{max}, \frac{x_2^{max}}{\lambda_2} - \sigma_{1,2,1}\} \geq \frac{\rho_1}{1-\rho_1}(\sigma_{1,2,1} + g_2^{min}). \tag{4.6m}$$

In Section 4.8 we show that we can always (and only) find a trajectory resulting in a finite average weighted queue length calculated with (4.1) whenever the inequalities in (4.6h)–(4.6m) hold.

4.3 Properties of Optimal Trajectories

In Chapter 5 of van Eekelen [10] we can find some lemmas on optimal trajectories, which van Eekelen proves by a proof of contradiction. In that chapter no restrictions on green times where considered, i.e. (4.5l) and (4.5m) are not in the behavior. When including these restrictions on green times the lemmas from [10] are not valid anymore. Therefore, we have developed new lemmas to include these restrictions on green times (minimum green times and maximum green times). Furthermore, with these new lemmas we avoided a circular argument found in [10]. The results are lemma 4.1– 4.3. Both Lemma 4.1 and Lemma 4.2 hold in general (for any intersection).

Lemma 4.1 *Without loss of generality it can be assumed that for an optimal trajectory in the behavior, during a green period a signal always uses the highest possible departure rate, after which it might idle, i.e. use a departure rate equal to zero. This highest possible departure rate equals μ_i when the queue is not empty ($x_i(t) > 0$), and equals the arrival rate λ_i otherwise.*

Proof. Suppose that we are given a green time of $g_i^k = t_f - t_0$, $k \geq 1$ that satisfies $g_i^{min} \leq g_i^k \leq g_i^{max}$ and a trajectory is given, for which at the beginning of g_i^k, the queue length of queue i equals x_i^0 and at the end of g_i^k the queue length equals x_i^f. Then one can consider the alternative trajectory which only differs from the original trajectory during g_i^k. This alternative trajectory serves signal i equally long and first lets traffic depart at the highest possible rate, i.e. at the maximum departure rate when the queue is not empty($x_i(t) > 0$) and the arrival rate otherwise. In the end, this alternative signal idles, i.e. we use a departure rate equal to zero, to make sure that at the end of g_i^k the queue length equals x_i^f (see Figure 4.3). Clearly, the alternative trajectory satisfies (4.5d)–(4.5m) whenever the original trajectory does. Further, during the green period the queue length cannot decrease faster in the beginning and cannot increase faster in the end. Therefore, for the alternative signal the queue length of type i is smaller or equal at every time instants. Further, the evolution of the queue lengths of the other queues remain the same for both trajectories and f_i is strictly increasing. Hence, the alternative trajectory works at least as good, i.e. the costs (calculated with (4.1)) of the alternative trajectory are not bigger than the costs of the original trajectory.

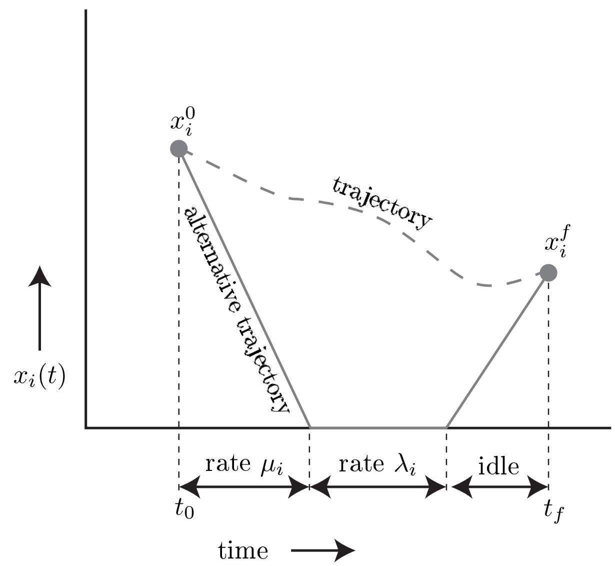


Figure 4.3: Graphical representation of Lemma 4.1.

Thus, whenever we are given a trajectory that does not satisfy the property given in this lemma, we can always find an alternative trajectory that does satisfy this property and that works at least as good. Hence, there must be an optimal trajectory that satisfies the property given in this lemma. ■

Lemma 4.2 *Without loss of generality it can be assumed that for an optimal trajectory in the behavior, a signal never idles during its green period.*

Proof. Suppose that an optimal trajectory would idle during a green period of signal i. Given the result in Lemma 4.1 this is at the end of this green period. Lets consider a trajectory that starts idling at t_0^i and finishes idling at t_f^i (at t_f^i the signal switches to red). We can find an alternative trajectory that does not idle and that works at least as good as the original trajectory, i.e. it results in costs J (calculated with (4.1)) that are not bigger than the costs of the original signal. Consider the following alternative trajectory which only differs only from the original trajectory after time t_0^i. During the interval $[t_0^i, t_f^i]$, we use $d_i(t) = \lambda_i$ instead of $d_i(t) = 0$. (see Figure 4.4). Hereafter, during each green period we let traffic depart at arrival rate λ_i until the time t_{cross}. At t_{cross} both trajectories result in the same queue length (see Figure 4.4). After the time t_{cross} both trajectories are exactly the same. Thus, at signal i the original trajectory only differs from the original trajectory in the interval $[t_0^i, t_{cross}]$.

Further, the evolution of the queue lengths of the other queues remain the same for both trajectories and f_i is strictly increasing. Hence, the alternative trajectory works at least as good, i.e. the costs (calculated with (4.1)) of the alternative trajectory are not bigger.

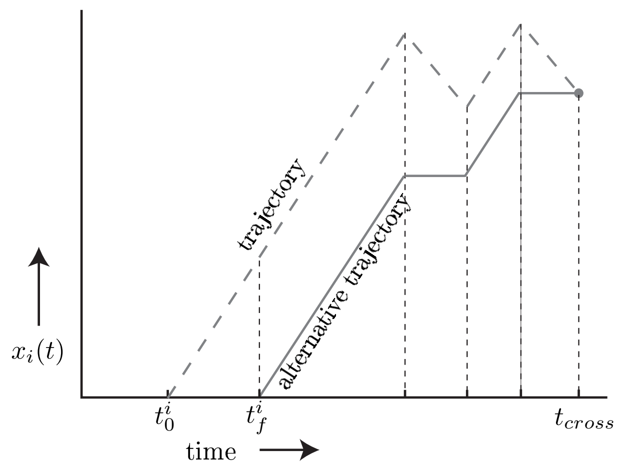


Figure 4.4: Graphical representation of Lemma 4.2.

Thus, whenever we are given a trajectory that does not satisfy the property given in this lemma, we can always find an alternative trajectory that does satisfy this property and that works at least as good. Hence, there must be an optimal trajectory that satisfies the property given in this lemma. ■

When we combine the results of Lemma 4.1 and Lemma 4.2 we see that (as expected) we always use the highest possible departure rate during a green period of signal $i = 1, 2$:

$$d_i(t) = \begin{cases} \mu_i & \text{if } x_i(t) > 0, \\ \lambda_i & \text{if } x_i(t) = 0. \end{cases}$$

It would have been a surprising yet interesting result, if using a lower departure rate can have a positive effect on the costs. If this turned out to be true, we could think about ways to control the departure rate of traffic at a traffic light.

Lemma 4.3 *Without loss of generality it can be assumed that for an optimal trajectory in the behavior, a queue is always emptied during its green period and green periods always take equally long, i.e.* $g_i^k = g_i^{k+1}$, $\forall k \geq 1$.

Proof. Lets consider a trajectory defined on the time interval $[0, \infty)$ where a queue is not emptied at least once or where the duration of the green periods is not always the same for a signal. Lets call this trajectory the 'original trajectory'. In Figure 4.5a we can see an example of the original trajectory.

We introduce the following notation for the average duration of g_i^k, r_i^k, $g_i^{\lambda,k}$ and $g_i^{\mu,k}$:

$$\bar{g}_i = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_i^k}{M}, \quad i = 1,2, \tag{4.7a}$$

$$\bar{r}_i = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{r_i^k}{M}, \quad i = 1,2, \tag{4.7b}$$

$$\bar{g}_i^\lambda = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_i^{\lambda,k}}{M}, i = 1,2, \tag{4.7c}$$

$$\bar{g}_i^\mu = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_i^{\mu,k}}{M}, i = 1,2. \tag{4.7d}$$

We can propose an alternative trajectory where a queue is always emptied during a green period and where the green times of a signal are always the same (see Figure 4.5b). For this alternative trajectory we take the green times and red times of signal i equal to respectively $\bar{g}_i$ and $\bar{r}_i$. We serve signal 1 during the red period of signal 2 and we serve signal 2 during the red period of signal 1.

We can prove that the costs J related to this alternative trajectory are not greater than the costs related to the original trajectory.

First we prove that the limits in (4.7) exist. We assume that the limits $\lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_i^k}{M}$, $i = 1,2$ exist (see Section 4.2.3). Note that whenever $\bar{g}_1$ and $\bar{g}_2$ both exist then $\bar{r}_1$ and $\bar{r}_2$ also exist because the average red period of a signal is related to the average green period of the other signal according to:

$$\bar{r}_1 = \bar{g}_2 + \sigma_{1,2,1},$$
$$\bar{r}_2 = \bar{g}_1 + \sigma_{1,2,1}.$$

Whenever signal $i = 1,2$ satisfies $\lambda_i \bar{r}_i \neq (\mu_i - \lambda_i)\bar{g}_i^\mu$ for the original trajectory, this means that the queue length of queue i would go to ∞ or $-\infty$ because:

$$\lim_{t\to\infty} x_i(t) = \lim_{M\to\infty} \sum_{k=1}^{M} (\lambda_i r_i^k - (\mu_i - \lambda_i) g_i^{\mu,k}) = \lim_{M\to\infty} M(\lambda_i \bar{r}_i - (\mu_i - \lambda_i)\bar{g}_i^\mu).$$

Note that we have used that each green time of signal i is finite. A queue length must be a non-negative number and therefore a trajectory where a queue length goes to $-\infty$ is not feasible. Further, whenever a queue length goes to ∞, the costs calculated with (4.1) are infinite. Hence, it must hold that:

$$\lambda_i \bar{r}_i = (\mu_i - \lambda_i)\bar{g}_i^\mu, i = 1,2. \tag{4.8}$$

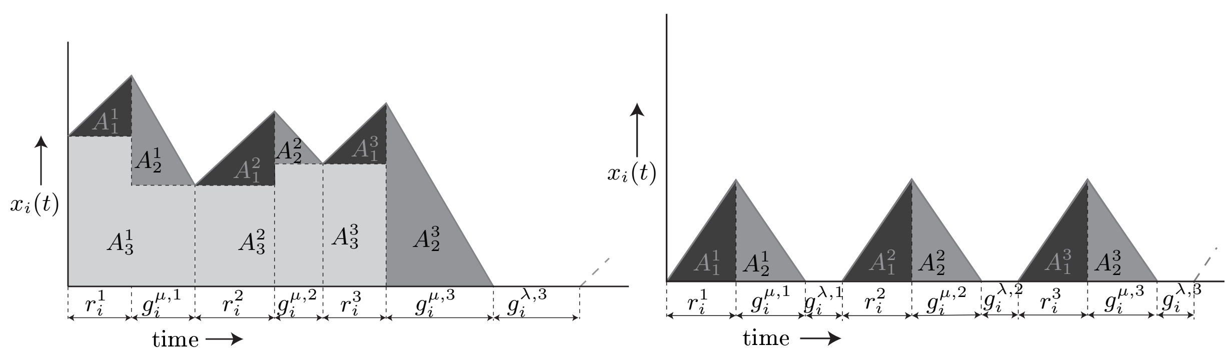
Thus, the amount of traffic that arrives during a red period of signal $i = 1,2$ is equal to $\lambda_i \bar{r}_i$ and we can let this amount of traffic depart during a period equal to exactly $\bar{g}_i^\mu$. As a result, from $\bar{g}_i = \bar{g}_i^\mu + \bar{g}_i^\lambda$ we can obtain that for the alternative policy the length of the slow mode is equal to $\bar{g}_i^\lambda$ during each green period. From (4.8) we can see that $\bar{g}_i^\mu$ exists (because $\bar{r}_i$ exists) and from $\bar{g}_i = \bar{g}_i^\mu + \bar{g}_i^\lambda$ we know that $\bar{g}_i^\lambda$ exists. Hence, $\bar{g}_i$, $\bar{r}_i$, $\bar{g}_i^\lambda$ and $\bar{g}_i^\mu$ all exist.

Also note that the alternative trajectory is always feasible. First of all, the green periods of the alternative trajectory (with duration $\bar{g}_i$) always take longer than the shortest green period of the original trajectory. Second of all, the green periods of the alternative trajectory (with duration $\bar{g}_i$) always take shorter than the longest green period of the original trajectory. Furthermore, the maximum queue length are less for the alternative trajectory than for original trajectory. As a result, whenever the original trajectory satisfies (4.5d)–(4.5m), the alternative trajectory does as well.

Now we are going to prove that the costs related to the alternative trajectory are not bigger than the costs related to the original trajectory. We use $b_{g_i^{\mu,k}}$, $k \geq 1$ and $b_{r_i^k}$, $k \geq 1$ for the time at which the green period g_i^k starts respectively the time at which the red period r_i^k starts. Further, we use $e_{g_i^{\mu,k}}$, $k \geq 1$ for the time at which queue i is emptied during g_i^k and we use $e_{r_i^k}$, $k \geq 1$ for the time at which r_i^k ends. We distinguish three different areas (see Figure 4.5): A_1^k, $k \geq 1$, A_2^k, $k \geq 1$ and A_3^k, $k \geq 1$.

$$
\begin{aligned}
A_1^k &= \int_{b_{g_i^{\mu,k}}}^{e_{g_i^{\mu,k}}} (x_i(t) - x_i(b_{g_i^{\mu,k}}))dt, && k \geq 1, \\
A_2^k &= \int_{b_{r_i^k}}^{e_{r_i^k}} (x_i(t) - x_i(e_{r_i^k}))dt, && k \geq 1, \\
A_3^k &= x_i(b_{g_i^{\mu,k}})(e_{g_i^{\mu,k}} - b_{g_i^{\mu,k}}) + x_i(e_{r_i^k})(e_{r_i^k} - b_{r_i^k}), && k \geq 1.
\end{aligned}
$$

In Figure 4.5, A_1^k is visualized in dark gray, A_2^k is visualized in medium gray and A_3^k is visualized in light gray.



(a) Queue length of signal i for an example of the original trajectory.
(b) Queue length of signal i for the alternative trajectory.

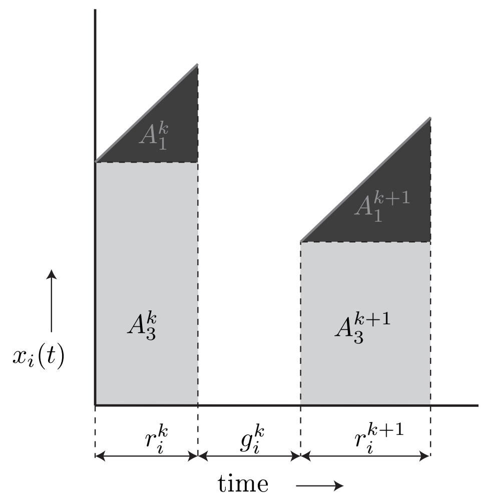
Figure 4.5: Visualization of the original trajectory and the alternative trajectory.

Because the queues are always emptied for the alternative trajectory, it holds that $A_3^k = 0$, $k \geq 1$ for this trajectory.

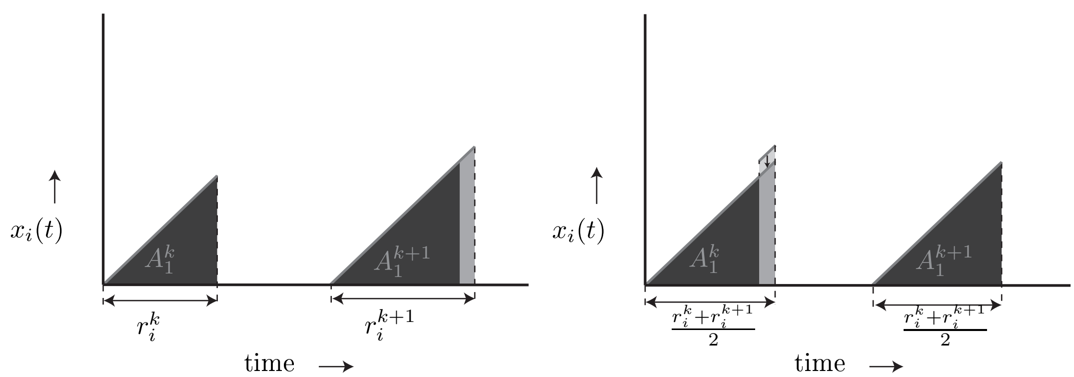
Now we are going to prove that the costs related to signal i and made during only the red periods are not bigger for the alternative trajectory than for original trajectory. Thus, we only consider the signal during the red periods of signal $i = 1, 2$, i.e. we cut out the parts where signal i is green (see Figure 4.6a).

Now we can shift each and every red period towards the $x_i(t) = 0$-axis for the original trajectory, i.e. removing the areas A_3^k. Since f_i is strictly increasing, shifting the red periods of the original trajectory towards the time axis cannot increase the costs related to the red periods of signal i.

On the left side of Figure 4.6b we can see A_1^k and A_1^{k+1} plotted for the shifted original trajectory. Without loss of generality we can assume that the first red period r_i^k is longer than the second red period r_i^{k+1} for two adjacent red periods. When we take both green times equal to $\frac{r_i^k+r_i^{k+1}}{2}$ we get the areas A_1^k and A_1^{k+1} as can be seen on the right side of Figure 4.6b. We can see that the dark gray areas are the same and that the medium gray areas differ (the difference is the light gray area). Since f_i is strictly increasing, taking the red time of two adjacent red periods equal to each other cannot increase the costs related to the red periods of signal i. Hence, taking all red periods equal to each other cannot increase the costs related to the red periods of signal i. Note, that the costs, of this shifted trajectory where all red periods are equal to each other, are exactly the costs made during the red periods of the alternative trajectory. Thus, the costs related to the red periods of the alternative trajectory cannot be bigger than the costs related to the red periods of the original trajectory.



(a) Visualization of only the red periods of the original trajectory.



(b) Left: visualization of the shifted red periods of the original trajectory, right: 2 equal red periods instead of 2 unequal red periods.

Figure 4.6: Comparing the costs made during the red periods for both trajectories.

In exactly the same way we can prove that the costs related to the green periods of signal $i = 1, 2$ cannot be bigger for the alternative trajectory than for the original trajectory. Hence, the costs of the alternative trajectory are not bigger than the costs of the original trajectory.

Thus, whenever we are given a trajectory that does not satisfy the property given in this lemma, we can always find an alternative trajectory that does satisfy this property and that works at least as good. Hence, there must be an optimal trajectory that satisfies the property given in this lemma. ■

4.4 Shape of the Periodic Optimal Trajectory

Using lemmas 4.1–4.3 we can find the following corollary for the simple intersection of two signals.

Corollary 4.4 *For the simple intersection of two signals there is always an optimal trajectory (minimizing (4.1)) that has the periodic shape shown in Figure 4.7, which consists out of the following phases (these phases repeat periodically):*

phase 1 *Signal 1 is green and* $d_1(t) = \mu_1$ *until queue 1 is empty.*

phase 2 *Signal 1 is green and* $d_1(t) = \lambda_1$.

phase 3 *perform a setup to signal 2, i.e. switch signal 1 to red and keep both signals red for a period equal to* $\sigma_{1,2}$

phase 4 *Signal 2 is green and* $d_2(t) = \mu_2$ *until queue 2 is empty.*

phase 5 *Signal 2 is green and* $d_2(t) = \lambda_2$.

phase 6 *perform a setup to signal 1, i.e. switch signal 2 to red and keep both signals red for a period equal to* $\sigma_{2,1}$

Because all green periods of a signal have the same duration and all red periods of a signal have the same duration, we use:

$$\begin{aligned} g_i &= g_i^k, \quad i = 1, 2, k \geq 1, \\ r_i &= r_i^k, \quad i = 1, 2, k \geq 1, \\ g_i^\lambda &= g_i^{\lambda,k}, i = 1, 2, k \geq 1, \\ g_i^\mu &= g_i^{\mu,k}, i = 1, 2, k \geq 1. \end{aligned}$$

Phase 2 and phase 4 are the so called slow modes and may have a duration equal to zero. We call this periodically repeated sequence of 6 phases a cycle. On the left hand side of Figure 4.7, this cycle is plotted in the (x_1, x_2)*-plane. The right hand side graphs shows the queue lengths over time, with the slopes annotated to them. The duration of a cycle is denoted with* c *and is equal to* $g_1 + g_2 + \sigma_{1,2,1}$.

A slow mode can reduce the cost function because it increases the cycle duration c and as a consequence the system switches less, i.e. there are less setups.

The traffic that arrives during a red period of signal 1 can (precisely) depart during g_1^μ. In the same way, the traffic that arrives during a red period of signal 2 can (precisely) depart during g_2^μ. Hence we can find:

$$g_1^\mu = \frac{\rho_1}{1 - \rho_1}(g_2 + \sigma_{1,2,1}), \tag{4.9a}$$

$$g_2^\mu = \frac{\rho_2}{1 - \rho_2}(g_1 + \sigma_{1,2,1}). \tag{4.9b}$$

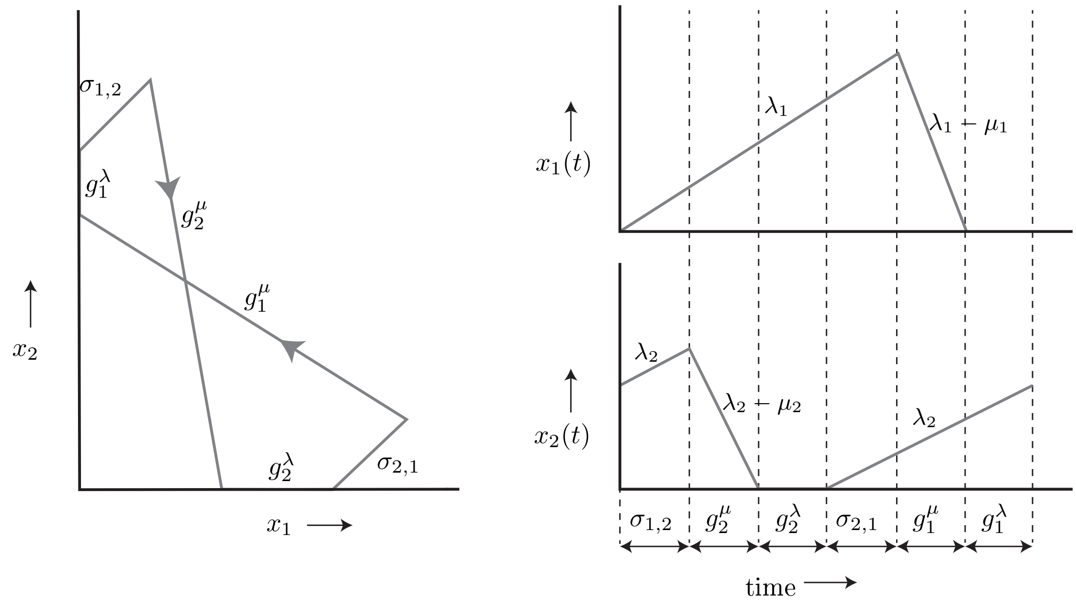


Figure 4.7: Shape of the periodic optimal trajectory. Left: periodic orbit. Right: queue lengths time.

We use $x_1^\sharp$ for the queue length of queue 1 when the green period of signal 2 is ended and we use $x_2^\sharp$ for the queue length of queue 2 when the green period of signal 1 ended:

$$x_1^\sharp = (g_2\sigma_{1,2})\lambda_1 \tag{4.10a}$$

$$x_2^\sharp = (g_1\sigma_{2,1})\lambda_2 \tag{4.10b}$$

We call the shape (consisting of phases 1 until 6) shown on the left hand side of Figure 4.7 the truncated bow tie curve. Whenever $g_1^\lambda = g_2^\lambda = 0$ we call this shape the pure bow tie curve (consisting of only phases 1, 3, 5 and 6). The pure bow tie curve is the curve with the shape shown in Figure 4.7 that has the smallest possible cycle duration c. When $g_1^\lambda = g_2^\lambda = 0$ the green times g_1 and g_2 are precisely large enough to let the amount of traffic depart that arrives during a red period. Thus for the pure bow tie curve it holds that:

$$g_1(\mu_1 - \lambda_1) = r_1\lambda_1 = (g_2 + \sigma_{1,2,1})\lambda_1, \tag{4.11a}$$

$$g_2(\mu_2 - \lambda_2) = r_2\lambda_2 = (g_1 + \sigma_{1,2,1})\lambda_2. \tag{4.11b}$$

From (4.11) we can obtain that for the pure bow tie curve it holds that:

$$c = \frac{\sigma_{1,2,1}}{1 - \rho_1 - \rho_2},$$
$$g_1 = g_1^{\mu} = \rho_1 \frac{\sigma_{1,2,1}}{1 - \rho_1 - \rho_2},$$
$$g_2 = g_2^{\mu} = \rho_2 \frac{\sigma_{1,2,1}}{1 - \rho_1 - \rho_2}.$$

The pure bow tie curve is shown in Figure 4.8.

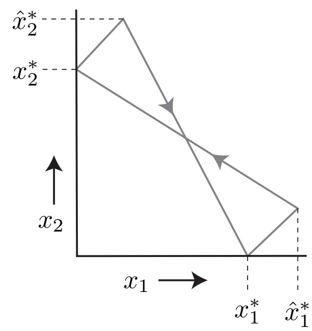


Figure 4.8: Pure bow tie curve.

The following expressions can be found for the coordinates of the pure bow tie curve:

$$x_1^* = \lambda_1 \left(\sigma_{1,2} + \frac{\rho_2 \sigma_{1,2,1}}{1 - \rho_1 - \rho_2} \right), \tag{4.12a}$$

$$\hat{x}_1^* = \lambda_1 \sigma_{1,2,1} \left(\frac{1 - \rho_1}{1 - \rho_1 - \rho_2} \right), \tag{4.12b}$$

$$x_2^* = \lambda_2 \left(\sigma_{2,1} + \frac{\rho_1 \sigma_{1,2,1}}{1 - \rho_1 - \rho_2} \right), \tag{4.12c}$$

$$\hat{x}_2^* = \lambda_2 \sigma_{1,2,1} \left(\frac{1 - \rho_2}{1 - \rho_1 - \rho_2} \right). \tag{4.12d}$$

4.5 An Optimal Trajectory Discarding Restrictions on Maximum Queue Lengths, Minimum Green Times and Maximum Green Times

In the rest of this chapter we consider a more specific form of the cost function J: the linear cost function presented in (4.13).

$$J = \frac{1}{c} \int_0^c [w_1 x_1(s) + w_2 x_2(s)] ds, \tag{4.13}$$

where, $w_1, w_2 > 0$.

In this section, we discard the restrictions on maximum queue lengths, the restrictions on minimum and the restrictions maximum green times. Thus, we discard (4.5k)–(4.5m) of the behavioral equations given in Section 4.2.2, i.e. the maximum queue lengths are infinite, the minimum green times are equal to zero and the maximum green times are infinite.

In [10], van Eekelen has proven that for this situation Theorem 4.1 holds. Without loss of generality he assumes that $w_1\lambda_1 \geq w_2\lambda_2$.

Theorem 4.1 *For a simple intersection of two signals the periodic optimal trajectory with respect to linear costs on queue lengths (4.13), has a slow mode for at most one signal (signal 1). The slow mode occurs if and only if* $w_1\lambda_1(\rho_1+\rho_2) - (w_1\lambda_1 - w_2\lambda_2)(1-\rho_2) < 0$

Proof. See appendix A.2 of [10] ■

Hence, when discarding behavioral equations (4.5k)–(4.5m) and we assume w.l.o.g. that $w_1\lambda_1 \geq w_2\lambda_2$, the optimal steady state cycle has the shape shown in Figure 4.9.

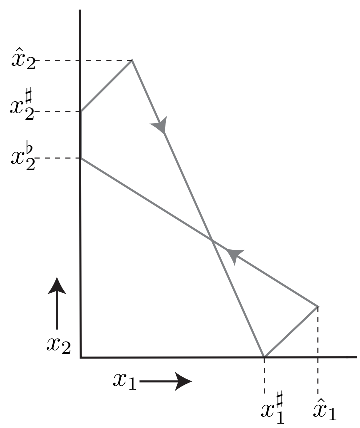


Figure 4.9: Optimal periodic trajectory when discarding behavioral equations (4.5k)–(4.5m). We assume w.l.o.g. that $w_1\lambda_1 \geq w_2\lambda_2$.

The coordinates of this optimal shape are presented below. Here it is used that the duration of the slow mode g_1^λ is equal to $\alpha_1\sigma_{1,2,1}$.

$$x_1^\sharp = \lambda_1\left(\sigma_{1,2} + \frac{\sigma_{1,2,1}\rho_2(1+\alpha_1(1-\rho_1))}{1-\rho_1-\rho_2}\right), \tag{4.14a}$$

$$\hat{x}_1 = \lambda_1\sigma_{1,2,1}\left(\frac{(1+\alpha_1\rho_2)(1-\rho_1))}{1-\rho_1-\rho_2}\right), \tag{4.14b}$$

$$x_2^\flat = \lambda_2\sigma_{1,2,1}\left(\frac{(1-\rho_2)(1+\alpha_1(1-\rho_1))}{1-\rho_1-\rho_2}\right), \tag{4.14c}$$

$$x_2^\sharp = \lambda_2\left(\sigma_{2,1} + \frac{\sigma_{1,2,1}(\alpha_1(1-\rho_1)(1-\rho_2)+\rho_1)}{1-\rho_1-\rho_2}\right), \tag{4.14d}$$

$$\hat{x}_2 = \lambda_2\sigma_{1,2,1}\left(\frac{(1-\rho_2)(1+\alpha_1(1-\rho_1))}{1-\rho_1-\rho_2}\right). \tag{4.14e}$$

In [10] we can find that α_1 equals:

$$\alpha_1 = \begin{cases} 0 & \text{if } w_1\lambda_1(\rho_1+\rho_2) - (w_1\lambda_1 - w_2\lambda_2)(1-\rho_2) \geq 0, \\ \text{positive root of (4.15)} & \text{otherwise.} \end{cases}.$$

$$[w_1\lambda_1\rho_2^2 + w_2\lambda_2(1-\rho_1)^2(1-\rho_2)]\alpha_1^2 + 2[w_1\lambda_1\rho_2^2 + w_2\lambda_2(1-\rho_1)(1-\rho_2)]\alpha_1 + [w_1\lambda_1(\rho_1+\rho_2) - (w_1\lambda_1 - w_2\lambda_2)(1-\rho_2)] = 0. \tag{4.15}$$

4.6 An Optimal Trajectory Discarding Restrictions on Minimum Green Times and Maximum Green Times

In this section we (only) discard behavioral equation (4.5l) and (4.5m) given in Section 4.2.2, i.e. the minimum green times are equal to zero and the maximum green times are infinite. In section 5.4 of [10], van Eekelen presents the effects of finite maximum queue lengths on the periodic optimal trajectory. In this section he again assumes w.l.o.g. that $w_1\lambda_1 \geq w_2\lambda_2$. We recapitulate his results quickly.

A trajectory can only be found whenever $x_i^{max} \geq \hat{x}_i^*, i = 1, 2$. With $\hat{x}_i^*$, $i = 1, 2$ as in (4.12). In the left hand side of Figure 4.10 the periodic optimal trajectory is shown for a finite maximum queue length of signal 1. In the right hand side of Figure 4.10 the periodic optimal trajectory is shown for a finite maximum queue length of signal 2.

The coordinates of the periodic optimal trajectory with queue length constraints are denoted with bars ($^-$). The original (unconstrained) periodic optimal trajectory is shown in light gray.

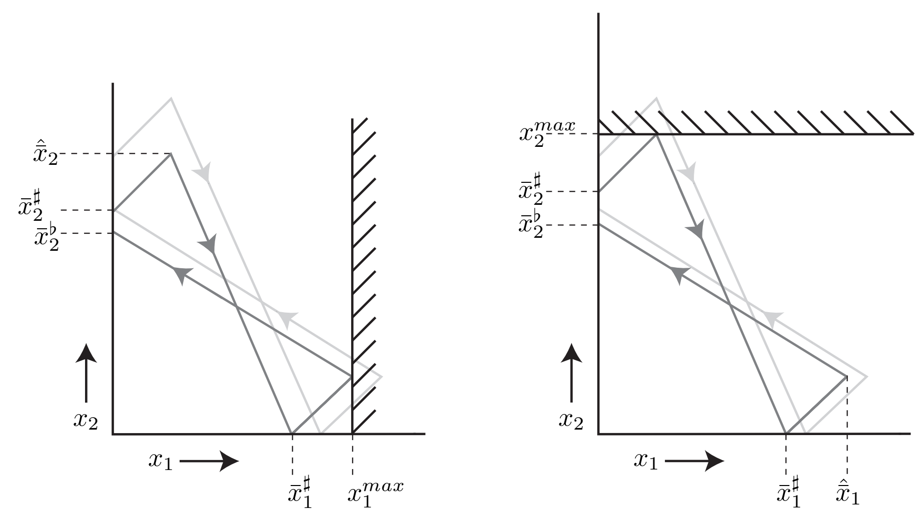


Figure 4.10: New periodic optimal trajectory, due to queue length constraints. the original unconstrained periodic optimal trajectory is visualized in light gray and the constrained optimal trajectory in dark gray.

Van Eekelen derived the following coordinates of the periodic optimal trajectory with finite queue lengths.

$$\begin{array}{lll}
\bar{x}_1^{\sharp} = \min\{x_1^{\sharp}, & x_1^{max} - \lambda_1\sigma_{2,1}, & \lambda_1\left(\sigma_{1,2} + \frac{x_2^{max}}{\mu_2-\lambda_2}\right)\}, \\
\hat{\bar{x}}_1 = \min\{\hat{x}_1, & x_1^{max}, & \lambda_1\left(\sigma_{1,2,1} + \frac{x_2^{max}}{\mu_2-\lambda_2}\right)\}, \\
\bar{x}_2^{\flat} = \min\{x_2^{\flat}, & x_1^{max}, & \lambda_1\left(\sigma_{1,2,1} + \frac{x_2^{max}}{\mu_2-\lambda_2}\right)\}, \\
\bar{x}_2^{\sharp} = \min\{x_2^{\sharp}, & \frac{\mu_2-\lambda_2}{\lambda_1}(x_1^{max} - \lambda_1\sigma_{1,2,1}) - \lambda_2\sigma_{1,2}, & x_2^{max} - \lambda_2\sigma_{1,2}\}, \\
\hat{\bar{x}}_2 = \min\{\hat{x}_2, & \frac{\mu_2-\lambda_2}{\lambda_1}(x_1^{max} - \lambda_1\sigma_{1,2,1}), & x_2^{max}\}.
\end{array}$$

With the expressions for $x_1^{\sharp}$, $x_2^{\sharp}$, $x_1^{\flat}$, $x_2^{\flat}$ and $\hat{x}_1$ as in (4.14).

4.7 An Optimal Trajectory Discarding Restrictions on Minimum Green times

In this section we (only) discard behavioral equation (4.5l) given in Section 4.2.2, i.e. the minimum green times are equal to zero.

Since we consider a hybrid fluid model and because of the shape of the periodic optimal trajectory shown in Figure 4.7, imposing a maximum green time on signal 1 is essentially the same as imposing a constraint on the maximum queue length of signal 2 and vice versa. This because when signal 1 has a maximum green time g_1^{max} it means that signal 2 has a maximum red time of $g_1^{max} + \sigma_{1,2,1}$. Therefore, the queue length of signal 2 can be equal to maximally $\lambda_2(g_1^{max} + \sigma_{1,2,1})$. When signal 2 is also subject to a queue length constraint, i.e. its queue can have a maximum length equal to x_2^{max}, it has to be determined which constraint is more restrictive: the maximum green time of signal 1 or the maximum queue length x_2^{max}. For this purpose we introduce the virtual maximum queue lengths x_1^{vmax} and x_2^{vmax} which can be calculated via:

$$\begin{aligned}
x_2^{\text{vmax}} &= \min\{\lambda_2(g_1^{max} + \sigma_{1,2,1}), x_2^{max}\}, \\
x_1^{\text{vmax}} &= \min\{\lambda_1(g_2^{max} + \sigma_{1,2,1}), x_1^{max}\}.
\end{aligned}$$

When the first term realizes this minimum, the maximum green time of the other signal is more restrictive than the maximum queue length. When the second term realizes this minimum, the maximum queue length is more restrictive than the maximum green time of the other signal.

However the reverse also holds: the maximum queue length of signal 1 can be seen as a maximum green time of signal 2 and vice versa. With the same reasoning we can find the virtual maximum green times g_1^{vmax} and g_2^{vmax}, which can be calculated using:

$$g_1^{\text{vmax}} = \min\{g_1^{max}, \frac{x_2^{max}}{\lambda_2} - \sigma_{1,2,1}\}, \tag{4.16a}$$

$$g_2^{\text{vmax}} = \min\{g_2^{max}, \frac{x_1^{max}}{\lambda_1} - \sigma_{1,2,1}\}. \tag{4.16b}$$

Note that the virtual maximum green time g_1^{vmax} and g_2^{vmax} and the virtual maximum queue lengths x_1^{vmax} and x_2^{vmax} are related according to:

$$\begin{aligned}
x_1^{\text{vmax}} &= \lambda_1(g_2^{\text{vmax}} + \sigma_{1,2,1}), \\
x_2^{\text{vmax}} &= \lambda_2(g_1^{\text{vmax}} + \sigma_{1,2,1}).
\end{aligned}$$

When only discarding constraints on minimum green times we can still use the knowledge from section 5.4 of [10]. Instead of using maximum queue lengths x_1^{max} and x_2^{max} we virtual maximum queue lengths x_1^{vmax} and x_2^{vmax}.

4.8 Periodic Optimal Trajectory

In this section we consider all behavioral equations that are given in Section 4.2.2. From Corollary 4.4 we know that w.l.o.g. we can assume that optimal trajectories are periodic. For these periodic optimal trajectories a queue is emptied during each green period. For these periodic trajectories we can rewrite the behavioral equations in Section 4.2.2.

We want to minimize the linear cost function (4.13). From the right side of Figure 4.7 we can obtain the following expression for the linear cost function of the simple intersection with two signals.

$$\begin{aligned} J &= \frac{1}{c}\int_0^c [w_1 x_1(s) + w_2 x_2(s)]ds, \\ &= \frac{w_1(\sigma_{1,2,1} + g_1^\mu + g_2)(\sigma_{1,2,1} + g_2)\lambda_1}{2(\sigma_{1,2,1} + g_1 + g_2)}, \\ &+ \frac{w_2(\sigma_{1,2,1} + g_2^\mu + g_1)(\sigma_{1,2,1} + g_1)\lambda_2}{2(\sigma_{1,2,1} + g_1 + g_2)}. \end{aligned} \tag{4.17}$$

Using (4.9) we can rewrite (4.17) to:

$$J = \frac{\frac{\lambda_1 w_1}{2(1-\rho_1)}(g_2 + \sigma_{1,2,1})^2 + \frac{\lambda_2 w_2}{2(1-\rho_2)}(g_1 + \sigma_{1,2,1})^2}{g_1 + g_2 + \sigma_{1,2,1}}.$$

Multiplying this objective function with $\frac{2(1-\rho_2)}{\lambda_2 w_2}$ results in (4.18). Note that multiplying an objective with a positive constant value does not change the position of the minimum, i.e. the values for g_1 and g_2 that minimize the objective function.

$$\min_{g_1, g_2} \frac{\frac{\lambda_1 w_1 (1-\rho_2)}{\lambda_2 w_2 (1-\rho_1)}(g_2 + \sigma_{1,2,1})^2 + (g_1 + \sigma_{1,2,1})^2}{g_1 + g_2 + \sigma_{1,2,1}}. \tag{4.18}$$

This objective function is subject to the following constraints. The green time of signal $i = 1, 2$ must be large enough for traffic, that arrives during a red period, to depart:

$$g_1 \geq \frac{\rho_1}{1-\rho_1}(\sigma_{1,2,1} + g_2), \tag{4.19a}$$

$$g_2 \geq \frac{\rho_1}{1-\rho_1}(\sigma_{1,2,1} + g_1). \tag{4.19b}$$

The maximum queue length of a signal must be larger than the amount of traffic that arrives during a red period:

$$g_1 \leq \frac{x_2^{max}}{\lambda_2} - \sigma_{1,2,1}, \tag{4.19c}$$

$$g_2 \leq \frac{x_1^{max}}{\lambda_1} - \sigma_{1,2,1}. \tag{4.19d}$$

The duration of a green period must be at least the minimum green time and may not exceed the maximum green time:

$$g_1 \geq g_1^{min}, \tag{4.19e}$$
$$g_2 \geq g_2^{min}, \tag{4.19f}$$
$$g_1 \leq g_1^{max}, \tag{4.19g}$$
$$g_2 \leq g_2^{max}. \tag{4.19h}$$

4.8.1 Solution of the Optimization Problem

Using (4.16) we can see that we can only find values for g_1 and g_2 satisfying constraints (4.19) whenever:

$$g_1^{\text{vmax}} \geq \frac{\rho_1 \sigma_{1,2,1}}{1 - \rho_1 - \rho_2}, \tag{4.20a}$$
$$g_2^{\text{vmax}} \geq \frac{\rho_2 \sigma_{1,2,1}}{1 - \rho_1 - \rho_2}, \tag{4.20b}$$
$$g_1^{\text{vmax}} \geq g_1^{min}, \tag{4.20c}$$
$$g_2^{\text{vmax}} \geq \frac{\rho_2}{1 - \rho_2}(\sigma_{1,2,1} + g_1^{min}), \tag{4.20d}$$
$$g_2^{\text{vmax}} \geq g_2^{min}, \tag{4.20e}$$
$$g_1^{\text{vmax}} \geq \frac{\rho_1}{1 - \rho_1}(\sigma_{1,2,1} + g_2^{min}). \tag{4.20f}$$

These inequalities make sure that the smallest possible periodic trajectory is possible without violating any constraints. Inequalities (4.20a) and (4.20b) make sure that the pure bow tie curve does not exceed the maximum queue lengths or exceed the maximum green times.

When the pure bow tie curve violates the minimum green times, inequalities (4.20c),(4.20d), (4.20e),(4.20f) make sure that either the smallest periodic trajectory where $g_1 = g_1^{min}$ or the smallest periodic trajectory where $g_2 = g_2^{min}$ is possible without violating any constraints.

This optimization problem can be solved analytically (see Appendix C.1). The periodic optimal trajectory can have 0, 1 or 2 slow modes. For more information see Appendix C.1. In this appendix we use the notation shown below. We assume w.l.o.g. that $0 < k \leq 1$.

$$
\begin{aligned}
k &= \frac{w_2\lambda_2(1-\rho_1)}{w_1\lambda_1(1-\rho_2)}, \\
y_1 &= \frac{g_1}{\sigma_{1,2,1}}, \\
y_2 &= \frac{g_2}{\sigma_{1,2,1}}, \\
y_1^{min} &= \frac{g_1^{min}}{\sigma_{1,2,1}}, \\
y_2^{min} &= \frac{g_2^{min}}{\sigma_{1,2,1}}, \\
y_1^{max} &= \frac{\min\{g_1^{max}, \frac{x_2^{max}}{\lambda_2} - \sigma_{1,2,1}\}}{\sigma_{1,2,1}}, \\
y_2^{max} &= \frac{\min\{g_2^{max}, \frac{x_1^{max}}{\lambda_1} - \sigma_{1,2,1}}{\sigma_{1,2,1}}.
\end{aligned}
$$

Chapter 5

Regulation: A Simple Intersection of Two Signals

In the previous chapter we considered the trajectory optimization problem. We showed how to derive periodic optimal trajectories for a simple intersection of two signals. To obtain these desired trajectories we assumed deterministic arrivals and deterministic departures. However, due to for example stochastic arrivals we may deviate from the desired trajectory. The second problem discussed in this thesis is regulation. In this chapter we consider the regulation problem for the simple intersection of two signals. First we explain the regulation problem more explicitly in Section 5.1. Subsequently, we propose a policy in Section 5.2.

5.1 Problem Description

In Polderman and Willems [24], the problem of regulation is described as the problem to design mechanisms that keep certain to be controlled variables at constant values against external disturbances that act on the plant that is being regulated or against changes in its properties. The system that is being controlled is usually referred to as the plant.

One of the central concepts of regulation is feedback; some of the variables in the plant are measured and used to determine what control actions to take. A feedback loop is depicted in Figure 5.1. Some variables are measured by sensors and send to the feedback controller. From these measured variables the controller determines what control inputs to send to the actuators.

In our case the components depicted in this figure are as follows.

Plant: The intersection.
Actuators: The color of a traffic light can change.
Sensors: Sensors that could measure queue lengths.
Exogenous-inputs Traffic arriving at the intersection.
To-be-controlled-output: The queue lengths.
Measured-outputs: The queue lengths.
Control-inputs: The signal state of each of the signals.

We want to find a policy to implement in the feedback controller. A policy is a set of rules that convert the measured outputs to the control inputs. This policy should make sure that when we deviate from the optimal trajectory (that follows from the trajectory optimization problem) we again return

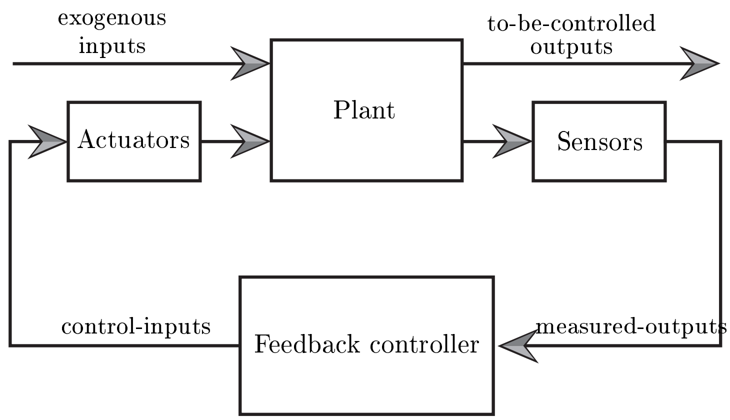


Figure 5.1: Visualization of a feedback loop.

to this optimal trajectory. In this chapter we model the intersection using the hybrid fluid model (see section 3.2) and we use the same assumptions as presented in Section 4.2.3.

5.2 Proposing a Policy

In this section we propose a policy for the simple intersection with two signals. We prove that for a hybrid fluid model a trajectory converge to the periodic optimal trajectory whenever this is possible.

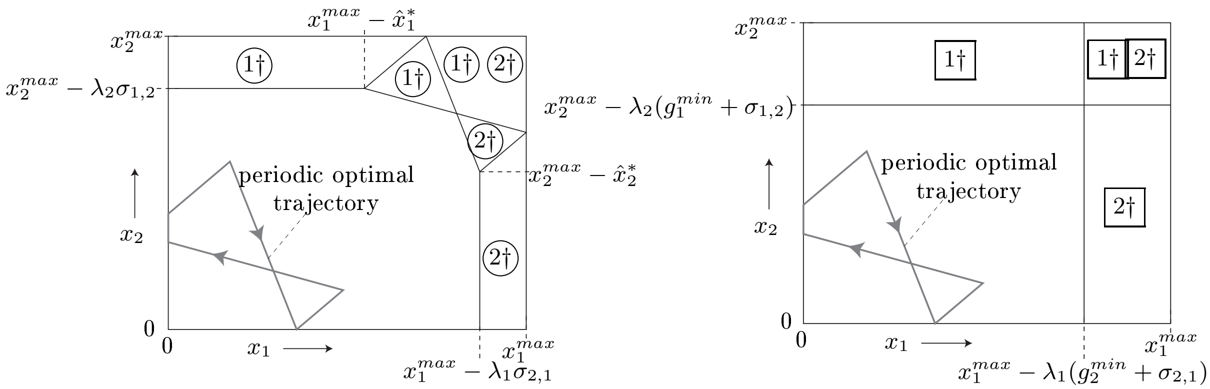
Convergence to the periodic optimal trajectory is not always possible. The (x_1, x_2)-plane can be divided into regions from which it is impossible to converge to the periodic optimal trajectory when in a certain mode. When entering the area annotated with 1† (see Figure 5.2a) while serving signal 1, eventually one of the constraints is violated. When performing a setup to signal 2, a maximum queue length is exceeded. Moreover, if we do not perform this setup, a maximum queue length is exceeded as well. Similarly, whenever entering the area annotated with 2† while serving signal 2, eventually one of the constraints is violated. If the trajectory is on the pure bow tie curve in the upper right corner the trajectory stays here (if the minimum green times allow so).

Further, because of the restrictions on the minimum green period duration, we may not start serving signal 1 respectively signal 2 in the areas annotated with $\boxed{1\dagger}$ respectively $\boxed{2\dagger}$ (see Figure 5.2b). Hence, when the initial queue lengths are in the area annotated with $\boxed{1\dagger}$ we have to start serving signal 2 and when the initial queue lengths are in the area annotated with $\boxed{2\dagger}$ we have to with serving signal 1. When the initial queue lengths are in the area with both $\boxed{1\dagger}$ and $\boxed{2\dagger}$, eventually a constraint is violated.

Assuming a hybrid fluid model, the policy must satisfy the restrictions on green times and the restrictions on maximum queue lengths. Note that in a stochastic setting it is theoretically impossible to make sure that a maximum queue length is not exceeded when assuming Poisson arrivals. To satisfy these restrictions on green times and maximum queue lengths, signal $i = 1, 2$ may only switch to red whenever:

$$L^i_\tau(t) \geq g_i^{min}.$$

Further, a signal must be switched to red whenever the maximum green time is reached:



(a) When entering the area annotated with an encircled 1† or an encircled 2† while serving signal 1 respectively signal 2, no convergence to the periodic optimal trajectory is possible anymore.

(b) When starting to serve signal 1 or signal 2 in the areas annotated with a boxed 1† respectively a boxed 2†, a constraint will be violated.

Figure 5.2: Convergence to the periodic optimal process is not always possible.

$$L_\tau^i(t) \geq g_i^{max}.$$

Signal 1 must switch to red whenever otherwise the maximum queue length of signal 2 is exceeded (assuming a hybrid fluid model). Thus signal 1 must switch to red when:

$$x_2(t) \geq x_2^{max} - \lambda_2\sigma_{1,2}.$$

Signal 2 must switch to red whenever otherwise the maximum queue length of signal 1 is exceeded (assuming a hybrid fluid model). Thus signal 2 must switch to red when:

$$x_1(t) \geq x_1^{max} - \lambda_1\sigma_{2,1}.$$

The policy proposed in Proposition 5.1 satisfies these restrictions.

Proposition 5.1 *A feedback policy which stabilizes an intersection with two signals to the desired periodic optimal trajectory if started from a feasible starting point (see Figure 5.2) is given by:*

- *Mode 1: Serve signal 1 at the highest possible departure rate. When* $(x_1(t) = 0 \wedge L_\tau^1(t) \geq g_1^{min} \wedge x_2(t) \geq x_2^\sharp) \vee L_\tau^1(t) \geq g_1^{max} \vee x_2(t) \geq x_2^{max} - \lambda_2\sigma_{1,2}$ *switch signal 1 to red and go to Mode 2.*
- *Mode 2: After* $\sigma_{1,2}$ *seconds go to Mode 3.*
- *Mode 3: Serve signal 2 at the highest possible departure rate. When* $(x_2(t) = 0 \wedge L_\tau^2(t) \geq g_2^{min} \wedge x_1(t) \geq x_1^\sharp) \vee L_\tau^2(t) \geq g_2^{max} \vee x_1(t) \geq x_1^{max} - \lambda_1\sigma_{2,1}$ *switch signal 2 to red and go to Mode 4.*
- *Mode 4: After* $\sigma_{2,1}$ *seconds go to Mode 1.*

Where $x_1^\sharp$ *and* $x_2^\sharp$ *are calculated via (4.10).*

Proof. See Appendix 8.1. In this appendix, we actually prove Proposition 8.1 which is proposed in Section 8. In Proposition 8.1 we propose a policy for an intersection with two signal groups. For an intersection with two signals this policy reduces to the policy proposed in Proposition 5.1. ■

Chapter 6

Quality of the Policy in a Stochastic Setting: A Simple Intersection of Two Signals

In Chapter 4 we derived periodic optimal trajectories for a simple intersection of two (conflicting) signals by modeling the intersection with a hybrid fluid model. In Chapter 5 we proposed a (feed-back) policy. In this chapter we consider the third problem discussed in this thesis: We address the quality of the proposed policy for an intersection with two signals in a stochastic setting. To this end, we model the intersection with the stochastic model described in Section 3.1. Recall that this stochastic model assumes Poisson arrivals and deterministic departures. To obtain results for the policy in a stochastic setting, a simulation program is made in the programming language χ3.0. The code of this simulation program is given in Appendix B.

For each test case we obtain the average delay δ (in seconds) of a road user at the intersection and we obtain the fraction of the time that the maximum queue length is exceeded at each of the queues. A road user could either be a vehicle, a cyclist or a pedestrian.

Before simulating a test case we calculate the following information about the periodic optimal trajectory (see Chapter 4.8).

- The coordinates $x_1^\sharp$ and $x_2^\sharp$ calculated with (4.10).
- The cycle duration $c = g_1 + g_2 + \sigma_{1,2,1}$.
- The queue length at signal 1 at the beginning of a green period, which is equal to $\lambda_1(g_2 + \sigma_{1,2,1})$.
- The average delay of a road user. This average delay is obtained using (4.18), where $w = w_1 = w_2 = 1$ and (4.3).

For each test case we perform at least 100 runs. We perform enough runs such that the 95% confidence interval for the average delay of a road user is at most 1% of the average delay of a road user. For each run we start serving signal 1. At the start of a run the queue length of queue 1 is taken equal to $\lceil \lambda_1(g_2 + \sigma_{1,2,1}) \rceil$ (obtained from the periodic optimal trajectory) and the queue length of queue 2 is zero. Each run simulates $100c$ seconds, were c is the cycle duration of the periodic optimal trajectory (see Section 4.4). We consider the following test cases.

test case 1	test case 2a	test case 2b
$\mu_1 = 0.5$	$\mu_1 = 0.5$	$\mu_1 = 0.5$
$\mu_2 = 0.5$	$\mu_2 = 0.5$	$\mu_2 = 0.5$
$\lambda_1 = \lambda_2 = 0.0125, 0.0250, \ldots, 0.2375$	$\lambda_1 = \frac{1}{15}$	$\lambda_1 = \frac{1}{15}$
	$\lambda_2 = \frac{5}{15}$	$\lambda_2 = \frac{5}{15}$
$\sigma_{1,2} = 2$	$\sigma_{1,2} = 2$	$\sigma_{1,2} = 2$
$\sigma_{2,1} = 2$	$\sigma_{2,1} = 2$	$\sigma_{2,1} = 2$
$g_1^{min} = 4$	$g_1^{min} = 4$	$g_1^{min} = 4$
$g_2^{min} = 4$	$g_2^{min} = 4$	$g_2^{min} = 4$
$g_1^{max} = \infty$	$g_1^{max} = \infty$	$g_1^{max} = 4, 4.5, \ldots, 9$
$g_2^{max} = \infty$	$g_2^{max} = 16, 16.5, \ldots, 26$	$g_2^{max} = \infty$
$x_1^{max} = \infty$	$x_1^{max} = \infty$	$x_1^{max} = \infty$
$x_2^{max} = \infty$	$x_2^{max} = \infty$	$x_2^{max} = \infty$

test case 3a	test case 3b
$\mu_1 = 0.5$	$\mu_1 = 0.5$
$\mu_2 = 0.5$	$\mu_2 = 0.5$
$\lambda_1 = \frac{1}{15}$	$\lambda_1 = \frac{1}{15}$
$\lambda_2 = \frac{5}{15}$	$\lambda_2 = \frac{5}{15}$
$\sigma_{1,2} = 2$	$\sigma_{1,2} = 2$
$\sigma_{2,1} = 2$	$\sigma_{2,1} = 2$
$g_1^{min} = 4$	$g_1^{min} = 4$
$g_2^{min} = 4$	$g_2^{min} = 4$
$g_1^{max} = \infty$	$g_1^{max} = \infty$
$g_2^{max} = \infty$	$g_2^{max} = \infty$
$x_1^{max} = \infty$	$x_1^{max} = 2, 3, \ldots, 7$
$x_2^{max} = 3, 4, \ldots, 30$	$x_2^{max} = \infty$

For test cases 2a, 2b, 3a and 3b it holds that $\rho_1 + \rho_2 = 0.8$. In this chapter we use $\mu = \mu_1 = \mu_2$. In sections 6.2–6.4 we show the results for these test cases

6.1 Theoretical Comparison to Exhaustive Policy

For the small intersection with 2 signals we want to compare the average delay of a road user obtained for our proposed policy to the average delay for an exhaustive policy. This exhaustive policy works as follows. A signal $i = 1, 2$ is always served until it is emptied (disregarding minimum green times, maximum green times and maximum queue lengths). When queue i is emptied, signal i switches to red and as soon as the setup time has elapsed the other signal switches to green. Thus, for the exhaustive policy there are no slow modes. Whenever both queues are empty, the exhaustive policy results in the following switch behavior. Whenever a queue is empty at the moment that it may switch to green (the setup time towards this signal is finished), this signal does not switch to green and we immediately start performing a setup towards the other signal. Thus, whenever both queues are empty, constantly setups are performed.

This exhaustive policy is analyzed in [3]. From [3] we can obtain an expression for the average delay of a vehicle for this exhaustive policy. This expression is given in (6.1). This equation assumes equal maximum departure rates, i.e. $\mu = \mu_1 = \mu_2$.

$$\delta = \frac{\rho}{2\mu(1-\rho)} + \frac{\sigma_{1,2,1}}{2} + \frac{\sigma_{1,2,1}\rho_1\rho_2}{\rho(1-\rho)} + \frac{1}{\mu}, \tag{6.1}$$

where

$$\rho = \rho_1 + \rho_2.$$

We compare the average delay of a road user obtained via simulation for our proposed policy to the average delay of the exhaustive policy obtained with (6.1).

6.2 Test Case 1: Effect of the Arrival Rates

In this test case we address the effect of the arrival rates on the delay; we want to determine $\delta(\lambda)$ for the proposed policy, where the arrival rates are varied as follows:

$$\lambda = \lambda_1 = \lambda_2 = 0.0125, 0.025, ..., 0.2375.$$

As a result $\rho = \rho_1 + \rho_2$ varies as follows:

$$\rho = 0.05, 0.1, ..., 0.95.$$

In Figure 6.1 the results are shown.

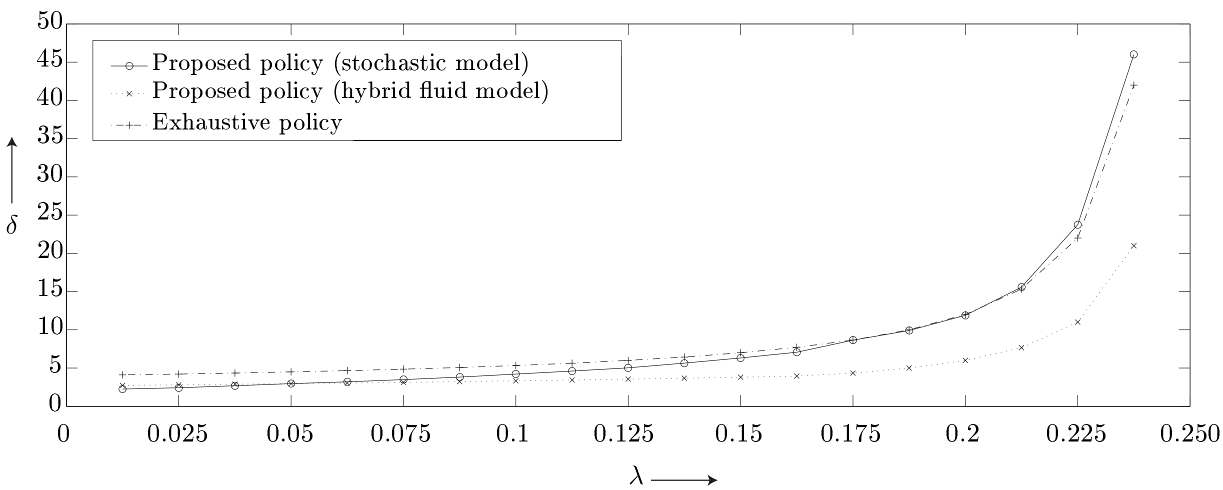


Figure 6.1: The average delay of a road user δ versus λ for test case 1.

For this test case the average delay $\delta(\lambda)$ goes to 2 for the proposed policy when $\lambda \to 0$ (assuming the stochastic model). We can explain this. For the proposed policy the signal that is green stays green when both queues are empty. For $\lambda \to 0$ the probability that both queues are empty at the time of an arrival is equal to 1. Since the maximum green times are infinite for this test case, the probability that the minimum green time has elapsed at the moment of an arrival goes to 1 for $\lambda \to 0$ (the inter-arrival time goes to infinity for $\lambda \to 0$) and the probability that no setup is being performed at the moment of an arrival goes to 1 for $\lambda \to 0$. Whenever a road user arrives at the signal that is red, the other signal switches to red immediately. This road user experiences a delay of 4 seconds: a setup time equal to 2 seconds plus a departure time equal to $\frac{1}{\mu} = 2$ seconds. Whenever a road user arrives at the signal that is green, this road user can cross the intersection without any delay. Since the arrival rates of both signals are the same (for this test case), the probability that an arbitrary road user arrives at signal $i = 1, 2$

is equal to 0.5. Hence, the average delay of a road user goes to 2 seconds for $\lambda \to 0$. Note that the average delay $\delta(\lambda)$ only goes to 2 for $\lambda \to 0$ if a maximum green time is infinite. When both maximum green times are finite the probability that at the moment of an arrival no setup is being performed and the minimum green time has elapsed does not go to 1 for $\lambda \to 0$.

Recall that the exhaustive policy serves signal $i = 1, 2$ until it is emptied disregarding minimum green times. For the exhaustive policy the average delay $\delta(\lambda)$ goes to 4 for $\lambda \to 0$ because for the exhaustive policy constantly a setup is performed (either $\sigma_{1,2}$ or $\sigma_{2,1}$) whenever both queues are empty. The probability that both queues are empty at the time of an arrival is equal to 1 for $\lambda \to 0$. Whenever a road user arrives, on average it takes $\frac{\sigma_{1,2,1}}{2} = 2$ seconds before this signal is switched to green for the exhaustive policy (see Section 6.1). After this residual setup of 2 seconds and a departure time of $\frac{1}{\mu} = 2$ seconds, this road user has crossed the intersection with a delay of 4 seconds.

Hence, the proposed policy works better than the exhaustive policy for $\lambda \to 0$. Note that there are policies that result in even lower values for the average delay $\delta(\lambda)$ for $\lambda \to 0$. For example when both signals are red if both queues are empty. At the moment of an arrival at signal $i = 1, 2$ we immediately switch this signal to green (if the other signal has been red for 2 seconds). Assuming infinite maximum green times, this policy results in an average delay $\delta(\lambda)$ of 0 seconds for $\lambda \to 0$.

Further, the proposed policy might result in smaller delays than the exhaustive policy because the proposed policy allows slow modes. A slow mode could reduce the average delay because the system switches less, i.e. there are less setups.

For low values of $\delta(\lambda)$ the average delay of the proposed policy is smaller for the stochastic model than the average delay obtained via trajectory optimization (Section 4).

For larger values of λ the average delay obtained via trajectory optimization is an underestimation of the average delay in the stochastic setting. For these larger values of λ the exhaustive policy results in lower values for $\delta(\lambda)$ than the proposed policy. For large values of $\delta(\lambda)$ the periodic optimal trajectories obtained via trajectory optimization do not have a slow mode. However, for large values of $\delta(\lambda)$ we still observe slow modes for the proposed policy in a stochastic setting. These slow modes cause the difference in $\delta(\lambda)$ for the proposed policy and the exhaustive policy at large values for λ. Thus, these slow modes have a positive effect for smaller values of λ and they have a negative effect for larger values of λ.

6.3 Test Case 2: Effect of The Maximum Green Time

In this section we address the effect of the maximum green times on the delay of a road user. For test case 2a and test case 2b the arrival rate at signal 2 is 5 times as large as the arrival rate at signal 1. We use 'low traffic signal' to refer to signal 1 and we use 'high traffic signal' to refer to signal 2.

6.3.1 Test Case 2a: Effect of The Maximum Green Time of the High Traffic Signal

For this test case the maximum green time of signal 2 is varied between 16 seconds and 26 seconds. A maximum green time of 16 seconds is the smallest maximum green time g_2^{max} satisfying (4.20) and thus the smallest maximum green time for which we can find an optimal trajectory. In Figure 6.2 we can see the results for test case 2a. The result obtained for $g_2^{max} = 16$ seconds is not shown in this figure because it results in instability: the queue length of queue 2 keeps increasing. We can explain this instability as follows. For the hybrid fluid model, during the maximum green time $g_2^{max} = 16$ seconds the traffic that arrives during a red period (with duration $g_1^{min} + \sigma_{1,2,1}$) can precisely depart during a green period:

$$g_2^{max} = \frac{\rho_2}{1-\rho_2}(g_1^{min} + \sigma_{1,2,1}).$$

Due to determinism, for the hybrid fluid model the red time of signal 2 is always equal to 8 seconds (the minimum green time of signal 1 plus the setup times). However, when including stochastic arrivals the average red time is greater than 8 seconds because every red time is at least 8 seconds (otherwise we do not satisfy the minimum green time of signal 1) and the red time exceeds 8 seconds whenever at least 3 road users depart during a green period of signal 1. This larger average red time causes the instability.

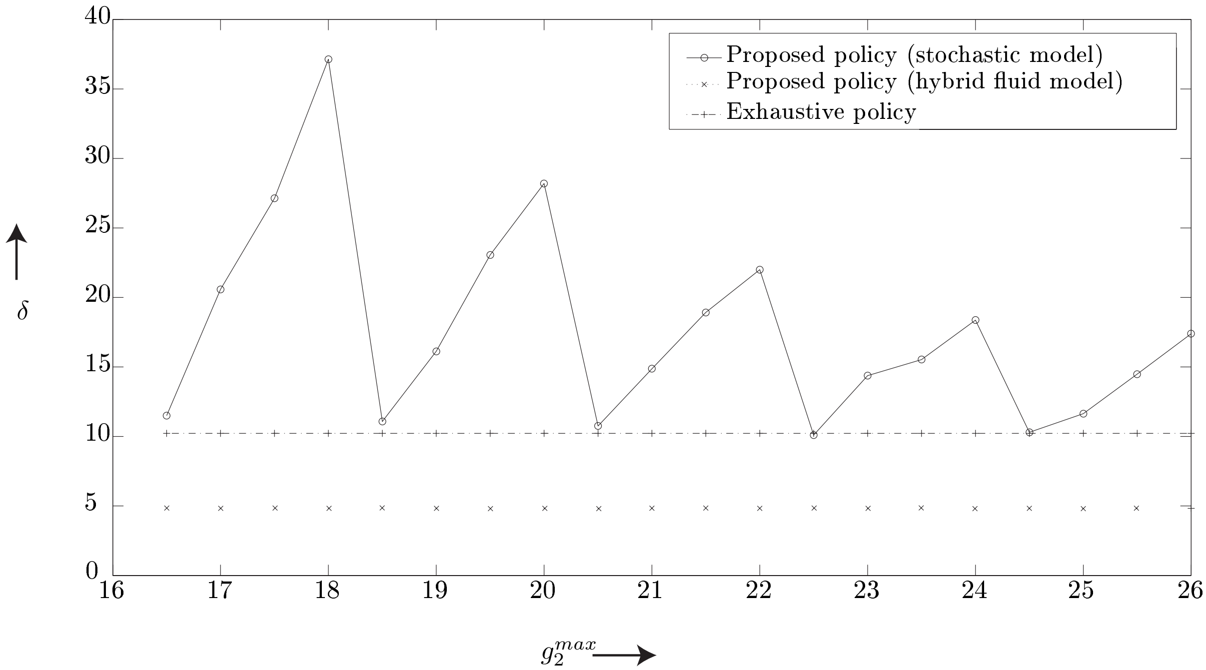


Figure 6.2: The average delay of a road user δ versus g_2^{max} for test case 2a.

In this figure we can see that the relation between the maximum green time and the average delay has the shape of a sawtooth. The proposed policy works better when the maximum green time is not a multiple of the inter-departure time $\frac{1}{\mu}$. This can be explained since a new departure process is started when, at the moment of a departure, the corresponding signal is green and its queue is not empty (see Section 3.1). Hence, during a maximum green time of g_2^{max}, $\lceil g_2^{max}\mu \rceil$ road users depart. We can see the function $\lceil g_2^{max}\mu \rceil$ for the different values of g_2^{max} in Figure 6.3. Thus, the number of departures during a maximum green period of 20 seconds is the same as the number of departures during a green period of 18.5 seconds which causes the sawtooth relation between the maximum green time and the average delay of a road user.

In Figure 6.2 we can see that the global trend (disregarding the sawtooth shape) is that smaller maximum green times result in larger delays.

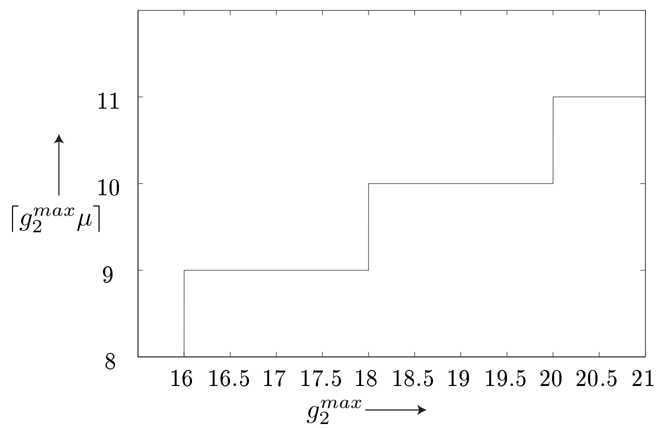


Figure 6.3: The number of departures during g_2^{max}.

6.3.2 Test Case 2b: Effect of the Maximum Green Time of the Low Traffic Signal

For test case 2b the maximum green time of signal 2 is infinite and the maximum green time of signal 1 is varied between 4 seconds and 9 seconds. A maximum green time of 4 seconds is the smallest maximum green time g_1^{max} satisfying (4.20) and thus the smallest maximum green time for which we can find an optimal trajectory. For the optimal trajectory we serve signal 1 for the minimum green time g_1^{min} (independent of g_1^{max}). In Figure 6.4 we can see the results.

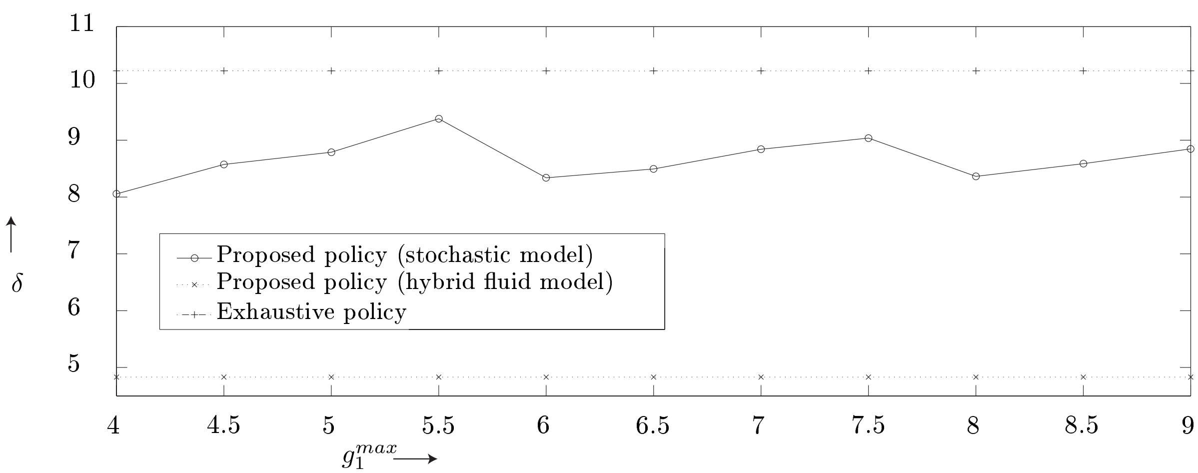


Figure 6.4: The average delay of a road user δ versus g_1^{max} for test case 2b.

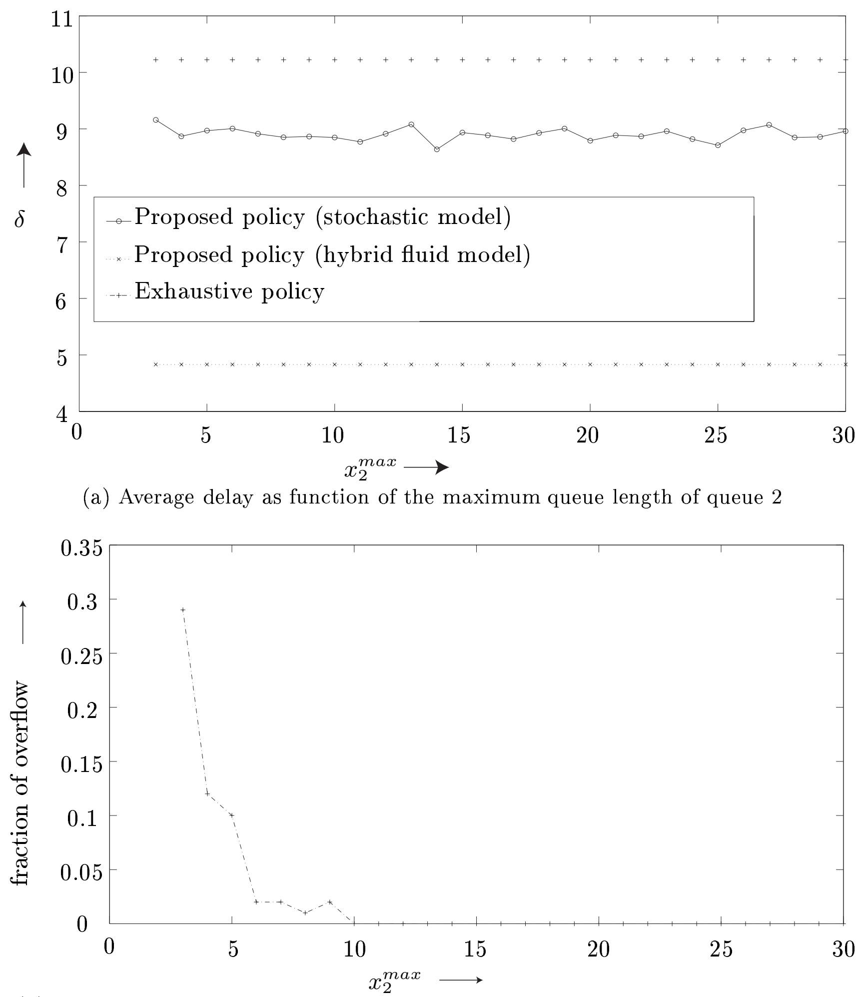
We again see the sawtooth relation between the maximum green time and the average delay of a road user. At signal 1 on average more traffic can depart during a minimum green time than what arrives during a red time. As a result, the low traffic signal is often already emptied before the minimum green time is reached (for all values of g_1^{max}). Hence, the effect of changing the maximum green time is (except for the sawtooth shape) limited.

6.4 Test Case 3: Effect of Maximum Queue lengths

In this section we address the effect of the maximum queue lengths. For test case 3a and test case 3b the arrival rate at signal 2 is 5 times as large as the arrival rate at signal 1. We use 'low traffic signal' to refer to signal 1 and we use 'high traffic signal' to refer to signal 2.

6.4.1 Test Case 3a: Maximum Queue Length of the High Traffic Signal

For test case 3a the maximum queue length of signal 1 is infinite and we vary the maximum queue length of signal 2 between 3 road users and 30 road users. A maximum queue length of $2\frac{2}{3}$ seconds is the smallest maximum queue length satisfying (4.20) and thus the smallest maximum queue length for which we can find an optimal trajectory. In Figure 6.5 we can see the results.



(a) Average delay as function of the maximum queue length of queue 2

(b) Fraction of the time that the maximum queue length at queue 2 is exceeded as function of the maximum queue length of queue 2

Figure 6.5: Effect of the maximum queue length of queue 2 for test case 3a.

Note that in a stochastic setting we switch signal 1 to red at the moment that $x_2(t) \geq x_2^{max} - \sigma_{1,2}\lambda_2$ (also whenever the minimum green time is not satisfied). Hence, for smaller x_2^{max} a green period of signal 1 is sometimes shorter than the minimum green time. As a result, changing the maximum queue length of queue 2 has a limited effect on the average delay of a road user. However, we can see that for smaller maximum queue lengths x_2^{max} the maximum queue length is exceeded more often.

When comparing Figure 6.5a with Figure 6.1 we can also see the effect of asymmetrical arrival rates. In Figure 6.1 we can see that when assuming infinite maximum green times and infinite queue lengths, the delay is about 12 seconds for symmetrical arrival rates and $\lambda = \lambda_1 = \lambda_2 = 0.2$ (resulting in $\rho_1 + \rho_2 = 0.8$). In Figure 6.5a we can see that when assuming infinite maximum green times the average delay is about 9 seconds seconds at large values of the maximum queue length for asymmetrical arrival rates (and $\rho_1 + \rho_2 = 0.8$). Thus, asymmetrical arrival rates result in smaller mean delays. We can see that the average delay goes to zero for $\frac{\lambda_2}{\lambda_1} \to \infty$. Assuming infinite maximum green times, the delay goes to zero for $\frac{\lambda_2}{\lambda_1} \to \infty$ because all of the road users arrive at signal 2. Hence, if queue 2 is emptied once, it always stays empty (slow mode).

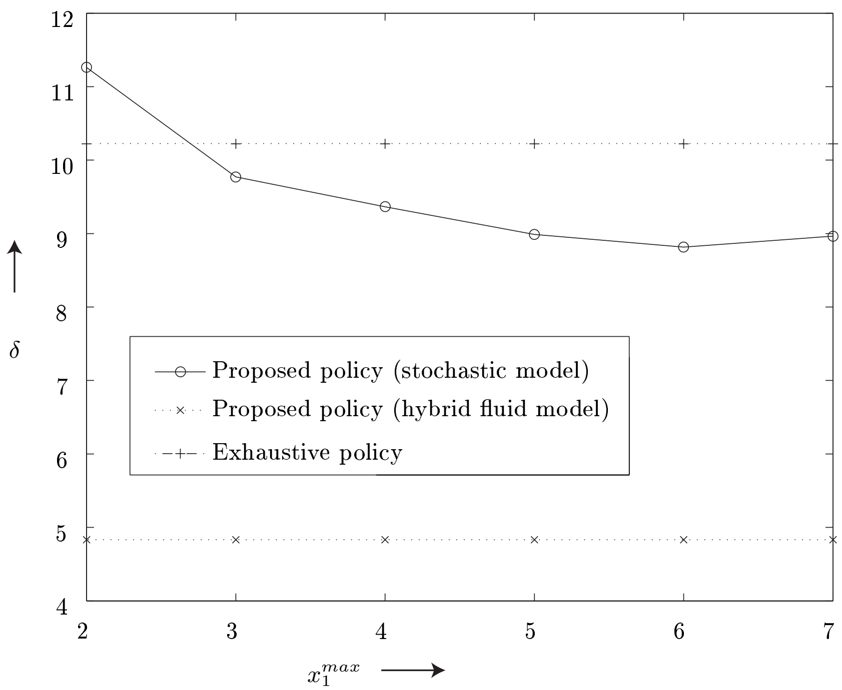
Further, we can see that for test case 3a the proposed policy results in a smaller average delay than the exhaustive policy. For the proposed policy slow modes where observed at signal 2. These slow modes are desirable because most of the traffic arrives at signal 2 and all traffic arriving during a slow mode crosses the intersection without delay. Hence, the proposed policy results in small delays (compared to the exhaustive policy) especially for asymmetrical arrival rates.

In Figure 6.5b we can see that the queue length is exceeded more often when the maximum queue length of queue 2 is smaller. For a maximum queue length of 10 or higher the maximum queue length is (almost) never exceeded.

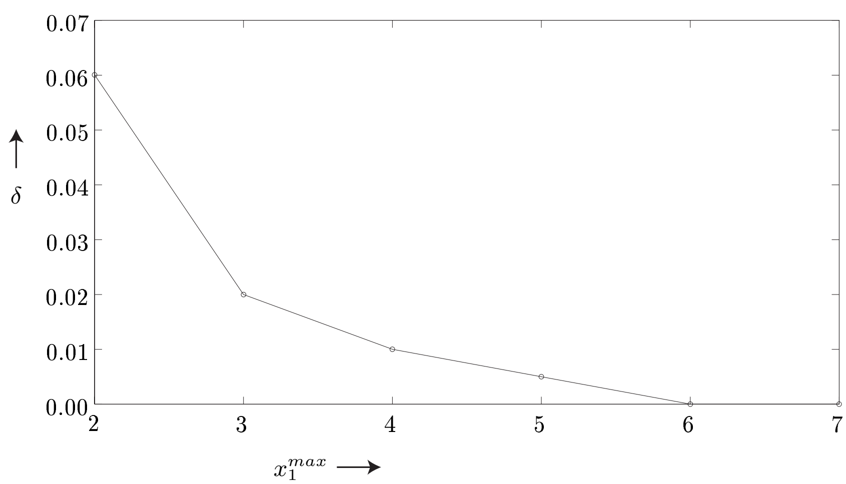
6.4.2 Test Case 3b: Maximum Queue Length of the Low Traffic Traffic Signal

For test case 3b the maximum queue length of signal 2 is infinite and the maximum green time of signal 1 is varied between 2 and 7. A maximum queue length of $1\frac{1}{3}$ seconds is the smallest value for x_1^{min} satisfying (4.20) and thus the smallest value for x_1^{max} for which we can find an optimal trajectory. In Figure 6.6 we can see the results.

We can see that for smaller values of x_1^{max} the average delay of a road user increases because the high traffic signal (signal 2) has to switch to red before its queue is emptied. The road users that could not cross the intersection during the green period experience large delays. Further, for smaller values of x_1^{max}, the maximum queue length of queue 1 is exceeded more often.



(a) Average delay as function of the maximum queue length of queue 1



(b) Fraction of the time that the maximum queue length at queue 1 is exceeded as function of the maximum queue length of queue 1

Figure 6.6: Effect of the maximum queue length of queue 1 for test case 3b.

Chapter 7

Trajectory Optimization: An Intersection with Two Signal Groups

In this chapter we again consider the trajectory optimization problem. However, this time we consider the trajectory optimization for a more general intersection with two signal groups (instead of an intersection with two signals). In Figure 1.2 we showed an example of an intersection with two signal groups. For this example one of the signal groups consists out of signals 1,3,7,8,11,12,15,16,19 and 20 and the other signal group consists out of signals 2,4,5,6,8,10,13,14,17 and 18.

In this chapter we assume without loss of generality that the signals in signal group 1 are numbered $1, 2, ..., N_1$ and that the signals in signal group 2 are numbered $N_1 + 1, N_1 + 2, ..., N$. We use $\mathcal{G}_1 = \{1, 2, ..., N_1\}$ and $\mathcal{G}_2 = \{N_1 + 1, ..., N\}$. First we explain the trajectory optimization problem for this more general intersection in Section 7.1 and Section 7.2. Subsequently, in Section 7.3 we prove that we can always find an optimal trajectory satisfying some properties. Using these properties an optimization problem is proposed in Section 7.5.

7.1 Problem Description

To solve the trajectory optimization problem we model the intersection with the hybrid fluid model given in Section 3.2. In Section 7.2 we present the behavioral equations of the hybrid fluid model for an intersection with two (or more) signal groups. A solution of these behavioral equations is called a trajectory and consists of the evolution (as function of time) of the following variables:

- $x_i(t)$, $i \in \mathcal{N}$: the queue lengths of all signals as a function of time.
- $m_i(t)$, $i \in \mathcal{N}$: the signal state of all signals as a function of time.
- $d_i(t)$, $i \in \mathcal{N}$: the departure rate of all signals as a function of time

We want to find a trajectory minimizing the average weighted queue length:

$$J = \limsup_{t\to\infty} \frac{1}{t} \int_0^t \sum_{i\in\mathcal{N}} f_i(x_i(s))ds, \tag{7.1}$$

where $f_i : \mathbb{R}_+ \to \mathbb{R}_+$ is a weight function. Weight function f_i relates the queue lengths at signal i to costs. We assume that the functions f_i are strictly increasing, i.e. larger queue lengths result in higher costs. In Section 7.5 we use more specific weight functions f_i: the linear weight functions

$f_i(x_i(t)) = w_i x_i(t)$, $i \in \mathcal{N}$. In Section 7.1.1 we show that minimizing the linear weight function, where $w = w_1 = \cdots = w_N$, is equivalent to minimizing the average delay of an arbitrary road user at this intersection.

7.1.1 Average Delay of A Road User At the Intersection

In this section we show that minimizing the linear weight function where $w = w_1 = \cdots = w_N$, is equivalent to minimizing the average delay of an arbitrary road user at this intersection. In this section we assume that each arrival rate λ_i, $i \in \mathcal{N}$ is given in number of vehicles per second, number of cyclists per second or number of pedestrians per second and that each queue length x_i, $i \in \mathcal{N}$ is given in number of vehicles, number of cyclists or number of pedestrians. When different types of traffic arrive at a signal it does not hold that minimizing the linear weight function where $w = w_1 = \cdots = w_N$, is equivalent to minimizing the average delay of an arbitrary road user at this intersection.

Similar to Section 4.1.1, we can find that when $f_i(x_i) = w x_i$, $i \in \mathcal{N}$ we can write (7.1) as follows:

$$J = w\lambda\delta, \tag{7.2}$$

where

$$\begin{aligned} \delta &= \sum_{i \in \mathcal{N}} \delta_i \frac{\lambda_i}{\lambda}, \\ \lambda &= \lambda_1 + \lambda_2. \end{aligned}$$

In this equation δ is the average delay of an arbitrary road user at the intersection and δ_i is the average delay of a road user at signal i.

To obtain (7.2) we have used:

$$\begin{aligned} \overline{x}_i &= \limsup_{t \to \infty} \frac{1}{t} \int_0^t x_i(s) ds, && i \in \mathcal{N}, \\ \delta_i &= \frac{\overline{x}_i}{\lambda_i}, && i \in \mathcal{N}, \\ \lambda &= \sum_{i \in \mathcal{N}} \lambda_i, \end{aligned}$$

where, $\overline{x}_i$ is the average queue length at queue i (including the road user that is departing) and δ_i is the average delay of a road user at signal i.

A fraction $\frac{\lambda_i}{\lambda}$ of the road users arrives at signal $i \in \mathcal{N}$. Hence, $\sum_{i \in \mathcal{N}} \delta_i \frac{\lambda_i}{\lambda}$ is the average delay of an arbitrary road user at the intersection. Note, that the optimal trajectory does not change when multiplying the objective function (cost function) with $\frac{1}{w\lambda} > 0$. Hence, minimizing the linear weight function, where $w = w_1 = \cdots = w_N$ results in the same optimal trajectory as minimizing the average delay of an arbitrary road user at the intersection.

7.2 Behavioral Equations of the Hybrid Fluid Model

In this section we give the behavioral equations of the hybrid fluid model. First, we introduce the variables that we use in these behavioral equations in Section 7.2.1. In Section 7.2.2 we give the behavioral equations of the hybrid fluid model for an intersection with two (or more) signal groups.

7.2.1 Manifest Variables and Latent Variables

We use the following manifest variables:

- $x_i(t) \in \mathbb{R}^+$, $i \in \mathcal{N}$: the queue length of queue i as a function of time. The function $x_i(t)$, $i \in \mathcal{N}$ is right-continuous.
- $m_i(t) \in \{$ⓘ,🛈$\}$, $i \in \mathcal{N}$: the signal state of signal i as a function of time. The function $m_i(t)$, $i \in \mathcal{N}$ is right-continuous.
- $d_i(t) \in \mathbb{R}^+$,$i \in \mathcal{N}$: the departure rate at signal i as a function of time. The function $d_i(t)$, $i \in \mathcal{N}$ is measurable.

Further, we use the following latent variables:

- $L^i_\tau(t) \in \mathbb{R}^+$, $i \in \mathcal{N}$: the time that has elapsed since the last change in the signal state of signal i.

7.2.2 Behavioral Equations

In this section we give the behavioral equations for an intersection with two (or more) signal groups. In these behavioral equations we use:

$$z_{i,j} = z_{j,i} = \begin{cases} 1 & \text{if signal } i \text{ and } j \text{ are conflicting,} \\ 0 & \text{otherwise.} \end{cases}$$

For an intersection with two signal groups, $z_{i,j}$ is 1 whenever signal i and signal j are partitioned in different signal groups. Whenever signal i and signal j are partitioned in the same signal group it holds that $z_{i,j}$ is 0.

The change in the queue length is equal to the net inflow (arrival rate minus departure rate):

$$\dot{x}_i(t) = \lambda_i - d_i(t), \quad i \in \mathcal{N}. \tag{7.3a}$$

The latent variable $L^i_\tau(t)$, $i \in \mathcal{N}$ denotes a time. Hence, its derivative with respect to time is equal to one:

$$\dot{L}^i_\tau(t) = 1, \quad i \in \mathcal{N}. \tag{7.3b}$$

The time that has elapsed since the last change in the signal state, is set to zero when the signal state changes:

$$L^i_\tau(t) = 0 \quad \text{if } m_i(t^-) \neq m_i(t), \quad i \in \mathcal{N}, \tag{7.3c}$$

where

$$m_i(t^-) = \lim_{y \uparrow t} m_i(y).$$

Whenever a signal is red, the traffic from the corresponding queue cannot cross the intersection:

$$d_i(t) = 0 \quad \text{if } m_i(t) = 🛈, \quad i \in \mathcal{N}, \quad \forall t \in \mathbb{R}^+. \tag{7.3d}$$

When there is no traffic waiting at queue $i \in \mathcal{N}$, traffic can depart at a rate that is smaller than or equal to the arrival rate λ_i (otherwise it would result in a negative queue length $x_i(t)$):

$$d_i(t) \leq \lambda_i \quad \text{if } x_i(t) = 0, \quad i \in \mathcal{N}, \quad \forall t \in \mathbb{R}^+. \tag{7.3e}$$

Traffic cannot depart at a rate that exceeds the maximum departure rate:

$$d_i(t) \leq \mu_i, \quad i \in \mathcal{N}, \quad \forall t \in \mathbb{R}^+. \tag{7.3f}$$

Two conflicting signals cannot be green at the same time:

$$m_i(t) = \boldsymbol{❶} \quad \text{if } \exists j \in \mathcal{N}\,(z_{i,j} = 1 \text{ and } m_j(t) = \textcircled{j}), \quad i \in \mathcal{N}. \tag{7.3g}$$

A signal can only switch to green whenever all corresponding setups have been performed:

$$m_i(t) = \boldsymbol{❶} \quad \text{if } \exists j \in \mathcal{N}\left(z_{i,j} = 1 \text{ and } m_j(t) = \boldsymbol{\text{🅙}} \text{ and } L_\tau^j(t) < \sigma_{j,i}\right), \quad i \in \mathcal{N}. \tag{7.3h}$$

The maximum queue length cannot be exceeded:

$$x_i(t) \leq x_i^{max}, \quad i \in \mathcal{N}. \tag{7.3i}$$

The duration of a green period must be at least the minimum green time and cannot exceed the maximum green time:

$$m_i(t) = \textcircled{i} \quad \text{if } T_\tau^i(t) < g_i^{min} \wedge m_i(t^-) = \textcircled{i}, \quad i \in \mathcal{N}, \tag{7.3j}$$

$$m_i(t) = \boldsymbol{❶} \quad \text{if } T_\tau^i(t) \geq g_i^{max} \wedge m_i(t^-) = \textcircled{i}, \quad i \in \mathcal{N}. \tag{7.3k}$$

A solution of these behavioral equations (the manifest variables as function of time) is called a trajectory. Note that we allow every initial condition as long as it satisfies (7.3).

7.2.3 Assumptions

We assume that the arrival rate and the maximum departure rate of a signal is positive:

$$\lambda_i, \mu_i > 0, \quad i \in \mathcal{N}. \tag{7.4a}$$

We assume that all setup times are non-negative and that the setup $\sigma_{1,N,1}$ is strictly positive:

$$\sigma_{i,j} \geq 0, \quad i, j \in \mathcal{N}, \tag{7.4b}$$

$$\sigma_{1,N,1} > 0. \tag{7.4c}$$

We assume that the minimum green times are non-negative:

$$g_i^{min} \geq 0, \quad i \in \mathcal{N}. \tag{7.4d}$$

We assume that for all signals the average green time and the average red time converges. Thus, we assume that the following limits exist for all signals in $\mathcal{N}$:

$$\bar{g}_i = \lim_{M \to \infty} \sum_{k=1}^{M} \frac{g_i^k}{M},$$

$$\bar{r}_i = \lim_{M \to \infty} \sum_{k=1}^{M} \frac{r_i^k}{M}.$$

We only consider trajectories where the signals are served in a fixed order. This is a desirable feature in practice because some of the vehicles, cyclists and pedestrians already start to accelerate when they expect their signal to switch to green. When the order in which these signals are served changes, these expectations may be wrong and can result in unsafe situations. For a fixed order for an intersection with two signal groups each signal in $\mathcal{G}_1$ is green during the red period of the signals in $\mathcal{G}_2$ and each signal in $\mathcal{G}_2$ is green during during the red period of the signals in $\mathcal{G}_1$. Note that because we only consider non-negative setup times, each signal in $\mathcal{G}_1$ is red whenever a signal in $\mathcal{G}_2$ is green and vice versa.

For example when $\mathcal{G}_1 = \{1\}$ and $\mathcal{G}_1 = \{2, 3\}$, we only consider the trajectory where signals 2 and 3 are both served during each red time of signal 1. In Section 7.5.4, we show that another trajectory (that does not satisfy this property) might result in a lower value for the cost function 7.1.

Further, we assume that the setup times are related according to:

$$\sigma_{i_1,i_2} - \sigma_{i_1,l_2} = \sigma_{l_1,i_2} - \sigma_{l_1,l_2}, \qquad \forall i_1, l_1 \in \mathcal{G}_1 \qquad i_2, l_2 \in \mathcal{G}_2 \tag{7.4e}$$

$$\sigma_{i_2,i_1} - \sigma_{i_2,l_1} = \sigma_{l_2,i_1} - \sigma_{l_1,l_2}, \qquad \forall i_1, l_1 \in \mathcal{G}_1 \qquad i_2, l_2 \in \mathcal{G}_2 \tag{7.4f}$$

Using this assumption we can always switch signal i_2 to green $\sigma_{i_1,i_2} - \sigma_{i_1,l_2}$ seconds after (if $\sigma_{i_1,i_2} - \sigma_{i_1,l_2} \geq 0$) or before (if $\sigma_{i_1,i_2} - \sigma_{i_1,l_2} < 0$) signal l_2 switches to green and we can always switch signal i_1 to green $\sigma_{i_2,i_1} - \sigma_{i_2,l_1}$ seconds after (if $\sigma_{i_2,i_1} - \sigma_{i_2,l_1} \geq 0$) or before (if $\sigma_{i_2,i_1} - \sigma_{i_2,l_1} < 0$) signal l_1 switches to green.

Whenever a green period is extremely short or extremely long (and as a result a red period of another signal is extremely long), road users can get irritated which probably results in more red negation, i.e. in more people ignoring a red light. Further, whenever a green period is extremely short or extremely long, road users might think the traffic lights are malfunctioning. From this practical point of view it is logical to assume that we are given restrictions on the maximum red times instead of a restriction on the maximum green times. Hence, we assume (7.4g) and (7.4h). Note that $g^{max}_{i_1} + \sigma_{i_1,i_2,i_1}$, $i_1 \in \mathcal{G}_1, i_2 \in \mathcal{G}_2$ is the maximum duration of a red period of signal i_2 and that $g^{max}_{i_2} + \sigma_{i_1,i_2,i_1}$, $i_1 \in \mathcal{G}_1, i_2 \in \mathcal{G}_2$ is the maximum duration of a red period of signal i_1.

$$g^{max}_{i_1} + \sigma_{i_1,i_2,i_1} = g^{max}_{j_1} + \sigma_{j_1,i_2,j_1}, \quad \forall i_1, j_1 \in \mathcal{G}_1, \quad \forall i_2 \in \mathcal{G}_2, \tag{7.4g}$$

$$g^{max}_{i_2} + \sigma_{i_1,i_2,i_1} = g^{max}_{j_2} + \sigma_{i_1,j_2,i_1}, \quad \forall i_1 \in \mathcal{G}_1, \qquad \forall i_2, j_2 \in \mathcal{G}_2. \tag{7.4h}$$

In Figure 7.1 we can see an example for the case where (7.4h) is not satisfied. In this example we can see that signal 2 is already finished performing the setup $\sigma_{2,1}$ while the setup $\sigma_{3,1}$ has not yet finished. Hence, at the moment that the setup $\sigma_{2,1}$ has finished, signal 1 cannot yet switch to green. Since the purpose of a maximum green time is to reduce the red times of another signal, signal 2 is red without purpose.

Further, we assume that inequalities (7.4i)–(7.4t) are satisfied for all $i_1 \in \mathcal{G}_1$ and for all $i_2 \in \mathcal{G}_2$. In Section 7.5.2 we show that we can always find a periodic trajectory satisfying the behavioral equations (7.3) if and only if (7.4i)–(7.4t) are satisfied for all $i_1 \in \mathcal{G}_1$ and for all $i_2 \in \mathcal{G}_2$. For more information see Section 7.5.2.

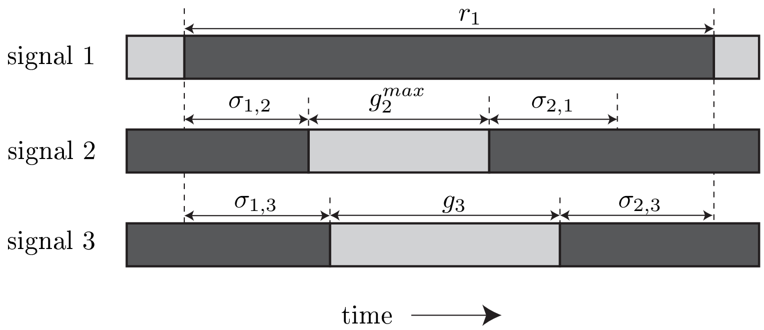


Figure 7.1: Situation where $g_2^{max} + \sigma_{1,2,1} > g_3^{max} + \sigma_{1,3,1}$. The dark gray rectangles visualize when a signal is red and the light gray rectangles visualize when a signal is green.

$$g_{i_1}^{max} \geq \frac{\sigma_{i_1,i_2,i_1}\rho_{i_1}}{1-\rho_{i_1}-\rho_{i_2}}, \tag{7.4i}$$
$$g_{i_1}^{max} \geq g_{i_1}^{min}, \tag{7.4j}$$
$$g_{i_1}^{max} \geq (g_{i_2}^{min} + \sigma_{i_1,i_2,i_1})\frac{\rho_{i_1}}{1-\rho_{i_1}}, \tag{7.4k}$$
$$g_{i_2}^{max} \geq \frac{\sigma_{i_1,i_2,i_1}\rho_{i_2}}{1-\rho_{i_1}-\rho_{i_2}}, \tag{7.4l}$$
$$g_{i_2}^{max} \geq g_{i_2}^{min}, \tag{7.4m}$$
$$g_{i_2}^{max} \geq (g_{i_1}^{min} + \sigma_{i_1,i_2,i_1})\frac{\rho_{i_2}}{1-\rho_{i_2}}, \tag{7.4n}$$
$$x_{i_1}^{max} \geq \lambda_{i_1}(\frac{\sigma_{i_1,i_2,i_1}\rho_{i_2}}{1-\rho_{i_1}-\rho_{i_2}} + \sigma_{i_1,i_2,i_1}), \tag{7.4o}$$
$$x_{i_1}^{max} \geq \lambda_{i_1}\frac{g_{i_1}^{min}\rho_{i_2} + \sigma_{i_1,i_2,i_1}}{1-\rho_{i_2}}, \tag{7.4p}$$
$$x_{i_1}^{max} \geq \lambda_{i_1}(g_{i_2}^{min} + \sigma_{i_1,i_2,i_1}), \tag{7.4q}$$
$$x_{i_2}^{max} \geq \lambda_{i_2}(\frac{\sigma_{i_1,i_2,i_1}\rho_{i_1}}{1-\rho_{i_1}-\rho_{i_2}} + \sigma_{i_1,i_2,i_1}), \tag{7.4r}$$
$$x_{i_2}^{max} \geq \lambda_{i_2}\frac{g_{i_2}^{min}\rho_{i_1} + \sigma_{i_1,i_2,i_1}}{1-\rho_{i_1}}, \tag{7.4s}$$
$$x_{i_2}^{max} \geq \lambda_{i_2}(g_{i_1}^{min} + \sigma_{i_1,i_2,i_1}). \tag{7.4t}$$

7.3 Properties of Optimal Trajectories

In section we prove that we can always find an optimal trajectory(minimizing (7.1)) that satisfies a few properties.

From Lemma 4.1 and Lemma 4.2 we know that there is always an optimal trajectory where we always use the highest possible departure rate during a green period of signal $i \in \mathcal{N}$:

$$
d_i(t) = \begin{cases} \mu_i & \text{if } x_i(t) > 0, \\ \lambda_i & \text{if } x_i(t) = 0. \end{cases}
$$

Lemma 7.1 *Without loss of generality it can be assumed that for an optimal trajectory in the behavior, it holds* $\forall k \geq 1$ *that:*

$$
g_{i_1}^k + \sigma_{i_1,i_2,i_1} = g_{j_1}^k + \sigma_{j_1,i_2,j_1}, \quad \forall i_1, j_1 \in \mathcal{G}_1, \quad \forall i_2 \in \mathcal{G}_2, \tag{7.5a}
$$

$$
g_{i_2}^k + \sigma_{i_1,i_2,i_1} = g_{j_2}^k + \sigma_{i_1,j_2,i_1}, \quad \forall i_1 \in \mathcal{G}_1, \qquad \forall i_2, j_2 \in \mathcal{G}_2. \tag{7.5b}
$$

Proof. In this proof we use the following notation:

$$
\overline{c} = \begin{cases} 2 & \text{if } c = 1, \\ 1 & \text{if } c = 2. \end{cases}
$$

Suppose that we are given a trajectory that satisfies $g_{i_c}^k + \sigma_{i_c,i_{\overline{c}},i_c} > g_{j_c}^k + \sigma_{j_c,i_{\overline{c}},j_c}$, $j_c, i_c \in \mathcal{G}_c$, $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$, $c = 1, 2$. Thus, the property given in this lemma is not satisfied. For this trajectory signal j_c switches to green at time $t_0 + \sigma_{i_{\overline{c}},j_c}$ and switches to red at time $t_f - \sigma_{j_c,i_{\overline{c}}}$ (see Figure 7.2).

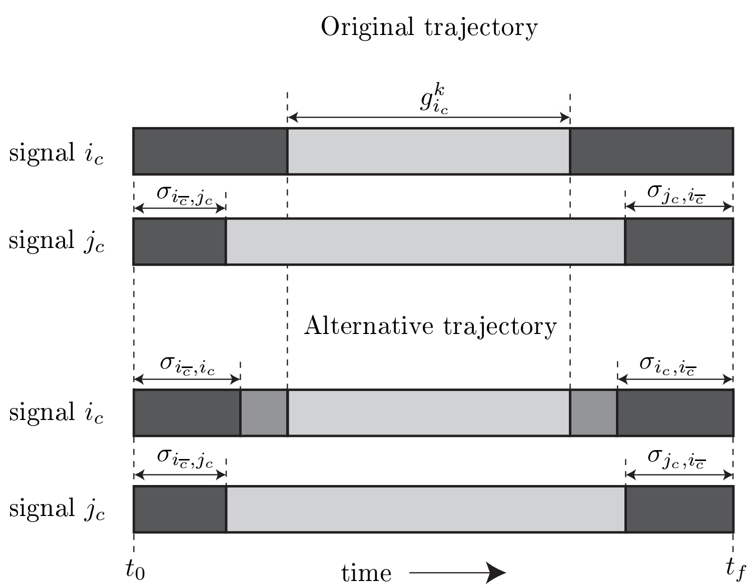


Figure 7.2: Visualization of Lemma 7.1. The dark gray rectangles visualize when a signal is red, the light gray rectangles visualize when a signal is green and the highest possible departure rate is used and the medium gray rectangles visualize when a signal is green and a departure rate equal to zero is used.

Consider the alternative trajectory, which only differs from the original trajectory for signal i_c and only differs from the original trajectory during the interval $[t_0, t_f]$. For this alternative trajectory, signal i_c switches to green at time $t_0 + \sigma_{i_{\overline{c}},i_c}$ and signal i_c switches to red at time $t_f - \sigma_{i_c,i_{\overline{c}}}$ (see Figure 7.2). The alternative trajectory uses a departure rate equal to zero at the times where signal i_c is green for the alternative trajectory and signal i_c is red for the original trajectory. The alternative trajectory uses the highest possible departure rate at the times where signal i_c is green for the alternative trajectory and signal i_c is also green for the original trajectory.

Hence, the evolution of the queue length of queue i_c is exactly the same for both trajectories. As a result the alternative trajectory satisfies the constraints on queue lengths whenever the original trajectory does. Further, from assumptions (10.1) and (10.2) we know that we satisfy behavioral equations (7.3g) and (7.3h). From (7.4g) and (7.4h) we know that the alternative trajectory satisfies the constraints on green time duration whenever the original trajectory does.

Since the evolution of the queue lengths is exactly the same for both trajectories, both trajectories result in the same costs (calculated via (7.1)).

Thus, whenever we are given a trajectory that does not satisfy the property given in this lemma, we can always give an alternative trajectory that does satisfy this property and that works at least as good. Hence, there must be an optimal trajectory that satisfies the property given in this lemma. ■

Lemma 7.2 *Without loss of generality it can be assumed that for an optimal trajectory in the behavior, a queue is always emptied during its green period and green periods always take equally long, i.e.* $g_i^k = g_i^{k+1}$, $\forall k \geq 1$.

Proof. The proof of this lemma is shown in Appendix C.3. The proof of this lemma is very similar to the proof of Lemma 4.3. ■

Lemma 7.3 *Without loss of generality it can be assumed that for an optimal trajectory in the behavior, a signal is green as long as possible during a red period of a conflicting signal:*

$$r_{i_2} = g_{i_1} + \sigma_{i_1,i_2,i_1}, \quad \forall i_1 \in \mathcal{G}_1, \quad \forall i_2 \in \mathcal{G}_2, \tag{7.6}$$

$$r_{i_1} = g_{i_2} + \sigma_{i_1,i_2,i_1}, \quad \forall i_1 \in \mathcal{G}_1, \quad \forall i_2 \in \mathcal{G}_2, \tag{7.7}$$

where, g_i *respectively* r_i *is the duration of all green times of signal* $i \in \mathcal{N}$ *and the duration of all red times of signal* $i \in \mathcal{N}$.

Proof. In this proof we use the following notation:

$$\overline{c} = \begin{cases} 2 & \text{if } c = 1, \\ 1 & \text{if } c = 2. \end{cases}$$

Suppose we are given a trajectory where the green times and red times of the different signals are given and denoted with g_i, $i \in \mathcal{N}$ and r_i, $i \in \mathcal{N}$ and that this trajectory does not satisfy the property given in this lemma. We can prove that there is always an alternative trajectory that does satisfy this property and that results in costs (calculated via (7.1)) that are not larger than the costs of the original trajectory. This alternative trajectory has the same green times as the original trajectory. The red times of this alternative trajectory are chosen such that the property given in this lemma is satisfied. We use r_i^{alt} for the red times of signal $i \in \mathcal{N}$ for the alternative trajectory. For the alternative trajectory we switch signal $i_1 \in \mathcal{G}_1$ to green exactly σ_{i_2,i_1} seconds after we switch signal $i_2 \in \mathcal{G}_2$ to red and we switch signal $i_2 \in \mathcal{G}_2$ to green exactly σ_{i_1,i_2} seconds after we switch signal $i_1 \in \mathcal{G}_1$ to red. Thus, we switch a signal to green as soon as its allowed. The alternative trajectory is shown in Figure 7.3.

Note that a red time of signal $i_c \in \mathcal{G}_c$, $c = 1, 2$ must satisfy $r_{i_c} \geq g_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ because otherwise constraint (7.3g) or (7.3h) is violated. Hence, it holds that $r_i^{alt} \leq r_i$, $\forall i \in \mathcal{N}$.

In Figure 7.4 we can see the queue length evolution of signal i for both original trajectory and the alternative trajectory. In this figure we use t_0^{alt} for the time at which a red period starts for the alternative trajectory and we use t_0^{org} for the time at which a red period starts for the original trajectory.

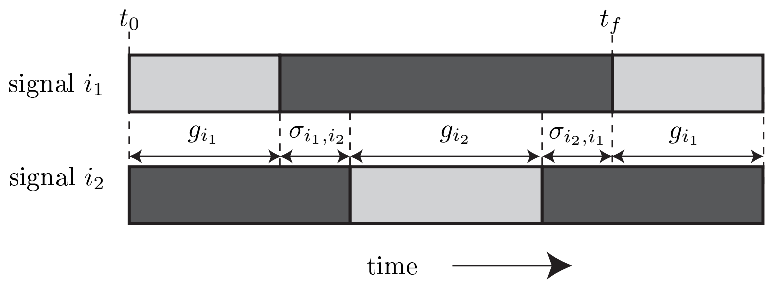


Figure 7.3: Visualization of the alternative trajectory. The dark gray rectangles visualize when a signal is red and the light gray rectangles visualize when a signal is green. The behavior in the interval $[t_0, t_f]$ repeats itself.

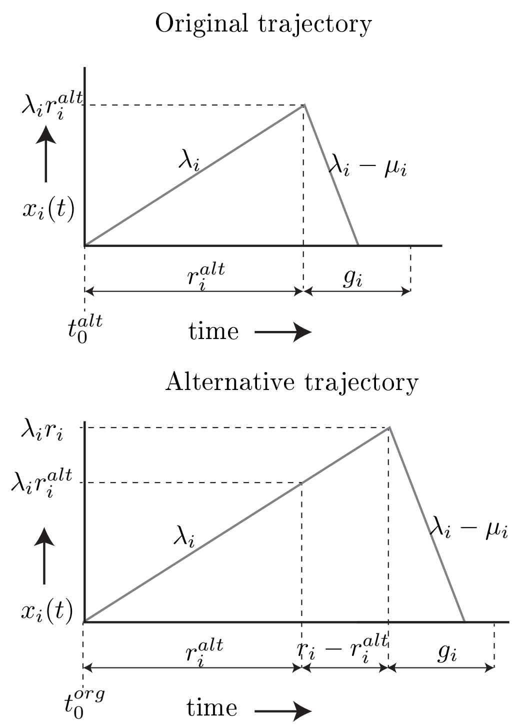


Figure 7.4: Queue length evolution for the original trajectory and the alternative trajectory. For the alternative trajectory the behavior in the interval $[t_0^{alt}, t_0^{alt} + r_i^{alt} + g_i]$ repeats itself. For the original trajectory the behavior in the interval $[t_0^{org}, t_0^{org} + r_i + g_i]$ repeats itself.

We introduce the following notation:

$J_{1,i}^{alt}$**:** The average costs related to signal i during a red period (during the interval $[t_0^{alt}, t_0^{alt} + r_i^{alt}]$) of the alternative trajectory.

$J_{1,i}^{org}$**:** The average costs related to signal i during the interval $[t_0^{org}, t_0^{org} + r_i^{alt}]$ of the original trajectory.

$J_{2,i}^{alt}$**:** The average costs related to signal i during a green period (during the interval $[t_0^{alt} + r_i^{alt}, t_0^{alt} + r_i^{alt} + g_i]$) of the alternative trajectory.

$J_{2,i}^{org}$**:** The average costs related to signal i during a green period (during the interval $[t_0^{org} + r_i, t_0^{org} + r_i + g_i]$) of the original trajectory.

$J_{3,i}^{org}$**:** The average costs for signal i during during the interval $[t_0^{org} + r_i^{alt}, t_0^{org} + r_i]$ of the original trajectory.

For the alternative trajectory the average costs related to signal i are equal to:

$$J_i^{alt} = \frac{r_i^{alt}}{r_i^{alt} + g_i} J_{1,i}^{alt} + \frac{g_i}{r_i^{alt} + g_i} J_{2,i}^{alt}.$$

We can see that the queue length evolution during $[t_0^{org}, t_0^{org} + r_i^{alt}]$ of the original trajectory is the same as the queue length evolution during the interval $[t_0^{alt}, t_0^{alt} + r_i^{alt}]$ of the alternative trajectory . Hence, $J_{1,i}^{alt} = J_{1,i}^{org}$. Furthermore, we can see from Figure 7.5 that $J_{2,i}^{alt} \leq J_{2,i}^{org}$ because f_i is strictly increasing. During the interval $[t_0^{org} + r_i^{alt}, t_0^{org} + r_i]$ the queue length of signal i for the original trajectory satisfies $x_i(t) \geq \lambda_i r_i^{alt}$. For the alternative trajectory the queue length (always) satisfies $x_i(t) \leq \lambda_i r_i^{alt}$. Thus, it holds that $J_{3,i}^{alt} \geq J_i^{alt}$ because f_i is strictly increasing.

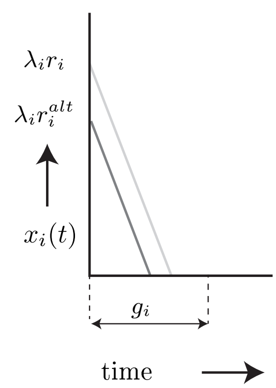


Figure 7.5: Queue length evolution during a green period of the original trajectory (visualized with a light gray line) and the queue length evolution during a green period of the alternative trajectory (visualized with a dark gray line).

Using this information we can derive that:

$$
\begin{aligned}
J_i^{org} &= \frac{r_i^{alt}}{r_i+g_i} J_{1,i}^{org} + \frac{g_i}{r_i+g_i} J_{2,i}^{org} + \frac{r_i - r_i^{alt}}{r_i+g_i} J_{2,i}^{org} \\
&= \frac{r_i^{alt}+g_i}{r_i+g_i} \left(\frac{r_i^{alt}}{r_i^{alt}+g_i} J_{1,i}^{org} + \frac{g_i}{r_i^{alt}+g_i} J_{2,i}^{org} \right) + \frac{r_i - r_i^{alt}}{r_i+g_i} J_{2,i}^{org} \\
&\geq \frac{r_i^{alt}+g_i}{r_i+g_i} \left(\frac{r_i^{alt}}{r_i^{alt}+g_i} J_{1,i}^{alt} + \frac{g_i}{r_i^{alt}+g_i} J_{2,i}^{alt} \right) + \frac{r_i - r_i^{alt}}{r_i+g_i} J_i^{alt} \\
&\geq \frac{r_i^{alt}+g_i}{r_i+g_i} J_i^{alt} + \frac{r_i - r_i^{alt}}{r_i+g_i} J_i^{alt} \\
&\geq J_i^{alt}.
\end{aligned}
$$

Hence, the costs related to each signal $i \in \mathcal{N}$ are not larger for the alternative trajectory than for the original trajectory.

Furthermore, we can see that the alternative trajectory is feasible because:

- Assuming (10.1) and (10.2) there is exactly enough time to perform the setups and thus to satisfy constraint (7.3g) and constraint (7.3h).
- For the original trajectory the queue length of signal $i \in \mathcal{N}$ is at most $\lambda_i r_i$ and for the alternative trajectory the queue length of signal i is at most $\lambda_i r_i^{alt} \leq \lambda_i r_i$. Hence, when the original trajectory satisfies the constraints on maximum queue lengths the alternative trajectory does as well because the queue length of signal i_c increases with the duration of a red period.
- both trajectories have the same green times. Hence, when the original trajectory satisfies the constraints on maximum queue lengths the alternative trajectory does as well.

■

7.4 Shape of the Periodic Optimal Trajectory

Using lemmas 4.1, 4.2, 7.1, 7.2 and 7.3 we can find the following corollary for the intersection with two signal groups.

Corollary 7.4 *For an intersection with two signal groups and assumptions given in Section 7.2.3 we can without loss of generality assume that an optimal trajectory (minimizing (7.1)) has the periodic shape shown in the (x_{i_1}, x_{i_2})-plane, $i_1 \in \mathcal{G}_1$,$i_2 \in \mathcal{G}_2$ that is shown in Figure 7.6. This periodic shape in the (x_{i_1}, x_{i_2})-plane consists of the following phases (these phases repeat periodically):*

phase 1 *Signal i_1 is green and $d_{i_1}(t) = \mu_{i_1}$ until queue i_1 is empty.*

phase 2 *Signal 1 is green and $d_{i_1}(t) = \lambda_{i_1}$.*

phase 3 *perform a setup to signal i_2, i.e. switch signal i_2 to red and keep both signals red for a period equal to σ_{i_1,i_2}*

phase 4 *Signal i_2 is green and $d_{i_2}(t) = \mu_{i_2}$ until queue i_2 is empty.*

phase 5 *Signal i_2 is green and $d_{i_2}(t) = \lambda_{i_2}$.*

phase 6 *perform a setup to signal i_1, i.e. switch signal i_2 to red and keep both signals red for a period equal to σ_{i_2,i_1}*

Since all green periods of a signal have the same duration and all red periods of a signal have the same duration, we use:

$$
\begin{aligned}
g_i &= g_i^k, \quad i \in \mathcal{N}, k \geq 1, \\
r_i &= r_i^k, \quad i \in \mathcal{N}, k \geq 1, \\
g_i^\lambda &= g_i^{\lambda,k}, i \in \mathcal{N}, k \geq 1, \\
g_i^\mu &= g_i^{\mu,k}, i \in \mathcal{N}, k \geq 1.
\end{aligned}
\tag{7.8}
$$

Step 2 and phase 4 are the so called slow modes and may have a duration equal to zero. We call this periodically repeated sequence a cycle. On the left hand side of Figure 4.7, this cycle is plotted in the (x_{i_1}, x_{i_2})*-plane. The right hand side graphs show the queue lengths over time, with the slopes annotated to them. The duration of a cycle is denoted with c and is equal to* $g_{i_1} + g_{i_2} + \sigma_{i_1,i_2,i_1}$.

The green times are related according to:

$$g_{i_1} + \sigma_{i_1,i_2,i_1} = g_{j_1} + \sigma_{j_1,i_2,j_1}, \quad \forall i_1, j_1 \in \mathcal{G}_1, \quad \forall i_2 \in \mathcal{G}_2, \tag{7.9a}$$

$$g_{i_2} + \sigma_{i_1,i_2,i_1} = g_{j_2} + \sigma_{i_1,j_2,i_1}, \quad \forall i_1 \in \mathcal{G}_1, \qquad \forall i_2, j_2 \in \mathcal{G}_2. \tag{7.9b}$$

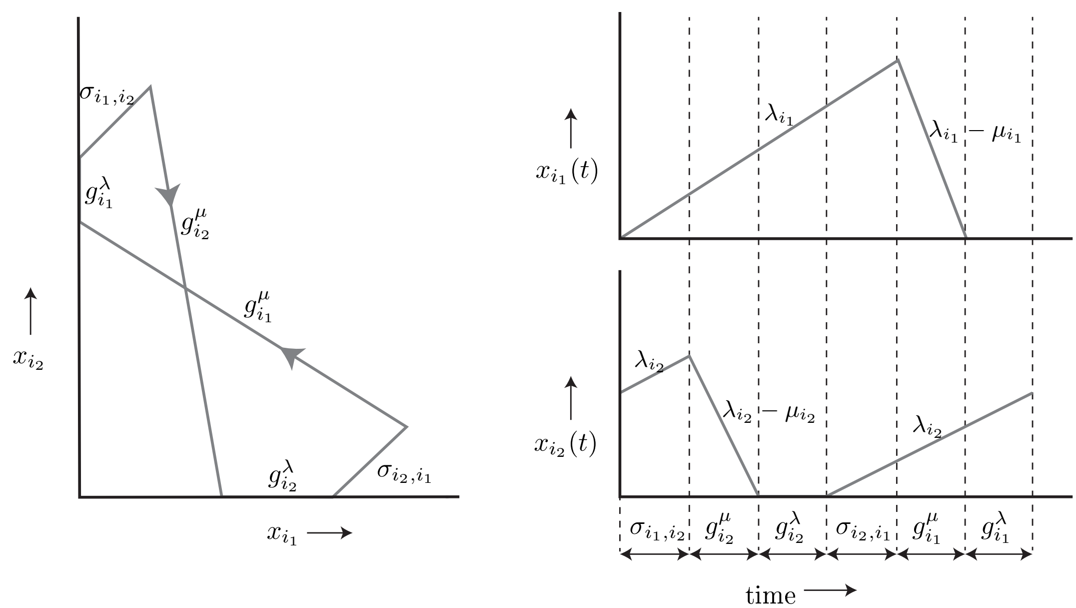


Figure 7.6: Shape of the periodic optimal trajectory. Left: periodic optimal trajectory in the (i_1, i_2)-plane. Right: queue lengths over time.

The traffic that arrives during a red period of signal $i \in \mathcal{N}$ can (precisely) depart during g_i^μ. Hence we can find:

$$g_{i_1}^\mu = \frac{\rho_{i_1}}{1-\rho_{i_1}}(g_{i_2} + \sigma_{i_1,i_2,i_1}), \quad i_{i_1} \in \mathcal{G}_{i_1}, \quad i_{i_2} \in \mathcal{G}_{i_2}, \tag{7.10a}$$

$$g_{i_2}^\mu = \frac{\rho_{i_2}}{1-\rho_{i_2}}(g_{i_1} + \sigma_{i_1,i_2,i_1}), \quad i_{i_1} \in \mathcal{G}_{i_1}, \quad i_{i_2} \in \mathcal{G}_{i_2}. \tag{7.10b}$$

7.5 Periodic Optimal Trajectory

From Corollary 7.4 we know that w.l.o.g. we can assume that optimal trajectories are periodic. For these periodic optimal trajectories a queue is emptied during each green period. Using these properties we can rewrite the behavioral equations (7.3). In Section 7.5.1 we give the rewritten form of these behavioral equations. In Section 7.5.3 we elaborate on the solutions of this optimization problem.

7.5.1 Optimization Problem

We want to minimize the linear cost function. From now on we use a more specific form of the cost function J: the linear cost function shown in (7.11).

$$J = \frac{1}{c}\int_0^c \sum_{i\in\mathcal{N}} w_i x_i(s) ds. \tag{7.11}$$

From the right side of Figure 7.6 and using (7.9) we can obtain the following expression for the linear cost function for an intersection with two signal groups.

$$\begin{aligned} J &= \frac{1}{c}\int_0^c \sum_{i\in\mathcal{N}} w_i x_i(s) ds, \\ &= \sum_{i\in\mathcal{G}_1} \frac{w_i(\sigma_{i,N,i} + g_N + g_i^\mu)(\sigma_{i,N,i} + g_N)\lambda_i}{2(\sigma_{1,N,1} + g_1 + g_N)}, \\ &+ \sum_{i\in\mathcal{G}_2} \frac{w_i(\sigma_{1,i,1} + g_N + g_i^\mu)(\sigma_{1,i,1} + g_1)\lambda_i}{2(\sigma_{1,N,1} + g_1 + g_N)}. \end{aligned} \tag{7.12}$$

Recall that by definition signal 1 is element of signal group 1 and signal N is element of signal group 2. Using (7.10) we can obtain:

$$\min_{g_1,g_N} \frac{\sum_{i\in\mathcal{G}_1} k_i(\sigma_{i,N,i} + g_N)^2 + \sum_{i\in\mathcal{G}_2} k_i(\sigma_{i,1,i} + g_1)^2}{g_1 + g_N + \sigma_{1,N,1}}, \tag{7.13}$$

where

$$\beta_i = \frac{\lambda_i w_i}{2(1-\rho_i)}, \quad i \in \mathcal{N}.$$

The green time of a signal must be large enough for the traffic, that arrives during a red period, to depart. Otherwise, we would not get the periodic optimal trajectory from Corollary 7.4 because a queue length goes to infinity.

$$g_i \geq \frac{\rho_i}{1-\rho_i} r_i, \quad \forall i \in \mathcal{N}. \tag{7.14a}$$

The maximum queue length of a signal must be larger than the amount of traffic that arrives during a red period:

$$r_i \leq \frac{x_i^{max}}{\lambda_i}, \quad \forall i \in \mathcal{N}. \tag{7.14b}$$

Each green time must exceed the minimum green time:

$$g_i \geq g_i^{min}, \quad \forall i \in \mathcal{N}. \tag{7.14c}$$

Each green time may not exceed the maximum green time:

$$g_i \geq g_i^{max}, \quad \forall i \in \mathcal{N}. \tag{7.14d}$$

Using the relations between green times given in (7.9) we can rewrite (7.14) to:

$$\begin{aligned}
g_1 &\geq \max_{i \in \mathcal{G}_1} \frac{\rho_i g_N + \sigma_{i,N,i}}{1-\rho_i} - \sigma_{1,N,1}, && (7.15a)\\
g_N &\geq \max_{i \in \mathcal{G}_2} \frac{\rho_i g_1 + \sigma_{1,i,1}}{1-\rho_i} - \sigma_{1,N,1}, && (7.15b)\\
g_1 &\leq \min_{i \in \mathcal{G}_2} \frac{x_i^{max}}{\lambda_i} - \sigma_{1,i,1}, && (7.15c)\\
g_N &\leq \min_{i \in \mathcal{G}_1} \frac{x_i^{max}}{\lambda_i} - \sigma_{i,N,i}, && (7.15d)\\
g_1 &\geq \max_{i \in \mathcal{G}_1} g_i^{min} + \sigma_{i,N,i} - \sigma_{1,N,1}, && (7.15e)\\
g_N &\geq \max_{i \in \mathcal{G}_2} g_i^{min} + \sigma_{1,i,1} - \sigma_{1,N,1}, && (7.15f)\\
g_1 &\leq g_1^{max}, && (7.15g)\\
g_N &\leq g_N^{max}. && (7.15h)
\end{aligned}$$

We want to find values for g_1 and g_N that satisfy constraints (7.15) and minimize the linear cost function (7.13). From these values for g_1 and g_N we can derive the green times of all other signals via (7.9).

7.5.2 Existence of a Solution

A solution to the optimization problem with constraints (7.15) is only possible if we can find values for g_{i_1}, $\forall i_1 \in \mathcal{G}_1$ and g_{i_2}, $\forall i_2 \in \mathcal{G}_2$ satisfying inequalities (7.14). We can find values for g_{i_1}, $\forall i_1 \in \mathcal{G}_1$ and g_{i_2}, $\forall i_2 \in \mathcal{G}_2$ satisfying inequalities (7.14) if and only if the following inequalities are satisfied for all signals $i_1 \in \mathcal{G}_1$ and for all signals $i_2 \in \mathcal{G}_2$:

$$g_{i_1}^{max} \geq \frac{\sigma_{i_1,i_2,i_1}\rho_{i_1}}{1-\rho_{i_1}-\rho_{i_2}}, \tag{7.16a}$$

$$g_{i_1}^{max} \geq g_{i_1}^{min}, \tag{7.16b}$$

$$g_{i_1}^{max} \geq (g_{i_2}^{min}+\sigma_{i_1,i_2,i_1})\frac{\rho_{i_1}}{1-\rho_{i_1}}, \tag{7.16c}$$

$$g_{i_2}^{max} \geq \frac{\sigma_{i_1,i_2,i_1}\rho_{i_2}}{1-\rho_{i_1}-\rho_{i_2}}, \tag{7.16d}$$

$$g_{i_2}^{max} \geq g_{i_2}^{min}, \tag{7.16e}$$

$$g_{i_2}^{max} \geq (g_{i_1}^{min}+\sigma_{i_1,i_2,i_1})\frac{\rho_{i_2}}{1-\rho_{i_2}}, \tag{7.16f}$$

$$x_{i_1}^{max} \geq \lambda_{i_1}(\frac{\sigma_{i_1,i_2,i_1}\rho_{i_2}}{1-\rho_{i_1}-\rho_{i_2}}+\sigma_{i_1,i_2,i_1}), \tag{7.16g}$$

$$x_{i_1}^{max} \geq \lambda_{i_1}\frac{g_{i_1}^{min}\rho_{i_2}+\sigma_{i_1,i_2,i_1}}{1-\rho_{i_2}}, \tag{7.16h}$$

$$x_{i_1}^{max} \geq \lambda_{i_1}(g_{i_2}^{min}+\sigma_{i_1,i_2,i_1}), \tag{7.16i}$$

$$x_{i_2}^{max} \geq \lambda_{i_2}(\frac{\sigma_{i_1,i_2,i_1}\rho_{i_1}}{1-\rho_{i_1}-\rho_{i_2}}+\sigma_{i_1,i_2,i_1}), \tag{7.16j}$$

$$x_{i_2}^{max} \geq \lambda_{i_2}\frac{g_{i_2}^{min}\rho_{i_1}+\sigma_{i_1,i_2,i_1}}{1-\rho_{i_1}}, \tag{7.16k}$$

$$x_{i_2}^{max} \geq \lambda_{i_2}(g_{i_1}^{min}+\sigma_{i_1,i_2,i_1}). \tag{7.16l}$$

The inequalities in (7.16) can be interpreted as follows. Whenever, a periodic trajectory satisfies $g_{i_1} = \frac{\sigma_{i_1,i_2,i_1}\rho_{i_1}}{1-\rho_{i_1}-\rho_{i_2}}$ and $g_{i_2} = \frac{\sigma_{i_1,i_2,i_1}\rho_{i_2}}{1-\rho_{i_1}-\rho_{i_2}}$ (and we let traffic depart at the maximum departure rate) we get a pure bow tie curve in the $(i_1, i_2)-$plane (see Figure 7.7). The green times $g_{i_1} = \frac{\sigma_{i_1,i_2,i_1}\rho_{i_1}}{1-\rho_{i_1}-\rho_{i_2}}$ and $g_{i_2} = \frac{\sigma_{i_1,i_2,i_1}\rho_{i_2}}{1-\rho_{i_1}-\rho_{i_2}}$ are the smallest green times for which all traffic that arrives during a cycle at signal i_1 and signal i_2 can depart during a cycle.

The inequalities (8.1h),(8.1k),(8.1n) and (8.1q) make sure that this pure bow tie curve in the (i_1, i_2)-plane does not violate the maximum green times $g_{i_1}^{max}$ and $g_{i_2}^{max}$ and it does not violate the maximum queue lengths $x_{i_1}^{max}$ and $x_{i_2}^{max}$.

However, a pure bow tie curve in the (i_1, i_2)-plane might violate a constraint on the minimum green time duration. The inequalities (8.1i),(8.1l),(8.1o) and (8.1r) make sure that there exists a periodic trajectory where $g_{i_1} = g_{i_1}^{min}$, such that the maximum green times $g_{i_1}^{max}$ and $g_{i_2}^{max}$ and the maximum queue lengths $x_{i_1}^{max}$ and $x_{i_2}^{max}$ are not violated. Similarly, inequalities (8.1j),(8.1m),(8.1p) and (8.1s) make sure that the maximum green times $g_{i_1}^{max}$ and $g_{i_2}^{max}$ and the maximum queue lengths $x_{i_1}^{max}$ and $x_{i_2}^{max}$ are not violated.

7.5.3 Solution

In this section we present the solution to the optimization problem for two cases. First we consider an intersection where $\mathcal{G}_1 = \{1\}$, $\mathcal{G}_2 = \{2, 3\}$ and $\sigma_{1,2,1} = \sigma_{1,3,1}$. Subsequently we consider an intersection where $\mathcal{G}_1 = \{1\}$, $\mathcal{G}_2 = \{2, 3\}$ and $\sigma_{1,2,1} \neq \sigma_{1,3,1}$.

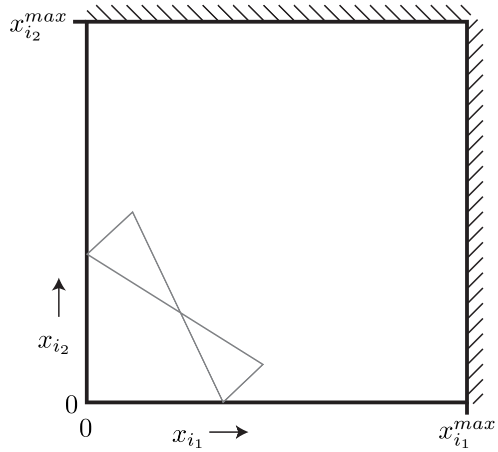


Figure 7.7: Pure bow tie curve in the (i_1, i_2) −plane

Equal Setup Times

In this section we consider an intersection where $\mathcal{G}_1 = \{1\}$, $\mathcal{G}_2 = \{2, 3\}$ and $\sigma_{1,2,1} = \sigma_{1,3,1}$. From (7.9) we know that in this case $g_2 = g_3$. For this intersections the objective function (7.13) reduces to:

$$J = \frac{1}{c}\int_0^c \sum_{i\in\mathcal{N}} w_i x_i(s)ds = \frac{\beta_1(\sigma_{1,3,1} + g_3) + (\beta_2 + \beta_3)(\sigma_{1,3,1} + g_1)^2}{\sigma_{1,3,1} + g_1 + g_3}, \tag{7.17}$$

where

$$\beta_i = \frac{\lambda_i w_i}{2(1-\rho_i)}, \quad i \in \mathcal{N}.$$

Further, the constraints (7.15) reduce to:

$$\begin{aligned}
g_1 &\geq \frac{\rho_1}{1-\rho_1}(\sigma_{1,3,1} + g_3),\\
g_3 &\geq \frac{\max\limits_{i=2,3}\rho_i}{1-\max\limits_{i=2,3}\rho_i}(\sigma_{1,3,1} + g_1),\\
g_1 &\leq \min_{i=2,3}\frac{x_i}{\lambda_i} - \sigma_{1,3,1},\\
g_3 &\leq \frac{x_1}{\lambda_1} - \sigma_{1,3,1},\\
g_1 &\geq g_1^{min},\\
g_3 &\geq \min\{g_2^{min}, g_3^{min}\},\\
g_1 &\leq g_1^{max},\\
g_3 &\leq g_3^{max}.
\end{aligned}$$

This optimization problem can be solved analytically (see Appendix C.1). The periodic optimal trajectory can have 0, 1 or 2 slow modes. For more information see Appendix C.1. The optimization problem with objective function (7.17) and constraints (7.18) is related to the optimization problem in C.1 as follows.

Whenever $(\beta_2 + \beta_3) \le \beta_1$, the two optimization problems are related according to:

$$
\begin{aligned}
k &= \frac{(\beta_2 + \beta_3)}{\beta_1}, \\
y_1 &= \frac{g_1}{\sigma_{1,3,1}}, \\
y_2 &= \frac{g_3}{\sigma_{1,3,1}}, \\
y_1^{min} &= \frac{g_1^{min}}{\sigma_{1,3,1}}, \\
y_2^{min} &= \frac{\max\{g_2^{min}, g_3^{min}\}}{\sigma_{1,3,1}}, \\
y_1^{max} &= \frac{\min\{g_1^{max}, \min_{i=2,3} \frac{x_i^{max}}{\lambda_i} - \sigma_{1,3,1}\}}{\sigma_{1,3,1}}, \\
y_2^{max} &= \frac{\min\{g_2^{max}, \frac{x_1^{max}}{\lambda_1} - \sigma_{1,3,1}\}}{\sigma_{1,3,1}}, \\
\alpha_1 &= \rho_1, \\
\alpha_2 &= \max_{i=2,3} \rho_i.
\end{aligned}
$$

Whenever $(\beta_2 + \beta_3) \ge \beta_1$, the two optimization problems are related according to:

$$
\begin{aligned}
k &= \frac{\beta_1}{(\beta_2 + \beta_3)}, \\
y_1 &= \frac{g_3}{\sigma_{1,3,1}}, \\
y_2 &= \frac{g_1}{\sigma_{1,3,1}}, \\
y_1^{min} &= \frac{\max\{g_2^{min}, g_3^{min}\}}{\sigma_{1,3,1}}, \\
y_2^{min} &= \frac{g_1^{min}}{\sigma_{1,3,1}}, \\
y_1^{max} &= \frac{\min\{g_2^{max}, \frac{x_1^{max}}{\lambda_1} - \sigma_{1,3,1}\}}{\sigma_{1,3,1}}, \\
y_2^{max} &= \frac{\min\{g_1^{max}, \min_{i=2,3} \frac{x_i^{max}}{\lambda_i} - \sigma_{1,3,1}\}}{\sigma_{1,3,1}}, \\
\alpha_1 &= \max_{i=2,3} \rho_i, \\
\alpha_2 &= \rho_1.
\end{aligned}
$$

Generalization: using the assumptions in Section 7.2.3, the analytical solution in Appendix C.1 can be used for any intersection with two conflict groups where:

$$\sigma_{1,N,1} = \sigma_{i_1,N,i_1}, \forall i_1 \in \mathcal{G}_1,$$
$$\sigma_{1,N,1} = \sigma_{i_2,1,i_2}, \ \forall i_2 \in \mathcal{G}_2.$$

For this class of intersections we can find the following expressions for the optimization problem in Appendix C.1. Without loss of generality we assume that $0 < k \leq 1$.

$$\begin{aligned}
k &= \frac{\sum\limits_{i \in \mathcal{G}_2} \beta_i}{\sum\limits_{i \in \mathcal{G}_1} \beta i}, \\
y_1 &= \frac{g_1}{\sigma_{1,N,1}}, \\
y_2 &= \frac{g_N}{\sigma_{1,N,1}}, \\
y_1^{min} &= \frac{\max\limits_{i \in \mathcal{G}_1} g_i^{min}}{\sigma_{1,N,1}}, \\
y_2^{min} &= \frac{\max\limits_{i \in \mathcal{G}_2} g_i^{min}}{\sigma_{1,N,1}}, \\
y_1^{max} &= \frac{\min\{g_1^{max}, \min\limits_{i \in \mathcal{G}_2} \frac{x_i}{\lambda_i} - \sigma_{1,N,1}\}}{\sigma_{1,N,1}}, \\
y_2^{max} &= \frac{\min\{g_N^{max}, \min\limits_{i \in \mathcal{G}_1} \frac{x_i^{max}}{\lambda_i} - \sigma_{1,N,1}\}}{\sigma_{1,N,1}}, \\
\alpha_1 &= \max_{i \in \mathcal{G}_1} \rho_i, \\
\alpha_2 &= \max_{i \in \mathcal{G}_2} \rho_i.
\end{aligned}$$

Unequal Setup Times

In this section we consider an intersection where $\mathcal{G}_1 = \{1\}$, $\mathcal{G}_2 = \{2,3\}$ and $\sigma_{1,2,1} \neq \sigma_{1,3,1}$. From (7.9) we know that $g_2 + \sigma_{1,2,1} = g_3 + \sigma_{1,3,1}$. For this intersection the objective function (7.13) reduces to:

$$J = \frac{1}{c} \int_0^c \sum_{i \in \mathcal{N}} w_i x_i(s) ds = \frac{\beta_1(\sigma_{1,3,1} + g_3)^2 + \beta_2(\sigma_{1,2,1} + g_1)^2 + \beta_3(\sigma_{1,3,1} + g_1)^2}{\sigma_{1,3,1} + g_1 + g_3}, \tag{7.19}$$

where

$$\beta_i = \frac{\lambda_i w_i}{2(1 - \rho_i)}, \quad i \in \mathcal{N}.$$

Further, the constraints (7.15) reduce to (7.20). Note that (7.20b) and (7.20c) both follow from (7.15d).

$$g_1 \geq \frac{\rho_1}{1-\rho_1}(\sigma_{1,3,1} + g_3), \tag{7.20a}$$
$$g_3 \geq \frac{\rho_3}{1-\rho_3}(\sigma_{1,3,1} + g_1), \tag{7.20b}$$
$$g_3 \geq \frac{\rho_2 g_1 + \sigma_{1,2,1}}{1-\rho_2} - \sigma_{1,3,1}, \tag{7.20c}$$
$$g_1 \leq \min_{i=2,3} \frac{x_i}{\lambda_i} - \sigma_{1,i,1}, \tag{7.20d}$$
$$g_3 \leq \frac{x_1}{\lambda_1} - \sigma_{1,3,1}, \tag{7.20e}$$
$$g_1 \geq g_1^{min}, \tag{7.20f}$$
$$g_3 \geq \min\{g_2^{min} + \sigma_{1,2,1} - \sigma_{1,3,1}, g_3^{min}\}, \tag{7.20g}$$
$$g_1 \leq g_1^{max}, \tag{7.20h}$$
$$g_3 \leq g_3^{max}. \tag{7.20i}$$

Assuming (with loss of generality) that $\beta_1 \geq (\beta_2 + \beta_3)$ and assuming w.l.o.g. that $\sigma_{1,3,1} \geq \sigma_{1,2,1}$ this optimization problem is solved analytically (see Appendix C.2). The periodic optimal trajectory can have 0, 1 or 2 slow modes. For more information see Appendix C.2. The optimization problem with objective function (7.19) and constraints (7.20) is related to the optimization problem in Appendix C.2 as follows.

$$\begin{aligned}
k_1 &= \frac{\beta_2}{\beta_1}, \\
k_2 &= \frac{\beta_3}{\beta_1}, \\
k_3 &= \frac{\sigma_{1,2,1}}{\sigma_{1,3,1}}, \\
y_1 &= \frac{g_1}{\sigma_{1,3,1}}, \\
y_2 &= \frac{g_3}{\sigma_{1,3,1}}, \\
y_1^{min} &= \frac{g_1^{min}}{\sigma_{1,3,1}} \\
y_2^{min} &= \frac{\max_{i=2,3} g_i^{min} + \sigma_{1,i,1}}{\sigma_{1,3,1}} - 1, \\
y_1^{max} &= \frac{\min\{g_1^{max}, \min_{i=2,3} \frac{x_i}{\lambda_i} + \sigma_{1,i,1} - \sigma_{1,3,1}\}}{\sigma_{1,3,1}}, \\
y_2^{max} &= \frac{\min\{g_N^{max}, \frac{x_1^{max}}{\lambda_1} - \sigma_{1,3,1}\}}{\sigma_{1,3,1}}, \\
\alpha_1 &= \rho_1, \\
\alpha_2 &= \rho_2, \\
\alpha_3 &= \rho_3.
\end{aligned}$$

Generalization: using the assumptions in Section 7.2.3, the analytical solution in Appendix C.2 can be used for the class of intersections satisfying the following properties:

- All signals $i_1 \in \mathcal{G}_1$ have the same setup time σ_{i_1,N,i_1}, i.e. $\sigma_{1,N,1} = \sigma_{i_1,N,i_1}, \forall i_1 \in \mathcal{G}_1$.
- Each signal $i \in \mathcal{G}_2$ can be partitioned into one of two sets $\mathcal{B}_1$ or $\mathcal{B}_2$. All signals $i \in \mathcal{B}_1$ have the same setup time $\sigma_{1,i,1} > 0$ which we denote with $\sigma_{\mathcal{B}_1}$. All signals $i \in \mathcal{B}_2$ have the same setup time $\sigma_{1,i,1} > 0$ which we denote with $\sigma_{\mathcal{B}_2}$.
- $\sum\limits_{i\in\mathcal{G}_1} \beta_i \geq \sum\limits_{i\in\mathcal{G}_2} \beta_i$.

For this class of intersections we can find the following expressions for the optimization problem in Appendix C.2. Without loss of generality we assume that $0 < k_3 \leq 1$ and w.l.o.g. we assume that signal N is partitioned in $\mathcal{B}_2$, i.e. $N \in \mathcal{B}_2$.

$$
\begin{aligned}
k_1 &= \frac{\sum\limits_{i\in\mathcal{B}_1} \beta_i}{\sum\limits_{i\in\mathcal{G}_1} \beta_i},\\
k_2 &= \frac{\sum\limits_{i\in\mathcal{B}_2} \beta_i}{\sum\limits_{i\in\mathcal{G}_1} \beta_i},\\
k_3 &= \frac{\sigma_{\mathcal{B}_1}}{\sigma_{\mathcal{B}_2}},\\
y_1 &= \frac{g_1}{\sigma_{1,N,1}},\\
y_2 &= \frac{g_N}{\sigma_{1,N,1}},\\
y_1^{min} &= \frac{\max\limits_{i\in\mathcal{G}_1} g_i^{min} + \sigma_{i,N,i}}{\sigma_{1,N,1}} - 1,\\
y_2^{min} &= \frac{\max\limits_{i\in\mathcal{G}_2} g_i^{min} + \sigma_{1,i,1}}{\sigma_{1,N,1}} - 1,\\
y_1^{max} &= \frac{\min\{g_1^{max}, \min\limits_{i\in\mathcal{G}_2} \frac{x_i}{\lambda_i} + \sigma_{1,i,1} - \sigma_{1,N,1}\}}{\sigma_{1,3,1}},\\
y_2^{max} &= \frac{\min\{g_N^{max}, \min\limits_{i\in\mathcal{G}_1} \frac{x_i}{\lambda_i} + \sigma_{i,N,i} - \sigma_{1,N,1}\}}{\sigma_{1,3,1}},\\
\alpha_1 &= \max_{i\in\mathcal{G}_1} \rho_i,\\
\alpha_2 &= \max_{i\in\mathcal{B}_1} \rho_i,\\
\alpha_3 &= \max_{i\in\mathcal{B}_2} \rho_i.
\end{aligned}
\tag{7.21}
$$

7.5.4 Fixed Order and Optimality

In this chapter we have only considered signals where we serve the signals in a fixed order; we alternate between serving all signals in $\mathcal{G}_1$ and serving all signals in $\mathcal{G}_2$. In practice, often signals are served in a fixed order. Some of the vehicles, cyclists and pedestrians already start to accelerate when they expect their signal to switch to green. When the order in which these signals are served changes, these expectations are likely to be wrong and can result in unsafe situations.

Using an example we show that trajectories that do not serve signals in a fixed order might results in a lower value for the cost function (7.1).

Example 7.5.1 *Consider an intersection with two signal groups: $\mathcal{G}_1 = \{1\}$ and $\mathcal{G}_2 = \{2,3\}$. We are given the following information about the intersection.*

$\lambda_1 = 0.3$ *vehicles per second,*	$\lambda_2 = 0.1$ *vehicles per second,*	$\lambda_3 = 0.001$ *vehicles per second*
$\mu_1 = 0.5$ *vehicles per second,*	$\mu_2 = 0.5$ *vehicles per second,*	$\mu_3 = 0.5$ *vehicles per second,*
$\rho_1 = 0.6$	$\rho_2 = 0.2,$	$\rho_3 = 0.002,$
$\sigma_{1,2} = 2.5$ *seconds,*	$\sigma_{2,1} = 2.5$ *seconds,*	
$\sigma_{1,3} = 20$ *seconds,*	$\sigma_{3,1} = 20$ *seconds.*	

We do not impose restrictions on minimum green times, maximum green times and maximum queue lengths. We consider the linear weight function in (7.11) where $w_1 = w_2 = w_3 = 1$. Thus, we like to minimize the average delay of a road user at the intersection (see Section 7.1.1).

For a fixed order, signal 2 and signal 3 are both served during every red period of signal 1. The green times of the optimal trajectory with a fixed order can be obtained by solving the optimization problem with objective function (7.20) and constraints (7.19). We can obtain the following green times:

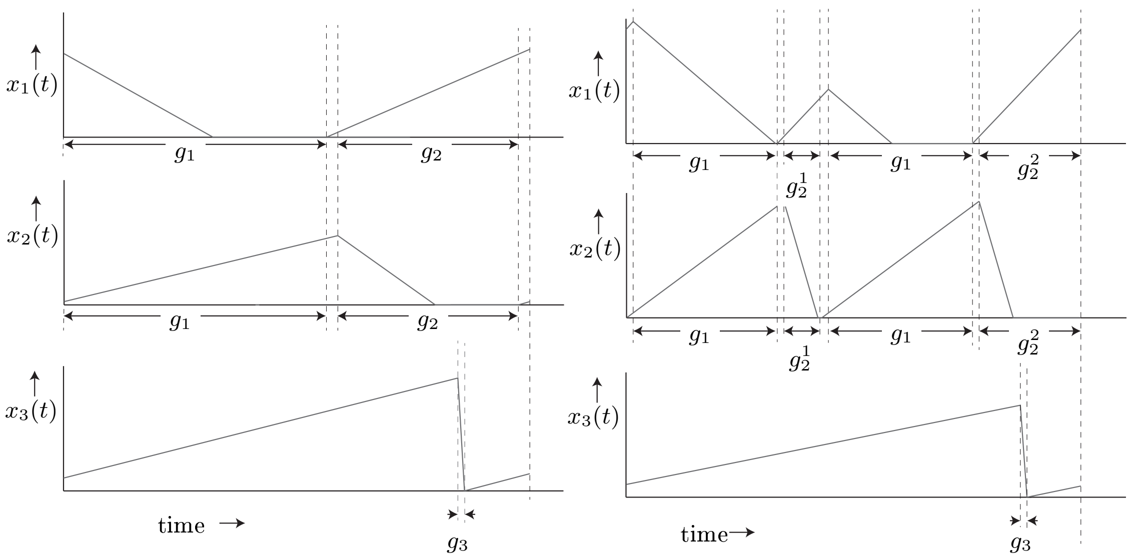
$$\begin{aligned} g_1 &= 63.6275 \textit{ seconds}, \\ g_2 &= 35.2077 \textit{ seconds}, \\ g_3 &= 0.2077 \textit{ seconds}. \end{aligned}$$

From these green times we can calculate the average delay of a vehicle via (7.19) and (7.2), which is $J = 21.7587$ seconds. This periodic trajectory is shown in Figure 7.8a.

However, in Figure 7.8b we show a trajectory where signal 2 is served twice as often as signal 3. The green times shown in this figure are:

$$\begin{aligned} g_1 &= 61 \textit{ seconds}, \\ g_2^1 &= 16.5 \textit{ seconds}, \\ g_2^2 &= 35.5 \textit{ seconds}, \\ g_3 &= 0.5 \textit{ seconds}. \end{aligned} \tag{7.22}$$

This trajectory reduces the value for the average delay of a vehicle to 18.294 seconds. This trajectory works better for this example because the setup time $\sigma_{1,3,1}$ is large. As a result whenever signal 3 is served, signal 1 has to wait very long until it is served again. Further, the arrival rate at signal 3 is very small. As a result, it is better to sometimes skip serving signal 3.



(a) Periodic optimal trajectory when the signals are served in a fixed order.

(b) A trajectory where the signals are not served in a fixed order.

Figure 7.8: Trajectories that do not serve the signals in a fixed order might result in a lower value for the cost function (7.1).

Chapter 8

Regulation: An Intersection with Two Signal Groups

In the previous chapter we considered the trajectory optimization problem. We showed how to derive periodic optimal trajectories for an intersection with two signal groups. To obtain these desired trajectories we assumed deterministic arrivals and deterministic departures. However, due to for example stochastic arrivals we may deviate from the desired trajectory. The second problem discussed in this thesis is regulation. In this chapter we consider the regulation problem for an intersection with two signal groups: signal group 1 and signal group 2. We assume without loss of generality that the signals in signal group 1 are numbered $1, 2, ..., N_1$ and the signals in signal group 2 are numbered $N_1+1, N_1+2, ..., N$. We use $\mathcal{G}_1 = \{1, 2, ..., N_1\}$ and $\mathcal{G}_2 = \{N_1+1, ..., N\}$. The problem description of the regulation problem is given in Section 5.1. In this chapter we use 'desired trajectory' to refer to the trajectory obtained via trajectory optimization.

8.1 Assumptions

We assume that the arrival rate and the maximum departure rate of a signal is positive:

$$\lambda_i, \mu_i > 0, \quad i \in \mathcal{N}. \tag{8.1a}$$

We assume that all setup times are non-negative and that the setup $\sigma_{1,N,1}$ is strictly positive:

$$\sigma_{i,j} \geq 0, \quad i, j \in \mathcal{N}, \tag{8.1b}$$

$$\sigma_{1,N,1} > 0. \tag{8.1c}$$

Further, we assume that the setup times are related according to:

$$\sigma_{i_1,i_2} - \sigma_{i_1,l_2} = \sigma_{l_1,i_2} - \sigma_{l_1,l_2}, \qquad \forall i_1, l_1 \in \mathcal{G}_1 \qquad i_2, l_2 \in \mathcal{G}_2 \tag{8.1d}$$

$$\sigma_{i_2,i_1} - \sigma_{i_2,l_1} = \sigma_{l_2,i_1} - \sigma_{l_1,l_2}, \qquad \forall i_1, l_1 \in \mathcal{G}_1 \qquad i_2, l_2 \in \mathcal{G}_2 \tag{8.1e}$$

Furthermore, we assume that the maximum green times are related according to (8.1f) and (8.1g). See Section 7.2.3 for more information about this assumption.

$$g_{i_1}^{max} + \sigma_{i_1,i_2,i_1} = g_{j_1}^{max} + \sigma_{j_1,i_2,j_1}, \quad \forall i_1, j_1 \in \mathcal{G}_1, \quad \forall i_2 \in \mathcal{G}_2, \tag{8.1f}$$

$$g_{i_2}^{max} + \sigma_{i_1,i_2,i_1} = g_{j_2}^{max} + \sigma_{i_1,j_2,i_1}, \quad \forall i_1 \in \mathcal{G}_1, \qquad \forall i_2, j_2 \in \mathcal{G}_2. \tag{8.1g}$$

Furthermore, we assume that (8.1h)–(8.1s) are satisfied. The inequalities (8.1h)–(8.1s) are the strict form of (7.16). For more information about these assumptions see Section 7.5.2.

$$g_{i_1}^{max} > \frac{\sigma_{i_1,i_2,i_1}\rho_{i_1}}{1-\rho_{i_1}-\rho_{i_2}}, \tag{8.1h}$$
$$g_{i_1}^{max} > g_{i_1}^{min}, \tag{8.1i}$$
$$g_{i_1}^{max} > (g_{i_2}^{min}+\sigma_{i_1,i_2,i_1})\frac{\rho_{i_1}}{1-\rho_{i_1}}, \tag{8.1j}$$
$$g_{i_2}^{max} > \frac{\sigma_{i_1,i_2,i_1}\rho_{i_2}}{1-\rho_{i_1}-\rho_{i_2}}, \tag{8.1k}$$
$$g_{i_2}^{max} > g_{i_2}^{min}, \tag{8.1l}$$
$$g_{i_2}^{max} > (g_{i_1}^{min}+\sigma_{i_1,i_2,i_1})\frac{\rho_{i_2}}{1-\rho_{i_2}}, \tag{8.1m}$$
$$x_{i_1}^{max} > \lambda_{i_1}(\frac{\sigma_{i_1,i_2,i_1}\rho_{i_2}}{1-\rho_{i_1}-\rho_{i_2}}+\sigma_{i_1,i_2,i_1}), \tag{8.1n}$$
$$x_{i_1}^{max} > \lambda_{i_1}\frac{g_{i_1}^{min}\rho_{i_2}+\sigma_{i_1,i_2,i_1}}{1-\rho_{i_2}}, \tag{8.1o}$$
$$x_{i_1}^{max} > \lambda_{i_1}(g_{i_2}^{min}+\sigma_{i_1,i_2,i_1}), \tag{8.1p}$$
$$x_{i_2}^{max} > \lambda_{i_2}(\frac{\sigma_{i_1,i_2,i_1}\rho_{i_1}}{1-\rho_{i_1}-\rho_{i_2}}+\sigma_{i_1,i_2,i_1}), \tag{8.1q}$$
$$x_{i_2}^{max} > \lambda_{i_2}\frac{g_{i_2}^{min}\rho_{i_1}+\sigma_{i_1,i_2,i_1}}{1-\rho_{i_1}}, \tag{8.1r}$$
$$x_{i_2}^{max} > \lambda_{i_2}(g_{i_1}^{min}+\sigma_{i_1,i_2,i_1}). \tag{8.1s}$$

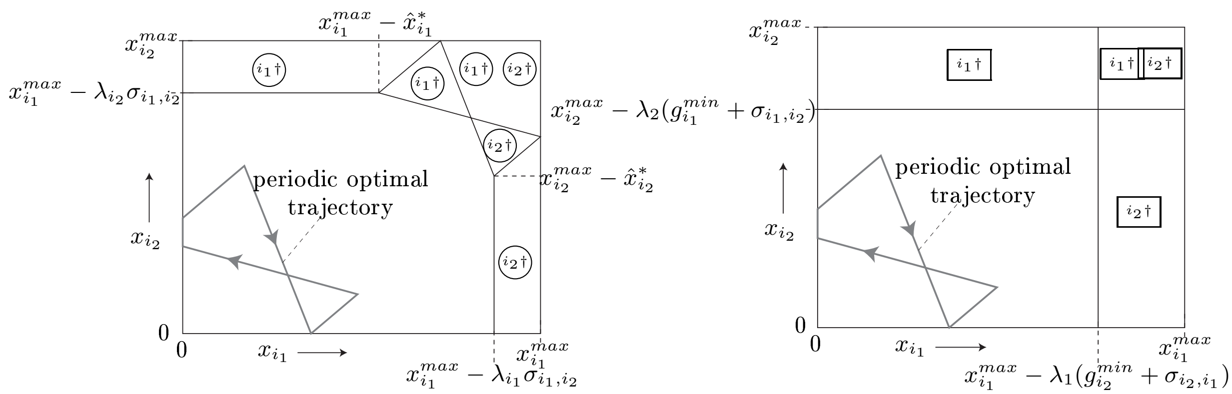
Further we assume that a desired trajectory (the trajectory that we want to converge to), satisfies the periodic shape from Corollary 7.4.

8.2 Convergence

Before proposing the policy we show that convergence to the desired trajectory is not always possible.

The (x_{i_1}, x_{i_2})-plane, $i_1 \in \mathcal{G}_1$, $i_2 \in \mathcal{G}_2$ can be divided into regions from which it is impossible to converge to the periodic optimal trajectory when in a certain mode. When entering the area annotated with $i_1\dagger$ (see Figure 8.1a) while serving signal i_1, eventually one of the constraints is violated. When performing a setup to signal 2, a maximum queue length is exceeded. Moreover, if we do not perform this setup, a maximum queue length is exceeded as well. Similarly, whenever entering the area annotated with $i_2\dagger$ while serving signal i_2, eventually one of the constraints is violated. If the trajectory is on the pure bow tie curve in the upper right corner the trajectory stays here (if the minimum green times allow so).

Further, because of restrictions on the minimum green period duration, we may not start serving signal i_1 respectively signal i_2 in the areas annotated with $\boxed{i_1\dagger}$ respectively $\boxed{i_2\dagger}$ (see Figure 8.1b). Hence, when the initial queue lengths are in the area annotated with $\boxed{i_1\dagger}$ we have to start serving signal group 2 and when the initial queue lengths are in the area annotated with $\boxed{i_2\dagger}$ we have to with serving signal group 1. When the initial queue lengths are in the area with both $\boxed{i_1\dagger}$ and $\boxed{i_2\dagger}$, eventually a constraint is violated.



(a) When entering the area annotated with an encircled $i_1\dagger$ or an encircled $i_2\dagger$ while serving signal i_1 respectively signal i_2 no convergence to the periodic optimal trajectory is possible anymore.

(b) When starting to serve signal i_1 or signal i_2 in the areas annotated with a boxed $i_1\dagger$ respectively a boxed $i_2\dagger$ a constraint will be violated.

Figure 8.1: Convergence to the periodic optimal process is not always possible.

8.3 Proposing a Policy

In this section we propose a policy that makes sure that a trajectory converges to the desired periodic trajectory. First we introduce some notation.

Just like the desired trajectory, the policy switches signal $i_2 \in \mathcal{G}_2$ to green σ_{i_1,i_2} seconds after signal $i_1 \in \mathcal{G}_1$ switched to red and the policy switches signal $i_2 \in \mathcal{G}_1$ to green σ_{i_2,i_1} seconds after signal $i_2 \in \mathcal{G}_2$ switched to red.

Without loss of generality we assume that the signals in signal group 1 are numbered such that:

$$\sigma_{1,N} \geq \sigma_{2,N} \geq \cdots \geq \sigma_{N_1,N}$$

and that the signals in signal group 2 are numbered such that:

$$\sigma_{N_1+1,1} \geq \sigma_{N_1+2,1} \geq \cdots \geq \sigma_{N,1}.$$

For this order, signal 1 is the first signal in $\mathcal{G}_1$ to switch to red (see Figure 8.2). Further, signal $i_1 \in \mathcal{G}_1$ switches to red $\sigma_{1,N} - \sigma_{i_1,N}$ seconds after signal 1 has switched to red and signal $i_2 \in \mathcal{G}_2$ switches to green σ_{1,i_2} seconds after signal 1 switched to red. In the same way, signal $N_1 + 1$ is the first signal in $\mathcal{G}_2$ to switch to red. Signal $i_2 \in \mathcal{G}_2$ switches to red $\sigma_{N_1+1,1} - \sigma_{i_2,1}$ seconds after signal $N_1 + 1$ has switched to red and signal $i_1 \in \mathcal{G}_1$ switches to green σ_{N_1+1,i_1} seconds after signal $N_1 + 1$ switched to red.

We want to derive a rule that defines when to switch signal 1 to red and when to switch signal $N_1 + 1$ to red. From these two switch actions, we can derive when to switch each of the signals to green and red.

For the policy that we propose we use $x_{i_1}^{\sharp}$ for the queue length of queue $i_1 \in \mathcal{G}_1$ at the moment that signal 1 switches to red for the desired trajectory. We use $x_{i_2}^{\sharp}$ for the queue length of queue $i_2 \in \mathcal{G}_2$ at the moment that signal $N_1 + 1$ switches to red for the desired trajectory. We can obtain the following expressions for $x_{i_1}^{\sharp}$ and $x_{i_2}^{\sharp}$:

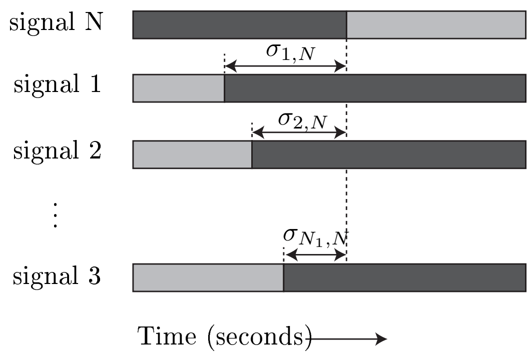


Figure 8.2: Sequence in which signals in $\mathcal{G}_1$ switch to red.

$$x^\sharp_{i_1} = (r_{i_1} - \sigma_{i_1,N})\lambda_{i_1}, \tag{8.2a}$$
$$x^\sharp_{i_1} = (r_{i_2} - \sigma_{i_2,1})\lambda_{i_2}. \tag{8.2b}$$

where r_i, $i \in \mathcal{N}$ is the red time of signal i for the desired trajectory. Furthermore, we use:

$$i_1^{r,f} = 1,$$
$$i_2^{r,f} = N_1 + 1.$$

Thus $i_c^{r,f}$, $c = 1, 2$ refers to the signal in the set $\mathcal{G}_c$ that is switched to red first. We use $i_1^{g,f}$ to refer to the signal in the set $\mathcal{G}_1$ that switches to green first:

$$i_1^{g,f} = \arg\min_{i_1 \in \mathcal{G}_1} \sigma_{N,i_1}. \tag{8.3}$$

We use $i_2^{g,f}$ to refer to the signal in the set $\mathcal{G}_2$ that is switches to green first:

$$i_2^{g,f} = \arg\min_{i_2 \in \mathcal{G}_2} \sigma_{1,i_2}. \tag{8.4}$$

Further we use $\sigma^{res}_{i_1}$ for the residual time that signal $i_1 \in \mathcal{G}_1$ has to be red for at the moment that signal $i_1^{g,f}$ switches to green:

$$\sigma^{res}_{i_1} = \sigma_{i_2,i_1} - \sigma_{i_2,i_1^{g,f}}, \quad i_2 \in \mathcal{G}_2. \tag{8.5}$$

We use $\sigma^{res}_{i_2}$ for the residual time that signal $i_2 \in \mathcal{G}_2$ has to be red for at the moment that signal $i_2^{g,f}$ switches to green:

$$\sigma^{res}_{i_2} = \sigma_{i_1,i_2} - \sigma_{i_1,i_2^{g,f}}, \quad i_1 \in \mathcal{G}_1. \tag{8.6}$$

We use τ_i for the time that has elapsed since the last mode change at signal $i \in \mathcal{N}$.

8.3.1 Overview of the Policy

First we give a short overview of the policy. In Section 8.3.2 we give a formal expression to determine when to switch signal 1 to red and when to switch signal $N_1 + 1$ to red.

We want to serve the signals in the set $\mathcal{G}_c$, $c = 1, 2$ long enough to satisfy the following 3 conditions:

1.1 all queues i_c, $i_c \in \mathcal{G}_c$ are (expected to be) emptied during their green period (assuming a hybrid fluid model).

1.2 all signals are served for at least the minimum green time.

1.3 the queue length of at least one of the queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ satisfies $x_{i_{\overline{c}}} \geq x^{\sharp}_{i_{\overline{c}}}$, whre $\overline{c} = \begin{cases} 1 & \text{if c=2,} \\ 2 & \text{if c=1.} \end{cases}$.

Whenever conditions 1.1–1.3 are satisfied we switch signal 1 (if $c = 1$) or signal $N_1 + 1$ (if $c = 2$) to red.

It might not be possible to serve the signals in the set $\mathcal{G}_c$ long enough to satisfy conditions 1.1–1.3. We might have to stop earlier because of condition 2 or condition 3:

2 The maximum green time of signal 1 (if $c = 1$) or the maximum green time of signal N_1+1 (if $c = 2$) is reached. From (8.1f) we know that when signal 1 is served for the maximum green time then all signals in $\mathcal{G}_1$ are served for the maximum green time and from (8.1g) we know that when signal $N_1 + 1$ is served for the maximum green time then all signals in $\mathcal{G}_2$ are served for the maximum green time.

3 queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ has reached a queue length of $x^{max}_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}\sigma_{1,i_{\overline{c}}}$ (if $c = 1$) or a queue length of $x^{max}_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}\sigma_{N_1+1,i_{\overline{c}}}$ (if $c = 2$). In this case queue $i_{\overline{c}}$ is switched to green when its queue length reaches $x^{max}_{i_{\overline{c}}}$ (assuming a hybrid fluid model).

8.3.2 Switching the Signals 1 and $N_1 + 1$ to Red

In this sections we give formal expressions for when the conditions, introduced in the previous section, are satisfied.

Formal expression for condition 1.1 Condition 1.1 is satisfied when all queues i_c, $i_c \in \mathcal{G}_c$ are (expected to be) emptied during their green period (assuming a hybrid fluid model).

When signal $i^{g,f}_{\overline{c}} \in \mathcal{G}_c$, $c = 1, 2$ switches to red at time t, signal $i_c \in \mathcal{G}_c$ is still red for $\max\{\sigma^{res}_{i_c} - \tau_{i^{g,f}_{\overline{c}}}(t), 0\} = (\sigma^{res}_{i_c} - \tau_{i^{g,f}_{\overline{c}}}(t))^+$ seconds (see Figure 8.3). When $(\sigma^{res}_{i_c} - \tau(t))^+$ is positive, this means that the setup towards signal i_c is not finished yet. During this residual part of the setup, traffic arrives at signal i_c at arrival rate λ_{i_c}.

Since signal i_c switches to red $\sigma_{i^{r,f}_c, i^{g,f}_{\overline{c}}} - \sigma_{i_c, i^{g,f}_{\overline{c}}}$ seconds after signal $i^{r,f}_c$ switches to red, signal $i_c \in \mathcal{G}_c$ is still green for a duration of $\sigma_{i^{r,f}_c, i^{g,f}_{\overline{c}}} - \sigma_{i_c, i^{g,f}_{\overline{c}}} - (\sigma^{res}_{i_c} - \tau_{i^{g,f}_{\overline{c}}}(t))^+$ if signal $i^{r,f}_c$ switches to red at time t.

Hence, condition 1.1 is satisfied when:

$$\forall i_2 \in \mathcal{G}_2 \left(x_{i_2}(t) \leq (\sigma_{i^{r,f}_c, i^{g,f}_{\overline{c}}} - \sigma_{i_c, i^{g,f}_{\overline{c}}})(\mu_{i_c} - \lambda_{i_c}) - (\sigma^{res}_{i_c} - \tau_{i^{g,f}_{\overline{c}}}(t))^+ \mu_{i_c} \right). \tag{8.7}$$

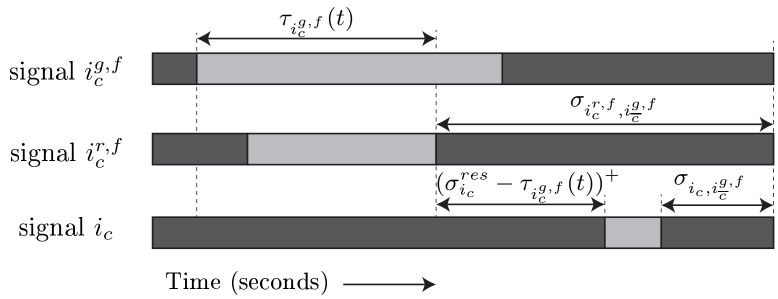


Figure 8.3: Visualization of the formal expression for condition 1.1. The dark gray rectangles visualize when a signal is red and the light gray rectangles visualize when a signal is green.

Formal expression for condition 1.2 Condition 1.2 is satisfied when all signals are served for at least the minimum green time.

Using the fact that signal i_c switches to green $\sigma_{i_c}^{res}$ seconds after signal $i_c^{g,f}$ switched to green (see Figure 8.4) and using the fact that signal i_c switches to red $\sigma_{i_c^{r,f},i_{\overline{c}}^{g,f}} - \sigma_{i_c,i_{\overline{c}}^{f}}$ seconds after signal $i_c^{r,f}$ switches to red we can find that condition 1.2 is satisfied when $\tau_{i_c^{g,f}}(t) \geq \max_{i_c \in \mathcal{G}_c}(g_{i_c}^{min} + \sigma_{i_c}^{res} + \sigma_{i_c,i_{\overline{c}}^{g,f}} - \sigma_{i_c^{r,f},i_{\overline{c}}^{g,f}})$.

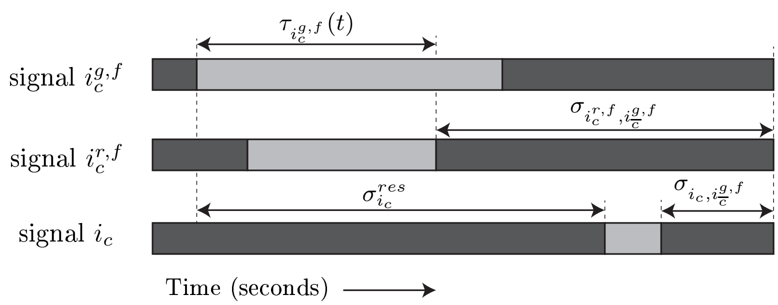


Figure 8.4: Visualization of the formal expression for condition 1.2. The dark gray rectangles visualize when a signal is red and the light gray rectangles visualize when a signal is green.

Formal expression for condition 1.3 Condition 1.3 is satisfied when $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} \left(x_{i_{\overline{c}}} \geq x_{i_{\overline{c}}}^{\sharp} \right)$.

Formal expression for condition 2 Condition 2 is satisfied whenever the maximum green time of signal 1 (if $c = 1$) or the maximum green time of signal $N_1 + 1$ (if $c = 2$) is reached.

The green period of signal $i_c^{r,f}$ starts $\sigma_{i_c^{r,f}}^{res}$ seconds after signal $i_c^{g,f}$ is switched to green. Using (8.1f) and (8.1g) we can see that condition 2 is satisfied whenever $\tau_{i_c^{g,f}}(t) \geq g_{i_c^{r,f}}^{max} + \sigma_{i_c^{r,f}}^{res}$.

Formal expression for condition 3 Condition 3 is satisfied when $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} \left(x_{i_{\overline{c}}}(t) \geq x_{i_{\overline{c}}}^{max} - \sigma_{i_c^{r,f},i_{\overline{c}}} \lambda_{i_{\overline{c}}} \right)$

Thus, we switch the signal $i_c^{r,f}$ to red when serving signal group $c = 1, 2$ if the following expression is true:

$$(t_{1.1} \wedge t_{1.2} \wedge t_{1.3}) \vee t_2 \vee t_3,$$

where

$$t_{1.1} = \forall i_2 \in \mathcal{G}_2 \left(x_{i_2}(t) \leq (\sigma_{i_c^{r,f}, i_{\overline{c}}^{g,f}} - \sigma_{i_c, i_{\overline{c}}^{g,f}})(\mu_{i_c} - \lambda_{i_c}) - (\sigma_{i_c}^{res} - \tau_{i_c^{g,f}}(t))^+ \mu_{i_c} \right),$$
$$t_{1.2} = \tau_{i_c^{g,f}}(t) \geq \max_{i_c \in \mathcal{G}_c} (g_{i_c}^{min} + \sigma_{i_c}^{res} + \sigma_{i_c, i_{\overline{c}}^{g,f}} - \sigma_{i_c^{r,f}, i_{\overline{c}}^{g,f}}),$$
$$t_{1.3} = \exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} \left(x_{i_{\overline{c}}} \geq x_{i_{\overline{c}}}^{\sharp} \right),$$
$$t_2 = \tau_{i_c^{g,f}}(t) \geq g_{i_c^{r,f}}^{max} + \sigma_{i_c^{r,f}}^{res},$$
$$t_3 = \exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} \left(x_{i_{\overline{c}}}(t) \geq x_{i_{\overline{c}}}^{max} - \sigma_{i_c^{r,f}, i_{\overline{c}}} \lambda_{i_{\overline{c}}} \right).$$

Proposition 8.1 *A feedback policy which stabilizes an intersection with two signal groups to the desired periodic trajectory if started from a feasible starting point (for which we avoid the areas in Figure 8.1 from which no convergence is possible) is given by:*

- *Mode 1: Serve signal $i_1 \in \mathcal{G}_1$ at the highest possible rate if signal i_1 is green. Switch signal $i_1 \in \mathcal{G}_1$ to green if $\tau_{i_1^{g,f}} \geq \sigma_{i_1}^{res}$. If (8.8) results in the boolean 'true' for $c = 1$ then switch signal 1 to red and go to Mode 2.*
- *Mode 2: Switch signal i_1 to red if $\tau_1 \geq \sigma_{1,N} - \sigma_{i_1,N}$. If $\tau_1 \geq \sigma_{1,i_2^{g,f}}$ then switch signal $i_2^{g,f}$ to green and go to Mode 3.*
- *Mode 3: Serve signal $i_2 \in \mathcal{G}_2$ at the highest possible rate if signal i_2 is green. Switch signal $i_2 \in \mathcal{G}_2$ to green if $\tau_{i_2^{g,f}} \geq \sigma_{i_2}^{res}$. If (8.8) results in the boolean 'true' for $c = 2$ then switch signal $N_1 + 1$ to red and go to Mode 4.*
- *Mode 2: Switch signal i_2 to red if $\tau_{N_1+1} \geq \sigma_{N_1+1,1} - \sigma_{i_2,1}$. If $\tau_{N_1+1} \geq \sigma_{N_1+1,i_1^{g,f}}$ then switch signal $i_1^{g,f}$ to green and go to Mode 1.*

Proof. Below we give a sketch of the proof. See Appendix D for the entire proof of this proposition.

We distinguish 5 different reasons why we switch signal $i_c^{r,f}$, $c = 1, 2$ to red: switch.1a, switch.1b, switch.2, switch.3a and switch.3b (see Appendix D).

We consider an infinite sequence of reasons why we switch the signals in the set $\mathcal{G}_1$ from green to red and why we switch the signals in the set $\mathcal{G}_2$ from green to red. Below we can see an example of such an infinite sequence. We use s^l for the lth switch reason. When we start serving the signals in the set $\mathcal{G}_1$ then the switch reasons s^{2l+1}, $l = 0, 1, 2, \ldots, \infty$ refer to why we switch signal 1 from green to red and all switch reasons s^{2l}, $l = 1, 2, 3, \ldots, \infty$ refer to why we switched the signal $N_1 + 1$ from green to red.

$$s^1 \to s^2 \to s^3 \to \cdots = switch.3a \to switch.3a \to switch.3a \to switch.3a \to switch.3a \to \cdots \to$$
$$switch.2 \to switch.3a \to switch.2 \to switch.3a \to \cdots \to switch.1a \to switch.1a \to switch.1a \to \ldots$$

We can prove that after some (finite) time only *switch.1a* occurs or only *switch.1b* occurs. Which of these two depends on the characteristics of the desired trajectory. We prove that whenever one of these two switch reasons (either $switch.1a$ or $switch.1b$) occurs until infinity we converge to the desired trajectory.

Below we show an overview of how to prove that eventually (in finite time) only $switch.1a$ occurs or only $switch.1b$ occurs. We consider combinations of 2 subsequent switch reasons $(s^l \rightarrow s^{l+1})$, $l > 1$ (for example $(s^l \rightarrow s^{l+1}) = (switch.3a \rightarrow switch.3a)$, $(s^l \rightarrow s^{l+1}) = (switch.2 \rightarrow switch.3a)$ etc.). We use $C_i = 1, \ldots, n_c$ to refer to a certain combination of switch reasons. We use $\mathcal{C}_i$, $i = 1, \ldots, n_s$ to refer to a set of combinations of switch reasons. These sets satisfy:

$$
\begin{aligned}
&\mathcal{C}_i \neq \emptyset, \\
&\bigcup_{i=1}^{i=n_c} \mathcal{C}_i = \mathcal{C}, \\
&\mathcal{C}_i \cap \mathcal{C}_j = \emptyset, \qquad && i = 1, \ldots, n_s, \quad j = 1, \ldots, n_s, i \neq j,
\end{aligned}
$$

where

$$
\begin{aligned}
&\mathcal{C}_{n_s-1} = \{(switch.1a \rightarrow switch1a)\}, \\
&\mathcal{C}_{n_s} = \{(switch.1b \rightarrow switch1b)\}, \\
&\mathcal{C} = \{C_i : i = 1, \ldots, n_c\}.
\end{aligned}
$$

Thus, none of the sets $\mathcal{C}_i$, $i = 1, \ldots, n_s$ is empty and each combination $C_i = 1, \ldots, n_c$ is partitioned in exactly one of the set $\mathcal{C}_i$, $i = 1, \ldots, n_s$. Note, that in total there are $5 \times 5 = 25$ combinations of 2 subsequent switch reasons possible because there are 5 different switch reasons ($switch.1a$, $switch.1b$, $switch.2$, $switch.3a$ and $switch.3b$). However, we do not use all combinations, i.e. $n_c < 25$. We have chosen the n_c combinations such that for every (feasible) infinite sequence of switch reasons, each of those switch reasons s^l, $l > 1$ makes a combination C_i, $i = 1, \ldots, n_c$ with either the previous switch reason, the next switch reason or both, i.e. $\forall l > 1 : (s^{l-1} \rightarrow s^l) \in \mathcal{C} \vee (s^l \rightarrow s^{l+1}) \in \mathcal{C}$. Thus, every switch reason is part of a combination C_i, $i = 1, \ldots, n_c$.

First of all, we can prove that whenever s^l, $l \geq l_{start}$ (where l_{start} is a finite integer) is part of a combination that is in $\mathcal{C}_i$, $1 < i \leq n_s$ (with either s^{l-1} or s^{l+1}) then s^{l+1} cannot be part of a combination that is in the set $\mathcal{C}_j$, $1 \leq j < i$ (with either s^l or s^{l+2}). Note that this means that whenever a combination in the set $(s_{l-1}, s_l) = \mathcal{C}_i$, $2 < i \leq n$, $l \geq l_{start}$ has occurred then a combination in the set $\mathcal{C}_j$, $1 \leq j < i$ can never occur again. Furthermore, we can prove that only a finite number of subsequent switch reasons s^l can be part of a combination in the set $\mathcal{C}_i$, $1 \leq i \leq n_s - 2$ (with either s^{l-1} or s^{l+1}). Hence, eventually only combinations in the sets $\mathcal{C}_{n_s-1}$ or $\mathcal{C}_{n_s}$ can occur. As previously mentioned for an infinite sequence of $switch.1a$ switch reasons or an infinite sequence of $switch.1b$ switch reasons we can show convergence to the desired signal.

■

Chapter 9

Quality of the Policy in a Stochastic Setting: An Intersection With Two Signal Groups

In Chapter 7 we derived periodic optimal trajectories for an intersection with two signal groups by modeling the intersection with a hybrid fluid model. In Chapter 8 we proposed a (feed-back) policy. In this chapter we consider the third problem discussed in this thesis; we address the quality of the proposed policy for an intersection with two signal groups in a stochastic setting. To this end, we model the intersection with the stochastic model described in Section 3.1. Recall that this stochastic model assumes Poisson arrivals and deterministic departures. To obtain results for the policy in a stochastic setting a simulation program is made in the programming language χ3.0. The code of this simulation program is given in Appendix B. For each test case we obtain the average delay δ (in seconds) of a road user at the intersection and we obtain the fraction of the time that the maximum queue length is exceeded at each of the queues. A road user could either be a vehicle, a cyclist or a pedestrian.

Before simulating a test case we calculate the following information about the periodic optimal trajectory (see Chapter 4).

- The coordinates $x^\sharp_{i_1}$, $i_1 \in \mathcal{G}_1$ $x^\sharp_{i_2}$, $i_2 \in \mathcal{G}_2$ calculated with (8.2).
- The cycle duration $c = g_1 + g_N + \sigma_{1,N,1}$.
- The queue lengths of the signals $i_1 \in \mathcal{G}_1$ at the beginning of a green period, which is equal to $\lambda_{i_1}(g_N + \sigma_{i_1,N,i_1})$, $i_1 \in \mathcal{G}_1$.
- The average delay of a road user. This average delay is obtained using (7.12), where $w = w_1 = \cdots = w_N = 1$ and (7.2).

For each test case we perform at least 100 runs. We perform enough runs such that the 95% confidence interval for the average delay of a road user is at most 1% of the average delay of a road user. For each run we start with the situation where all signals in signal group 1 are green. At the start of a run the queue length of queue $i_1 \in \mathcal{G}_1$ is taken equal to $\lceil \lambda_{i_1}(g_N + \sigma_{i_1,N,i_1}) \rceil$ (obtained from the periodic optimal trajectory). At the start of a run the queue length of queue $i_2 \in \mathcal{G}_2$ is zero. Each run simulates $100c$ seconds, were c is the cycle duration of the periodic optimal trajectory (see Section 7.4). We consider the following test cases.

test case 1a	test case 1b	test case 1c
$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0.0125, 0.0250, \ldots, 0.2375$	$\lambda_1 = \lambda_2 = \frac{0.4}{1+f}$	$\lambda_1 = \lambda_3 = 0.2$
	$\lambda_3 = \lambda_4 = \frac{0.4f}{1+f}$	$\lambda_2 = \lambda_4 = \frac{0.2}{f}$
	$f = 1, 1.2, ..., 4$	$f = 1, 1.2, ..., 4$
$g_1^{max} = g_2^{max} = \infty$	$g_1^{max} = g_2^{max} = \infty$	$g_1^{max} = g_2^{max} = \infty$
$g_3^{max} = g_4^{max} = \infty$	$g_3^{max} = g_4^{max} = \infty$	$g_3^{max} = g_4^{max} = \infty$
$x_1^{max} = x_2^{max} = \infty$	$x_1^{max} = x_2^{max} = \infty$	$x_1^{max} = x_2^{max} = \infty$
$x_3^{max} = x_4^{max} = \infty$	$x_3^{max} = x_4^{max} = \infty$	$x_3^{max} = x_4^{max} = \infty$

test case 2a	test case 2b
$\lambda_1 = \lambda_3 = \frac{1}{15}$	$\lambda_1 = \lambda_3 = \frac{1}{15}$
$\lambda_2 = \lambda_4 = \frac{5}{15}$	$\lambda_2 = \lambda_4 = \frac{5}{15}$
$g_1^{max} = g_3^{max} = \infty$	$g_1^{max} = g_3^{max} = 4, 4.5, \ldots, 9$
$g_2^{max} = g_3^{max} = 16, 16.5, \ldots, 26$	$g_2^{max} = g_4^{max} = \infty$
$x_1^{max} = x_3^{max} = \infty$	$x_1^{max} = x_3^{max} = \infty$
$x_2^{max} = x_4^{max} = \infty$	$x_2^{max} = x_4^{max} = \infty$

test case 3a	test case 3b
$\lambda_1 = \lambda_3 = \frac{1}{15}$	$\lambda_1 = \lambda_3 = \frac{1}{15}$
$\lambda_2 = \lambda_4 = \frac{5}{15}$	$\lambda_2 = \lambda_4 = \frac{5}{15}$
$g_1^{max} = g_3^{max} = \infty$	$g_1^{max} = g_3^{max} = \infty$
$g_2^{max} = g_4^{max} = \infty$	$g_2^{max} = g_4^{max} = \infty$
$x_1^{max} = x_3^{max} = \infty$	$x_1^{max} = x_3^{max} = 2, 3, \ldots, 7$
$x_2^{max} = x_4^{max} = 3, 4, \ldots, 30$	$x_2^{max} = x_4^{max} = \infty$

For all these test cases it holds that:

$$
\begin{aligned}
\mathcal{G}_1 &= \{1, 2\}, \\
\mathcal{G}_2 &= \{3, 4\}, \\
\mu_1 &= \mu_2 = \mu_3 = \mu_4 = 0.5, \\
\sigma_{1,3} &= \sigma_{1,4} = \sigma_{2,3} = \sigma_{2,4} = \sigma_{3,1} = \sigma_{3,2} = \sigma_{4,1} = \sigma_{4,2} = 2 \\
g_1^{min} &= g_2^{min} = g_3^{min} = g_4^{min}
\end{aligned}
$$

For test cases 1b, 1c, 2a, 2b, 3a and 3b it holds that $\max\{\rho_1, \rho_2\} + \max\{\rho_3, \rho_4\} = 0.8$. In this chapter we use $\mu = \mu_1 = \mu_2$. In sections 9.1–9.3 we show the results for these test cases. In Section 6 we compared our proposed policy to an exhaustive policy. We do not compare our proposed policy to an exhaustive policy in this section because (6.1) considers intersections where 1 signal is green at a time.

9.1 Test Case 1: Effect of the Arrival Rates

In this section we address the effect of the arrival rate on the average delay of a road user. The results for the different test cases are shown in sections 9.1.1–9.1.3.

9.1.1 Test Case 1a: Effect of Increasing Arrival Rates

In this test case we address the effect of increasing the arrival rates on the delay; we want to determine $\delta(\lambda)$ for the proposed policy, where the arrival rates are varied as follows:

$$\lambda = \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0.0125, 0.025, ..., 0.2375.$$

As a result $\rho = \max\{\rho_1, \rho_2\} + \max\{\rho_3, \rho_4\}$ varies as follows:

$$\rho = 0.05, 0.1, ..., 0.95.$$

In Figure 9.1 the results are shown.

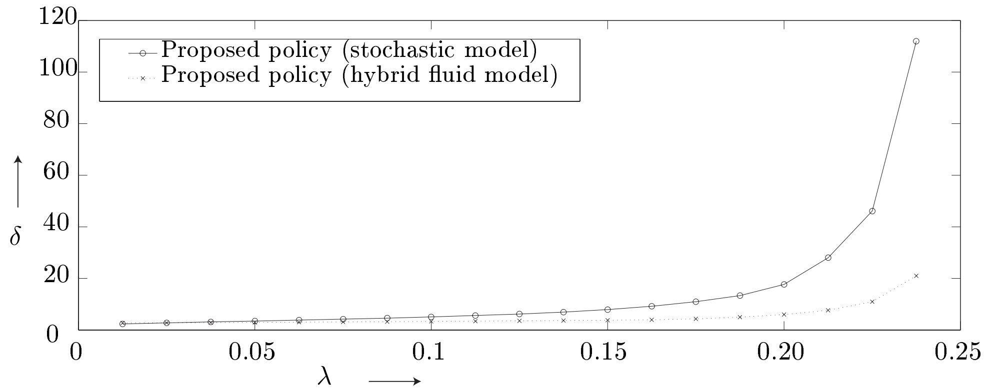


Figure 9.1: The average delay of a road user δ versus λ for test case 1a.

The results shown in this figure are very similar to the results shown in Figure 6.1. Especially for $\lambda \leq 0.15$ the delays obtained for an intersection of 4 signals are close to the delays obtained for an intersection of 2 signals. As seen in Section 6.2 we can again observe that $\delta(\lambda)$ goes to 2 for the proposed policy when $\lambda \to 0$ (assuming the stochastic model). However, for larger arrival rates we can see that the difference in delays obtained for an intersection with 4 signals and obtained for an intersection with 2 signals gets larger. The delays are larger for an intersection with 4 signals because for this intersection it takes longer to satisfy condition 1.2 given in Section 8.3.1, i.e. it takes longer before all queues in the same signal group are emptied.

9.1.2 Test Case 1b: Effect of Asymmetrical Arrival Rates Between Signal Group

For this test case the arrival rates are varied as follows:

$$\lambda_1 = \lambda_2 = \frac{0.4}{1+f},$$
$$\lambda_3 = \lambda_4 = \frac{0.4f}{1+f},$$
$$f = 1, 1.2, ..., 4.$$

The results are shown in Figure 9.2. In this figure we can see that larger differences in arrival rates (and the same value for ρ) result in smaller delays. When the differences in arrival rates between signal groups is larger, a larger proportion of the road users arrives at signals from signal group 2. As a result, more road users can benefit from a slow mode at a signal in signal group 2. We can see that the average delay even goes to zero for $\frac{\max\{\lambda_1,\lambda_2\}}{\max\{\lambda_3,\lambda_4\}} \to \infty$ when assuming infinite maximum green times and infinite

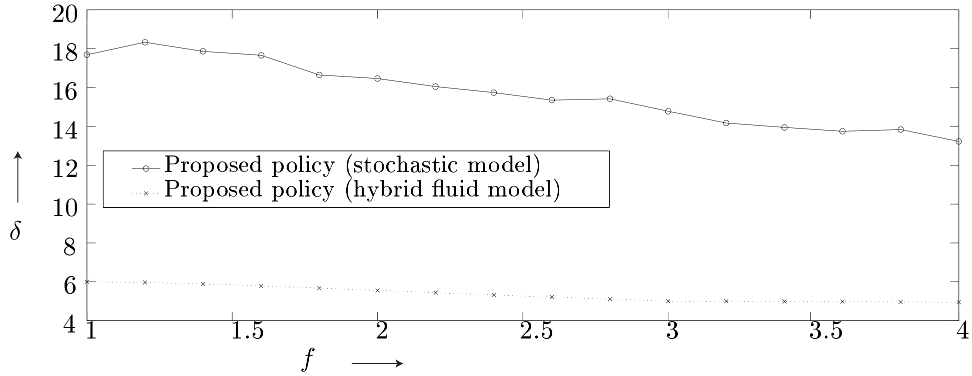


Figure 9.2: The average delay of a road user δ when the difference in arrival rates between the signals in different signal groups increases.

maximum queue lengths. The delay goes to zero for $\frac{\lambda_2}{\lambda_1} \to \infty$ because all of the road users arrive at signals in signal group 2. Hence, signals in signal group 2 can always be green. As a result, if queue $i_2 \in \mathcal{G}_2$ is emptied once, it always stays empty (slow mode) and each of the road users arriving during a slow mode experiences a delay of zero seconds.

9.1.3 Test Case 1c: Effect of Asymmetrical Arrival Rates in a Signal Group

For this test case the arrival rates are varied as follows:

$$\lambda_1 = \lambda_3 = 0.2$$
$$\lambda_2 = \lambda_4 = \frac{0.2}{f}$$
$$f = 1, 1.2, ..., 4$$

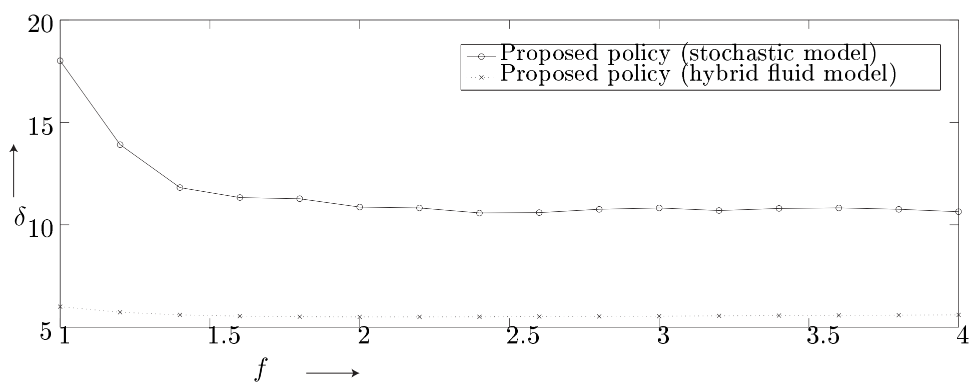


Figure 9.3: The average delay of a road user δ when the difference in arrival rates between signals in the same signal group increases.

The results are shown in Figure 9.3. When f increases the arrival rates at signal 2 and 4 decrease. Hence, for $f \to \infty$ all traffic arrives at signals 1 and 3 and the intersection with 4 signals is equivalent to an intersection with 2 signals. Thus, we can conclude that for an intersection with two signal groups the average delay of a road user increasing for an increasing number of signals in each of the signal group.

9.2 Test Case 2: Effect of The Maximum Green Time

In this section we address the effect of the maximum green times on the delay of a road user. For test case 2a and test case 2b the arrival rates at the signals in signal group 2 are 5 times as large as the arrival rates at the signals in signal group 1. We use 'low traffic signals' to refer to the signals in signal group 1 and we use 'high traffic signals' to refer to signals in signal group 2.

9.2.1 Test Case 2a: Effect of The Maximum Green Time of the High Traffic Signals

For this test case the maximum green time of the signals in signal group 2 are varied between 16 seconds and 26 seconds:

$$g_3^{max} = g_4^{max} = 16, 16.5, ..., 26. \tag{9.1}$$

A maximum green time of 16 seconds is the smallest maximum green time g_2^{max} satisfying (4.20) and thus the smallest maximum green time for which we can find an optimal trajectory. However, for the same reason as explained in Section 6.3 a maximum green time $g_3^{max} = g_4^{max} = 16$ seconds, does not result in stability; the queue lengths of queue 3 and queue 4 keep increasing.

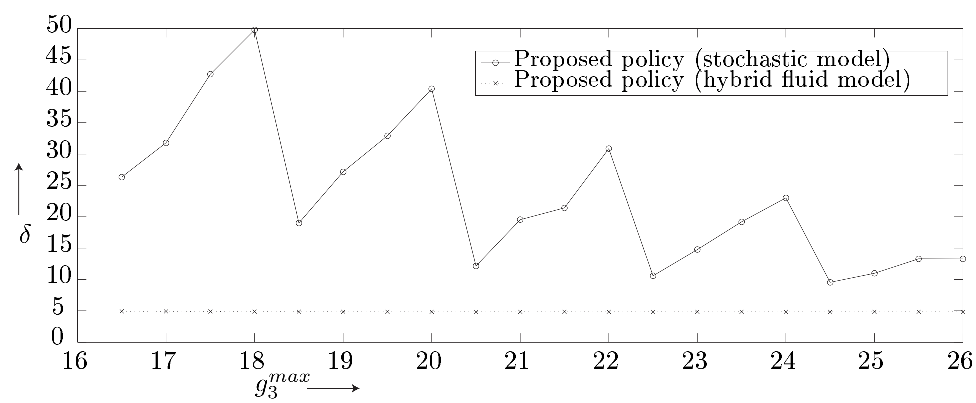


Figure 9.4: The average delay of a road user δ versus $g_3^{max} = g_4^{max}$ for test case 2a.

Just like in Figure 6.2 we can see the sawtooth shape function of the average delay of a road user as function of the maximum green time g_2^{max}. In Figure 9.4 we can see that the global trend (disregarding the sawtooth shape) is that smaller maximum green times result in larger delays. This global trend is more obvious for the intersection of 4 signals (Figure 9.4) than it was for the intersection of two signals (Figure 6.2).

9.2.2 Test Case 2b: Effect of the Maximum Green Time of the Low Traffic Signals

For test case 2b the maximum green time of the signals in signal group 1 are varied between 4 seconds and 9 seconds:

$$g_1^{max} = g_2^{max} = 4, 4.5, ..., 9.$$

A maximum green time of 4 seconds is the smallest maximum green time satisfying (4.20) and thus the smallest maximum green time for which we can find an optimal trajectory. For the optimal trajectory we serve the signals in signal group 1 for the minimum green time of 4 seconds (independent of g_1^{max} and g_2^{max}). In Figure 9.5 we can see the results.

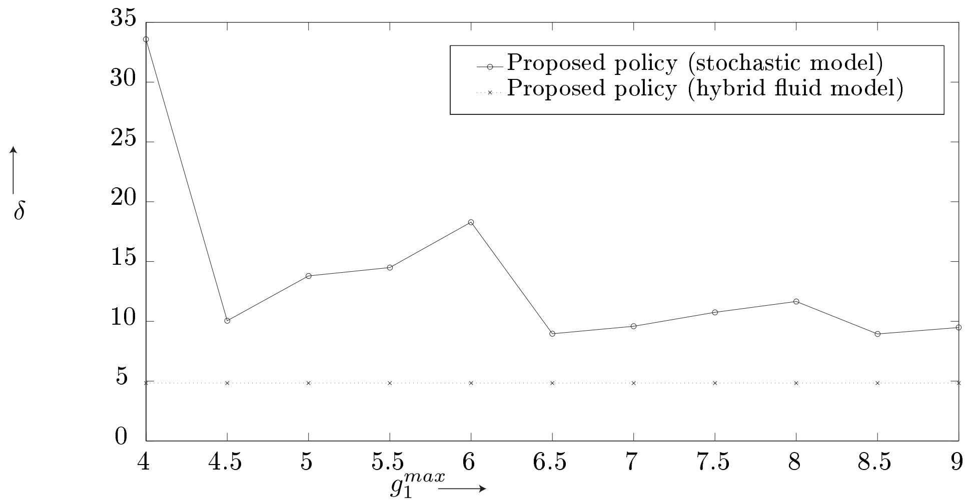


Figure 9.5: The average delay of a road user δ versus $g_1^{max} = g_2^{max}$ for test case 2b.

We again see the sawtooth relation between the maximum green time and the average delay of a road user. The global trend (disregarding the sawtooth shape) is that smaller maximum green times result in larger delays.

9.3 Test Case 3: Effect of Maximum Queue lengths

In this section we address the effect of the maximum queue lengths on the average delay of a road user. For test case 3a and test case 3b the arrival rates of the signals in signal group 2 are 5 times as large as the arrival rates of the signals in signal group 2. We use 'low traffic signals' to refer to the signals in signal group 1 and we use 'high traffic signals' to refer to the signals in signal group 2.

9.3.1 Test Case 3a: Maximum Queue Length of the High Traffic Signals

For test case 3a the maximum queue lengths of the signals in signal group 2 are varied between between 3 road users and 30 road users:

$$x_3^{max} = x_4^{max} = 3, 4, ..., 30.$$

A maximum queue length of $2\frac{2}{3}$ seconds is the smallest maximum green time satisfying (4.20) and thus the smallest maximum green time for which we can find an optimal trajectory. In Figure 9.6 we can see the average delay of a road user as function of the maximum queue lengths of signal 3 and signal 4.

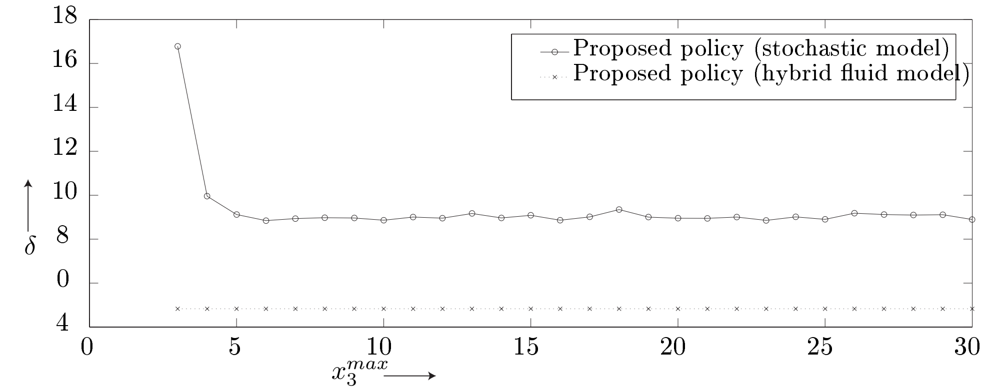


Figure 9.6: The average delay of a road user δ versus $x_3^{max} = x_4^{max}$ for test case 3a.

We can see that the average delay of a road user is about 9 seconds except when the maximum queue length is close to 3. In Figure 9.7 we can see that the fraction of the time that the maximum queue length (of signals 3 and 4) is exceeded, increases for decreasing maximum queue length. In this figure we can see that the variation in the results obtained for the fraction of overflow is quite large since the fraction of overflow should be the same for queue 3 and for queue 4 (because both signals have the same characteristics).

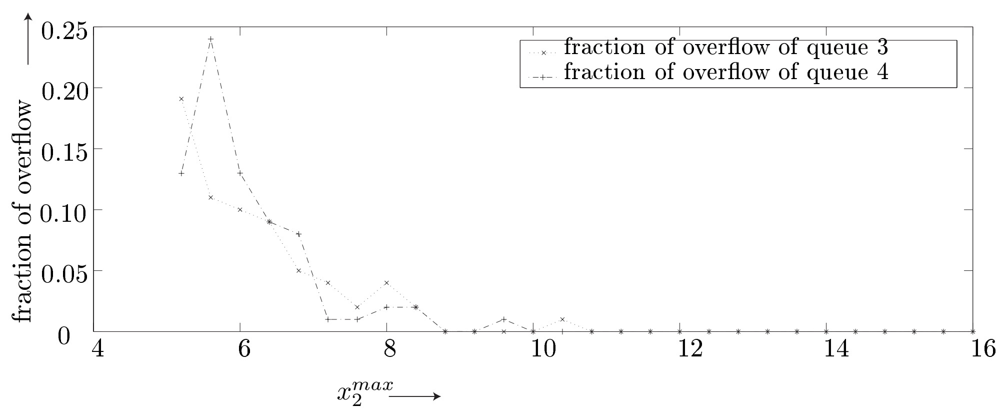


Figure 9.7: The fraction of the time that the maximum queue lengths of queues 1 and queue 2 are exceeded versus x_2^{max} for test case 3a.

9.3.2 Test Case 3b: Maximum Queue Length of the High Traffic Signals

For test case 3b the maximum queue lengths of the signals in signal group 1 are varied between between 2 road users and 11 road users:

$$x_1^{max} = x_2^{max} = 2, 3, ..., 11.$$

In Figure 9.8 we can see that the average delay of a road user is about 9 seconds except when the maximum queue length is close to 2.

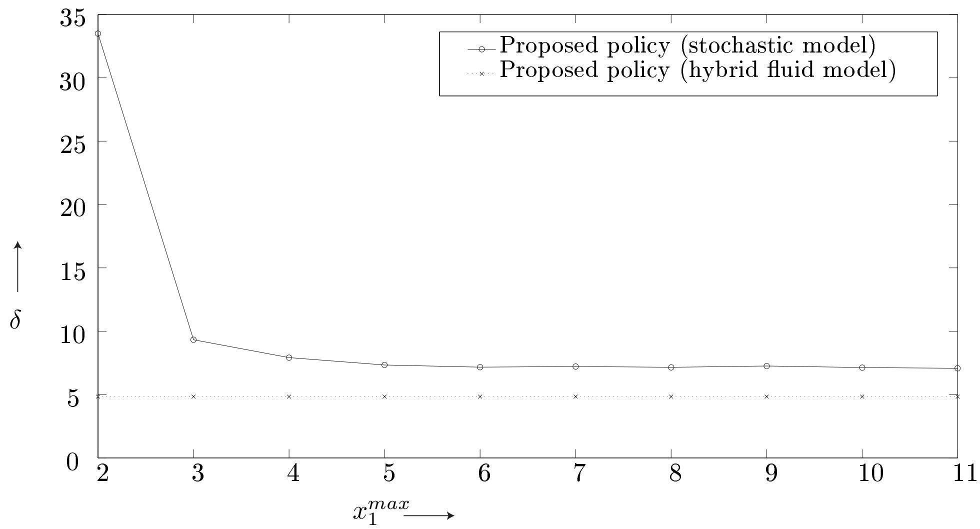


Figure 9.8: The average delay of a road user δ versus $x_1^{max} = x_2^{max}$ for test case 3b.

Chapter 10

Conclusions and Recommendations for Further Research

In this chapter we state the most important conclusions of this master thesis, thereafter we state our recommendations for further research

10.1 Conclusions

In this master thesis we have considered the following three problems for intersections with two conflict groups.

1 Trajectory optimization: finding an optimal trajectory minimizing the average weighted queue length at an intersection.
2 Regulation: finding a set of rules (a policy) that defines when to switch the state of a traffic light.
3 Addressing the quality of the proposed policy in a stochastic setting.

We state our most important conclusions for these three problems in sections 10.1.1–10.1.3.

10.1.1 Trajectory optimization

To solve the trajectory optimization problem we modeled intersections with a hybrid fluid model. This hybrid fluid model assumes deterministic arrivals and departures.

For an intersection with two signal groups and using assumptions (7.2.3) we derived that we can w.l.o.g. assume that an optimal trajectory, minimizing the average weighted queue length at an intersection, satisfies the following properties:

- always the highest possible departure rate is used during a green period of signal $i \in \mathcal{N}$:

$$d_i(t) = \begin{cases} \mu_i & \text{if } x_i(t) > 0, \\ \lambda_i & \text{if } x_i(t) = 0. \end{cases}$$

- a queue is always emptied during its green period
- green periods always take equally long

- a signal is green as long as possible during a red period of a conflicting signal:

$$r_{i_2} = g_{i_1} + \sigma_{i_1,i_2,i_1}, \quad \forall i_1 \in \mathcal{G}_1, \quad \forall i_2 \in \mathcal{G}_2,$$
$$r_{i_1} = g_{i_2} + \sigma_{i_1,i_2,i_1}, \quad \forall i_1 \in \mathcal{G}_1, \quad \forall i_2 \in \mathcal{G}_2,$$

where, g_i respectively r_i is the duration of all green times of signal $i \in \mathcal{N}$ and the duration of all red times of signal $i \in \mathcal{N}$.

Using these properties, we proposed an optimization problem which we could solve (analytically) for two classes of intersections (see Section 7.5.3).

10.1.2 Regulation

A policy is proposed for an intersection with two signal groups. For this policy we try to serve the signals in a signal group long enough to satisfy conditions 1.1–1.3

1.1 all queues in this signal group are (expected to be) emptied during their green period (assuming a hybrid fluid model).
1.2 all signals are served for at least the minimum green time.
1.3 the queue length of a signal i in the other signal group satisfies $x_i(t) \geq x_i^\sharp$.

It might not be possible to serve the signals in the set long enough to satisfy conditions 1.1–1.3. We might have to switch earlier because otherwise a maximum green time or a maximum queue length (assuming a hybrid fluid model) is exceeded.

We have proven that when the intersection is modeled with a hybrid fluid model, trajectories converge to the desired trajectory (derived with the trajectory optimization problem) if started from a feasible starting point.

10.1.3 Addressing the Quality of the proposed policy

We have tested the proposed policy on several test cases. For these test cases we varied the arrival rates, maximum green times and maximum queue lengths. For intersections with two conflicting signals the proposed policy is compared to an exhaustive policy. The proposed policy works better than the exhaustive policy for smaller arrival rates. For large arrival rates the exhaustive policy works better.

Further, for an intersection with two signals the proposed policy works better than than the exhaustive policy if most of the traffic arrives at one of the signals (asymmetrical arrival rates). In this case slow modes at the high traffic signal are desirable because all traffic arriving during a slow mode crosses the intersection without delay.

For the proposed policy, a signal could have slow modes in a stochastic environment even if this signal does not have any slow modes for the desired trajectory (derived with the trajectory optimization problem).

10.2 Recommendations for Further Research

Here we state our recommendations for further research on the topics treated in this thesis.

10.2.1 Comparison of proposed policy to existing policies

In this thesis the proposed policy is only compared to the exhaustive policy for an intersection of two signals (see Section 6). To address the quality of the proposed policy, this proposed policy can be compared to other existing policies. Further, the proposed policy has to be tested for intersections with more than 4 signals.

We can compare the proposed policy to the policy proposed in Newell and Osuna [22] for an intersection of two two-way streets. In [22] it is proposed to switch both signals in a sign group to red at the moment that both queues are cleared. Further, we can compare the proposed policy to the policy proposed in Haijema and van der Wal [12]. In [12] the decision when to switch and which signals to serve next is modeled as a Markovian decision process. Furthermore, the policy can be compared to a fixed cycle traffic light control.

10.2.2 Improving our Policy

It can occur that the policy proposed in this thesis does not satisfy the restriction on minimum green time duration in a stochastic setting. In a stochastic setting the proposed policy might switch a signal to red before the minimum green time has elapsed whenever the queue of a conflicting signal is close to its maximum queue length (see Section 6). For safety reasons, the policy should always satisfy these minimum green times. When changing (8.8) to the following equation we always satisfy the restrictions on green times:

$$(t_{1.1} \wedge t_{1.2} \wedge t_{1.3}) \vee t_2 \vee (t_3 \wedge t_{1.2}),$$

$t_{1.1}, t_{1.2}, t_{1.3}, t_2$ and t_3 remain as defined in 8.8

For the hybrid fluid model we only considered situations for which none of the constraints were violated. Hence, Chapter 5 and Chapter 8 are still valid for this altered policy. However, the outcome of test cases 3a and 3b in Chapter 6 and Chapter 9 will be different for this altered policy.

Furthermore, we observed that a signal could have a slow mode in a stochastic setting if it does not have a slow mode for the desired trajectory (for more information see Section 6). The cause is that we switch a signal only to red when a conflicting signal i satisfies $x_i(t) \geq x_i^\sharp$. We could adjust the policy by adjusting the definition of $x_{i_1}^\sharp$, $i_1 \in \mathcal{G}_1$ and $x_{i_2}^\sharp$, $i_2 \in \mathcal{G}_2$:

$$x_{i_1}^\sharp = \begin{cases} (r_{i_1} - \sigma_{i_1,N})\lambda_{i_1} & \text{if } \forall i_1 \in \mathcal{G}_1 \left(g_{i_1}^\lambda > 0\right), \\ 0 & \text{if } \exists i_1 \in \mathcal{G}_1 \left(g_{i_1}^\lambda = 0\right). \end{cases}$$

$$x_{i_2}^\sharp = \begin{cases} (r_{i_2} - \sigma_{i_2,1})\lambda_{i_2} & \text{if } \forall i_2 \in \mathcal{G}_2 \left(g_{i_2}^\lambda > 0\right), \\ 0 & \text{if } \exists i_2 \in \mathcal{G}_2 \left(g_{i_2}^\lambda = 0\right). \end{cases}$$

Note that this new definition for $x_i^\sharp$, $i \in \mathcal{N}$ differs from the definition used in this thesis only when a signal in the same signal group has no slow mode for the desired trajectory (derived via trajectory optimization).

For these new definitions the proof of Proposition 8.1 is not entirely valid anymore and has to be adjusted.

10.2.3 Setup Times

In this thesis we assumed Non-negative setup times:

$$\sigma_{i_1,i_2} \geq 0, \quad i_1 \in \mathcal{G}_1, i_2 \in \mathcal{G}_2,$$
$$\sigma_{i_2,i_1} \geq 0, \quad i_1 \in \mathcal{G}_1, i_2 \in \mathcal{G}_2,$$

For further research, we might drop this assumption. The proofs of Lemma 4.1 and Lemma 4.2 are still valid when dropping the assumption of positive setup times. The proofs of lemmas 7.1–7.3 need some (minor) modifications. Further, the shape of the optimal trajectory given in Corollary 7.4 does not hold anymore because for negative green times two conflicting signals can be green at the same time.

The policy that is proposed in Section 8 switches a signal to green only if all conflicting signals are red. If a setup time is negative, two conflicting signals can be green at the same time. Hence, a new policy must be proposed.

Further, in this thesis we assumed that:

$$\sigma_{i_1,i_2} - \sigma_{i_1,l_2} = \sigma_{l_1,i_2} - \sigma_{l_1,l_2}, \qquad \forall i_1, l_1 \in \mathcal{G}_1 \qquad i_2, l_2 \in \mathcal{G}_2 \tag{10.1}$$
$$\sigma_{i_2,i_1} - \sigma_{i_2,l_1} = \sigma_{l_2,i_1} - \sigma_{l_1,l_2}, \qquad \forall i_1, l_1 \in \mathcal{G}_1 \qquad i_2, l_2 \in \mathcal{G}_2 \tag{10.2}$$

This assumption was needed in the proof of Lemma 7.1. Hence, when dropping this assumption Lemma 7.1 is not valid anymore. Lemma 7.2 is still valid. However, when dropping assumptions (10.1) and (10.2), we have to formally prove that the alternative trajectory satisfies behavioral equations (7.3g) and (7.3h).

10.2.4 Multiple Signal Groups

In this thesis we have considered intersections with two signal groups. In practice, often more signal groups are needed. When considering more than two signal groups lemmas 7.1–7.3 have to be adjusted. Furthermore, the order in which these signal groups are served has to be determined and a new policy has to be derived for the case of more than two signal groups.

10.2.5 Networks of Intersections

In this thesis we considered isolated intersections; the arrival rates where assumed to be constant. For a network of intersections these arrival rates are not constant and so called platoons can arise. A first step towards deriving optimal trajectories for a network of intersections is to consider an isolated intersection with piecewise constant arrivals. A possible starting point might be the research done by van Eekelen in [10]. In Section 5.8 of [10], an intersection with two conflicting signal with piecewise constant arrivals is considered. In Section 5.8 of [10] no constraints on green times and maximum queue lengths are considered.

Appendix A

Table of Symbols (Used in χ3.0 Simulation Code

A.1 Types

In this section we give the newly defined types used in the χ simulation program.

Type	Definition
Trafficlight	The type 'trafficlight' contains information about a certain traffic light.
Trafficlight.mu	Maximum departure rate of this traffic light.
Trafficlight.lambda	Arrival rate of this traffic light.
Trafficlight.tif	Amount of time between the moment that this traffic light switches to red and the moment that the first conflicting traffic light switches to green
Trafficlight.tfi	Amount of time between the moment that the first traffic light in the same signal group switches to green and the moment that this traffic light switches to green.
Trafficlight.Xsharp	Value for $x_i^\sharp$ for this traffic light.
Trafficlight.Xmax	The maximum queue length of this trafficlight.
Trafficlight.gmin	Minimum green time of this traffic light.
Trafficlight.gmax	Maximum green time of this traffic light.
Trafficlight.ArrivalTime	List of arrival times of the road users waiting at this traffic light.
IntervalType	An interval consists out of a start time and an end time.
IntervalType.StartTime	Start time of an interval.
IntervalType.EndTime	End time of an interval.
Output1Type	Type used to store information that is written to the output file 'output1.txt'
Output1Type.AvgDelay	The average delay of a road user at the intersection
Output1Type.Overflow	List of the fraction of the time that the maximum queue length is exceeded for each of the traffic lights.

Output2Type	Type used to store information that is written to the output file 'output2.txt'. This information can be used to visualize the green periods, red periods and slow modes of each traffic lights. Furthermore, it can be used to plot the queue length at each of the traffic lights over time.
Output2Type.Green	Intervals of green periods for each of the traffic lights.
Output2Type.Red	Intervals of red periods for each of the traffic lights.
Output2Type.Slowmode	Intervals of green periods for each of the traffic lights.
Output2Type.X	List with queue lengths for each of the traffic lights. This list can be used to plot the queue length as function of time for each traffic light (together with Output2Type.Time).
Output2Type.Time	List with times for the traffic lights. This list can be used to plot the queue length as function of time for each traffic light (together with Output2Type.X).
tlControlInfoType	This type contains the input that the policy needs and the output that the policy returns.
tlControlInfoType.SetServed	The index (either 0 for signal group 1 and 1 for signal group 2) of the signal group that is currently served.
tlControlInfoType.SetNotServed	The index (either 0 for signal group 1 and 1 for signal group 2) of the signal group that is currently not served.
tlControlInfoType.tlServedFirst	This integer refers to the traffic light in the set 'SetServed' that (always) switches to green first (of all traffic lights in the set 'SetServed').
tlControlInfoType.tlSwitchedFirst	This integer refers to the traffic light in the set 'SetServed' that (always) switches to red first (of all traffic lights in the set 'SetServed').
tlControlInfoType.ToLS	Time of the most recent time that the traffic light tlServedFirst switched from red to green or that the traffic light tlSwitchedFirst switched from green to red.
tlControlInfoType.Switched	Time of the most recent time that the traffic light tlServedFirst switched from red to green or that the traffic light tlSwitchedFirst switched from green to red.
tlControlInfoType.Switched	This boolean is true whenever the the signal tlSwitchedFirst is red.
tlControlInfoType.Green	List with the state of each of the traffic lights: True whenever a traffic light is green and False whenever a traffic light is red.

A.2 Symbols

In this section we give the most important symbols used in the χ simulation program.

Symbol	Definition
AvgDelayHFM	The average delay of a road user obtained via trajectory optimization. This variable is read from an input file.
SimDelay	The average delay of a road user obtained for a single run.
ListSimDelay	List of average delays obtained for different runs of a test case.
SimFracOverflow	List with the fraction of overflow (fraction of time that a queue length is exceeded) for each of the traffic lights for a single run.
ListSimFracOverflow	List of the variables 'SimFracOverFlow' obtained for different runs of a test case.
SimDuration	The duration that is simulated during a run.
StartSimTime	The simulated time keeps increasing for a chi simulation. StartSimTime is the duration that is simulation when a run start. This variable is used to determine the simulated time since the start of a run.
FirstRun	Only during the first run of a test case we write information to output2.txt. FirstRun is a boolean that is true whenever it is the first run of a test case.
G	G[0] contains the indices of the signals in signal group 1 and G[1] contains the indices of the signals in signal group 2.
N	= Number of traffic lights at the intersection.
NumOfDepartures	Number of road users that have crossed the intersection.
DepProcStarted	This variable is true whenever a departure process is started.

Appendix B

χ 3.0 Simulation Code

```
# This simulation considers a traffic intersection with two signal groups where the traffic light control is subject to:
#    - maximum and minimum green times
#    - maximum queue lengths. This is a soft constraint; a queue could exceed the maximum queue length but we try to avoid it.
#   We keep track of the fraction of the time that this maximum queue length is exceeded.
#  - Clearance/setup times. When traffic light i and traffic light j are conflicting (may not be green at the same time) then traffic
#   light i must be red for a certain amount of time before traffic light j may turn green (and vice versa).

# In this simulation we try to follow periodic behavior using a policy
# a policy is a set of rules that specify when to switch traffic lights from green to red and from red to green.



#######################################################
##################Defining types #####################
#######################################################

type trafficlight = tuple(list real ArrivalTime; real mu; real lambda; real tif; real tfi; real Xsharp; real Xmax; real gmin; real gmax);

type IntervalType = tuple(list real StartTime; list real EndTime);

type Output1Type = tuple(real AvgDelay; list real FracOverflow);

type Output2Type = tuple(list IntervalType Green; list IntervalType Red; list IntervalType Slowmode; list list int X; list list real Time);

type tlControlInfoType = tuple(int SetServed; int SetNotServed; int tlServedFirst; int tlSwitchedFirst; real ToLS; bool Switched; list bool Green);

#######################################################
###################### Model ########################
#######################################################

model intersection():
  # Variable declaration
  real stddev;
```

```
real SimDelay;
real AvgDelayHFM;
real StartSimTime;
real SimDuration;
real real_temp;
real Width95IntSimDelay = 0.0;
real AvgSimDelay = 0.0;
list real ListSimDelay;
list real SimFracOverflow;
list list real ListSimFracOverflow;

int NumberOfRuns = 0;
int NumberOfTestcases;
int id;
int N;
list(2) set int G;

bool FirstRun;

trafficlight tl_temp;
list trafficlight tl;

chan real chan_Delay;
chan list real chan_Overflow;
chan int a_temp, d_temp;
list chan int a,d;
chan void t;

file InputFile,Output1File, Output2File;

# Opening input and output files
InputFile = open("input.txt","r");
Output1File = open("Output1.txt","w");
Output2File = open("Output2.txt","w");

# Read the number of test cases
NumberOfTestcases = read(InputFile,int);

for k in range(NumberOfTestcases):

  # emptying some lists for the new test case
  a = a[1:1];
  d = d[1:1];
  SimFracOverflow = SimFracOverflow[1:1];
  ListSimFracOverflow = ListSimFracOverflow[1:1];
  ListSimDelay = ListSimDelay[1:1];

  # resetting some variables for the new testcase
  Width95IntSimDelay = 0.0;
  AvgSimDelay = 0.0;

  # read all inputs for this testcase from the input file.
  (id, AvgDelayHFM, SimDuration, N, G, tl) = ReadInput(InputFile);

  # write some of the information to outputfiles
  write(Output1File, "%s \t", id);
  write(Output1File, "%s \t", AvgDelayHFM);
  write(Output1File, "%s \t", N);
  write(Output2File, "%s \t", id);
  write(Output2File, "%s \t", N);

```

```
    # Making sure that lists have the right size
    for i in range(N):
      a = a + [a_temp];
      d = d + [d_temp];
      SimFracOverflow = SimFracOverflow + [real_temp];
    end;

    FirstRun =  true;
    NumberOfRuns = 0;


    while Width95IntSimDelay >= 0.01*AvgSimDelay or NumberOfRuns < 100:

      StartSimTime = time;         # The time at which we start a run.
      # We start a run
      start buffer(StartSimTime, SimDuration, tl, G, FirstRun, chan_Delay, chan_Overflow, a, d, t, Output1File, Output2File);

      # We obtain the results of this run via channels
      chan_Delay?SimDelay;
      chan_Overflow?SimFracOverflow;

      # add the obtained results of this run to arrays that contain the results for all runs of this testcase
      ListSimDelay = ListSimDelay + [SimDelay];
      ListSimFracOverflow = ListSimFracOverflow + [SimFracOverflow];

      # From all runs of this testcase obtain the average delay of a vehicle at the intersection.
      AvgSimDelay = mean(ListSimDelay);
      # From all runs of this testcase obtain the width of the 95% confidence interval of the average delay.
      stddev = StdDev(ListSimDelay);
      Width95IntSimDelay = 2*1.960*stddev/size(ListSimDelay);

      FirstRun =  false;           # We already performed at leatst one run
      NumberOfRuns = NumberOfRuns + 1;  # The number of runs is increased with one.

      # writing information to the screen
      write("Simulating case %s \n", k+1);
      write("Number of runs %s \n", NumberOfRuns);
      write("average Delay %s \n", AvgSimDelay);
      write("width of 95-confidence interval %s \n", Width95IntSimDelay);
    end

    # write information obtained for a testcase to the file Output1File
    write(Output1File,"%s\t",size(ListSimDelay));
    for i in range(size(ListSimDelay)):
      write(Output1File,"%s\t",ListSimDelay[i]);
    end
    for i in range(N):
      for j in range(size(ListSimFracOverflow)):
        write(Output1File,"%s\t",ListSimFracOverflow[j][i]);
      end
    end
    write(Output1File,"\n");
  end;
end

#######################################################
##################### Processes ######################
```

```
##########################################################

proc buffer(real StartSimTime; real SimDuration; list trafficlight tl; list set int G; bool FirstRun; chan! real chan_Delay; chan! list real chan_Overflow; list chan? int a,d; chan void t; file Output1File; file Output2File):

  # Variable declaration
  int N = size(tl);
  int k;
  int NumOfDepartures = 0;
  list (N) list int list_int_temp;

  real deltaT = 0.001; # We sample the feedback of the controller every deltaT seconds.
  list(N) real real_temp;
  list (N) list real list_real_temp;

  list(N) bool DepProcStarted;
  list(N) bool PreviousGreen;
  list(N) bool SlowmodeStarted;
  list(N) bool bool_temp;

  Output1Type Output1;
  Output2Type Output2;

  tlControlInfoType tlControlInfo;

  list(N) IntervalType interval_temp;


  # making lists the right size
  tlControlInfo.Green = bool_temp;
  Output1.FracOverflow = real_temp;

  Output2.Green = interval_temp;
  Output2.Red = interval_temp;
  Output2.Slowmode = interval_temp;

  Output2.X = list_int_temp;
  Output2.Time = list_real_temp;

  # Initialization:
  (tlControlInfo, Output2, SlowmodeStarted) = Initialization(G, FirstRun,tl, Output2, SlowmodeStarted, tlControlInfo);
  # We start with the situation where all traffic lights in set 1 (G[0]) are green and all traffic lights in set 2 (G[1]) are red

  # Starting departure processes for the traffic lights that are green and whose queues are non-empty.
  # When a traffic light is green and its queue is empty then traffic can immediately cross the intersection (without delay).
  for i in G[0]:
    if size(tl[i].ArrivalTime)> 0:
      start departure(tl[i].mu,i,d);
      DepProcStarted[i] = true;
    else:
      DepProcStarted[i] = false;
    end
  end;
  for i in G[1]: # All traffic lights in set 2 (G[1]) are red. Traffic cannot depart and we start a red period
    DepProcStarted[i] = false;
```

```
  end;

  # Start arrival processes for all traffic lights
  for i in range(N):
    start arrival(tl[i].lambda,i,a);
  end

  # Start the timer process. This process sends a signal every deltaT seconds.
  start timerprocess(deltaT,t);


  while (time - StartSimTime) < SimDuration:
    select
      # Sampling the Controller decision whenever we receive a void from channel t.
      t?: PreviousGreen = tlControlInfo.Green;
        tlControlInfo = CalcGreenTls(tl,G,time-StartSimTime ,tlControlInfo);
        if FirstRun: # Some information we will only gather during the first run of a testcase
          (Output2, SlowmodeStarted) = UpdateOutput2(PreviousGreen, tlControlInfo.Green, Output2, tl, time - StartSimTime);
        end

        # If a traffic light is green and its queue is not empty then we start a departure process (if it was not already started)
        for i in G[tlControlInfo.SetServed]:
          if tlControlInfo.Green[i] and not DepProcStarted[i] and size(tl[i].ArrivalTime)>0:
            start departure(tl[i].mu,i,d);
            DepProcStarted[i] = true;
          end
        end;

        # We again start the timerprocess
        start timerprocess(deltaT,t);
    alt
      unwind j in range(N):
        a[j]?k: # If we receive a signal via channel a[j] then a vehicle arrives at traffic light j
          start arrival(tl[k].lambda,k,a); # We again start a new arrival process

          # The arrived vehicle is only added to the queue when the queue is not empty
          # or the traffic light is red. We assume that a vehicle arriving when
          # the queue is empty and the traffic light is green can immediately
          # cross the intersection.
          if size(tl[k].ArrivalTime)>0 or not tlControlInfo.Green[k]:
            Output1.FracOverflow[k] = UpdateFracOverflow((time - StartSimTime), Output2.Time[k][-1], Output1.FracOverflow[k], size(tl[k].ArrivalTime), tl[k].Xmax);
            if FirstRun:  # Some information we will only gather during the first run of a test case
              Output2.X[k] = Output2.X[k] + [size(tl[k].ArrivalTime)];
              Output2.Time[k] = Output2.Time[k] + [time - StartSimTime];
            end
            tl[k].ArrivalTime = tl[k].ArrivalTime + [time - StartSimTime];  # Store the time of arrival
            Output2.X[k] = Output2.X[k] + [size(tl[k].ArrivalTime)];    # Update the queue length
            Output2.Time[k] = Output2.Time[k] + [time - StartSimTime];    # Update the time of the last change in queue length
          elif size(tl[k].ArrivalTime) == 0 and tlControlInfo.Green[k]:     # If a vehicle arrived when the traffic light was green
```

```
                                            # and the queue was empty then this vehicle experienced no delay
          NumOfDepartures = NumOfDepartures + 1;
          Output1.AvgDelay = UpdateAvgDelay(NumOfDepartures, Output1.AvgDelay, 0.0);
        end
      end
  alt
    unwind l in range(N):
      d[l]?k: # If we receive a signal via channel d[j] then a vehicle has just departed at traffic light j
        # Update the average delay
        NumOfDepartures = NumOfDepartures + 1;
        Output1.AvgDelay = UpdateAvgDelay(NumOfDepartures, Output1.AvgDelay, (time - StartSimTime - tl[k].ArrivalTime[0]));
        Output1.FracOverflow[k] = UpdateFracOverflow((time - StartSimTime), Output2.Time[k][-1], Output1.FracOverflow[k], size(tl[k].ArrivalTime), tl[k].Xmax);

        if FirstRun:  # Some information we will only gather during the first run of a testcase
          Output2.X[k] = Output2.X[k] + [size(tl[k].ArrivalTime)];
          Output2.Time[k] = Output2.Time[k] + [time - StartSimTime];
        end
        tl[k].ArrivalTime = tl[k].ArrivalTime[1:];          # Erase the first element of the array 'ArrivalTime'
                                        # (the vehicle that corresponds to the erased element has just departed).
        Output2.X[k] = Output2.X[k] + [size(tl[k].ArrivalTime)];  # Update the queue length
        Output2.Time[k] = Output2.Time[k] + [time - StartSimTime];  # Update the time of the last change in queue length
        if  tlControlInfo.Green[k] and size(tl[k].ArrivalTime)>0:   # Again start a departure process whenever the traffic light is green and
                                        # the queue length is non-zero
          start departure(tl[k].mu,k,d);
          DepProcStarted[k] = true;
        else:
          DepProcStarted[k] = false;
        end
        if FirstRun:  # Some information we will only gather during the first run of a test case
          if size(tl[k].ArrivalTime) == 0 and tlControlInfo.Green[k] and not SlowmodeStarted[k]: # If the queue length is zero and the traffic light
                                                        # is green then a slow mode starts (if it was not already started).
            SlowmodeStarted[k] = true;
            Output2.Slowmode[k].StartTime = Output2.Slowmode[k].StartTime + [time-StartSimTime];
          end
        end
    end
  end;
end;


# let all started departure processes finish:
for i in range(N):
  while DepProcStarted[i]: # for some reason it does not work with an if loop
    d[i]?k;
    DepProcStarted[i] = false;
```

```
    end
  end

  # let all started arrival processes finish:
  for i in range(N):
    a[i]?k;
  end
  # let the timer process finish:
  t?;

  if FirstRun:
  Output2 = Write2Output2File(Output2File, Output2, tl, time - StartSimTime); # This function writes information to the file 'Output2File'.
  # We received 'Output2' from the function. However 'Output2' has not changed. A function must always return something.
  end

  chan_Delay!Output1.AvgDelay;
  chan_Overflow!Output1.FracOverflow;
end

# The timerprocess sends a void signal after deltaT seconds
proc timerprocess(real deltaT; chan! void t):
  delay deltaT;
  t!
end

# The process 'arrival' sends a interger i over chanel a[i] after exponentially distributed amount of time has elapsed.
# When such a signal is send this means that a vehicle has arrived at traffic light 'i'.
proc arrival(real lambda; int i; list chan! int a):
  dist real interarrivaltimedist = exponential(1/lambda);
  real interarrivaltime;

  interarrivaltime = sample(interarrivaltimedist);
  delay interarrivaltime;
  a[i]!i;
end

# The process 'departure' sends a integer i over channel a[i] after 1/mu seconds has elapsed.
# When such a signal is send this means that a vehicle has departed at traffic light 'i'.
proc departure(real mu; int i; list chan! int d):
  delay 1/mu;
  d[i]!i;
end

#####################################################
##################### Functions #####################
#####################################################

func tuple(int id; real AvgDelayHFM; real SimDuration; int N; list(2) set int G; list trafficlight tl) ReadInput(file InputFile):

  # Variable declaration
  real AvgDelayHFM;
  real SimDuration;

  int X0_temp;
```

```
int id;
int N, N1, N2;
list int X0;
list(2) set int G;

trafficlight tl_temp;
list trafficlight tl;

# Read the id of the test case
id = read(InputFile, int);

# Read the average delay for this test case that was obtained via the Hybrid Fluid Model (calculated with matlab)
AvgDelayHFM = read(InputFile, real);

# Read the simulation time of a single run
SimDuration = read(InputFile, real);

# Read the number of traffic lights in set 1 (N1) and the number of traffic lights in set 2 (N2)
N1 = read(InputFile, int);
N2 = read(InputFile, int);
N = N1+N2;

# The first N1 traffic lights (0,...,N1) are in set 1 (G[0])
for i in range(N1):
  G[0] = G[0] + {i};
end;

# The other traffic lights (N1+1,...,N) are in set 2 (G[1])
for i in range(N1,N):
  G[1] = G[1] + {i};
end

# Making sure that lists have the right size
for i in range(N):
  tl = tl + [tl_temp];
  X0 = X0 + [X0_temp];
end;

# Reading the initial queue lengths
for i in range(N):
  X0[i] = read(InputFile, int);
end

# At the start there are already X0[i] vehicles waiting in front of traffic light i. We assume these vehicles have arrived at time 0.0.
for i in range(N):
  for j in range(X0[i]):
    tl[i].ArrivalTime = tl[i].ArrivalTime + [0.0];
  end
end

# Reading the maximum departure rates
for i in range(N):
  tl[i].mu = read(InputFile, real);
end

# Reading the arrival rates
for i in range(N):
  tl[i].lambda = read(InputFile, real);
```

```
  end

  # tl[i].tif: Always when traffic light i has been red for tl[i].tif seconds, the
      first traffic light in the other set is switched to green.
  for i in range(N):
    tl[i].tif = read(InputFile, real);
  end

  # tl[i].tfi: traffic light i is switched to green tl[i].tfi seconds after the first
      traffic light in the same set is switched to green.
  for i in range(N):
    tl[i].tfi = read(InputFile, real);
  end

  # Xsharp (X^{#}) is needed for the controller. See ??????? for more information about
       Xsharp
  for i in range(N):
    tl[i].Xsharp = read(InputFile, real);
  end

  # Reading the maximum queue lengths
  for i in range(N):
    tl[i].Xmax = read(InputFile, real);
  end

  # Reading the minimum green times
  for i in range(N):
    tl[i].gmin = read(InputFile, real);
  end

  # Reading the minimum green times
  for i in range(N):
    tl[i].gmax = read(InputFile, real);
  end;

  return (id, AvgDelayHFM, SimDuration, N, G, tl)
end

func tuple(tlControlInfoType tlControlInfo; Output2Type Output2; list bool
    SlowmodeStarted) Initialization(list(2) set int G; bool FirstRun; list trafficlight
     tl; Output2Type Output2; list bool SlowmodeStarted; tlControlInfoType
    tlControlInfo):

  tlControlInfo.Switched = false; # We start with the situation where all traffic light
       in group 1 are green.
  tlControlInfo.ToLS = 0.0;
  tlControlInfo.SetServed = 0; # We start serving set 1 (G[0])
  tlControlInfo.SetNotServed = 1;

  for i in G[0]:
    tlControlInfo.Green[i] = true; # the traffic lights in set 1 (G[0]) are switched to
         green
  end

  for i in G[1]:
    tlControlInfo.Green[i] = false; # the traffic lights in set 2 (G[1]) are switched
         to red
  end

  # Calculating the traffic lights in the set 1 (G[0]) that is the first traffic light
      (in the set 1) to switch to green.
```

```
  tlControlInfo.tlServedFirst = CalcServedFirst(G[0], tl);
  # Calculating the traffic light in the set 1 (G[0]) that is the first traffic light (in the set 1) to switch to red
  tlControlInfo.tlSwitchedFirst = CalcServedFirst(G[0], tl);

  for i in range(size(tl)):
    Output2.X[i] = Output2.X[i] + [size(tl[i].ArrivalTime)]; # The initial queu length
    Output2.Time[i] = Output2.Time[i] + [0.0];          # The initial time
  end

  # Initialization of all other information in Output2.
  if FirstRun:
    (Output2, SlowmodeStarted) = InitializationOutput2Info(G, tl, Output2, SlowmodeStarted);
  end

  return (tlControlInfo, Output2, SlowmodeStarted)
end

func tuple(Output2Type Output2; list bool SlowmodeStarted) InitializationOutput2Info(list(2) set int G; list trafficlight tl; Output2Type Output2; list bool SlowmodeStarted):

  # We start with the situation where all traffic lights in set 1 (G[0]) are green and all traffic lights in set 2 (G[1]) are red
  for i in G[0]:
    Output2.Green[i].StartTime = Output2.Green[i].StartTime + [0.0];  # All traffic lights in set 1 start a green period.
    if size(tl[i].ArrivalTime)> 0: # If the queue (in set 1) is not empty at the start then we start the departure process
      SlowmodeStarted[i] = false;
    else: # If the queue (in set 1) is empty at the start then this is the start of a slowmode
      SlowmodeStarted[i] = true;
      Output2.Slowmode[i].StartTime = Output2.Slowmode[i].StartTime + [0.0];
    end
  end;
  for i in G[1]: # All traffic lights in set 2 (G[1]) are red. Traffic cannot depart and we start a red period
    SlowmodeStarted[i] = false;
    Output2.Red[i].StartTime = Output2.Red[i].StartTime + [0.0];
  end

  return (Output2, SlowmodeStarted)
end

func tuple(Output2Type Output2; list bool SlowmodeStarted) UpdateOutput2(list bool PreviousGreen; list bool Green; Output2Type Output2; list trafficlight tl; real CurrentTime):

  # Variable declaration
  int N = size(Green);
  list(N) bool SlowmodeStarted;

  for i in range(N):
    if PreviousGreen[i] and not Green[i]: # If a traffic light was green and is now red then this is the end of a green period and the start of a red period.
      Output2.Red[i].StartTime = Output2.Red[i].StartTime + [CurrentTime];
      Output2.Green[i].EndTime = Output2.Green[i].EndTime + [CurrentTime];
```

```
      if size(Output2.Slowmode[i].StartTime) > size(Output2.Slowmode[i].EndTime): # When a slowmode started during the previous green time, the slowmode is ended.
        Output2.Slowmode[i].EndTime = Output2.Slowmode[i].EndTime + [CurrentTime];
        SlowmodeStarted[i] = false;
      end
    elif not PreviousGreen[i] and Green[i]: # If a traffic light was red and is now green then this is the end of a red period and the start of a green period.
      Output2.Green[i].StartTime = Output2.Green[i].StartTime + [CurrentTime];
      Output2.Red[i].EndTime = Output2.Red[i].EndTime + [CurrentTime];
      if size(tl[i].ArrivalTime)== 0 and not SlowmodeStarted[i]: # If a traffic light is empty at the beginning of its green period, a slowmode is started.
        SlowmodeStarted[i] = true;
        Output2.Slowmode[i].StartTime = Output2.Slowmode[i].StartTime + [CurrentTime];
      end
    end
  end

  return (Output2, SlowmodeStarted)
end

func Output2Type Write2Output2File(file Output2File; Output2Type Output2; list trafficlight tl; real CurrentTime):

  int N = size(tl); # Number of traffic lights

  for i in range(N):
    Output2.X[i] = Output2.X[i] + [size(tl[i].ArrivalTime)];
    Output2.Time[i] = Output2.Time[i] + [CurrentTime];
  end

  # for all traffic lights we first write the number of green periods of the traffic light to the file 'Output2File'.
  # Herafter we write all the start times of these green periods to the file 'Output2' followed by all the end times of these green periods
  for i in range(N):
    if size(Output2.Green[i].StartTime) > size(Output2.Green[i].EndTime): # A green period is not finished yet
      Output2.Green[i].EndTime = Output2.Green[i].EndTime + [CurrentTime];
    end
    write(Output2File, "%s \t", size(Output2.Green[i].StartTime));
    for j in range(size(Output2.Green[i].StartTime)):
      write(Output2File, "%s \t", Output2.Green[i].StartTime[j]);
    end
    for j in range(size(Output2.Green[i].StartTime)):
      write(Output2File, "%s \t", Output2.Green[i].EndTime[j]);
    end
  end;

  # for all traffic lights we first write the number of red periods of this traffic light to the file 'Output2File'.
  # Herafter we write all the start times of these red periods to the file  followed by all the end times of these red periods to the file 'Output2File'.
  for i in range(N):
    if size(Output2.Red[i].StartTime) > size(Output2.Red[i].EndTime): # A green period is not finished yet
      Output2.Red[i].EndTime = Output2.Red[i].EndTime + [CurrentTime];
    end
    write(Output2File, "%s \t", size(Output2.Red[i].StartTime));
    for j in range(size(Output2.Red[i].StartTime)):
      write(Output2File, "%s \t", Output2.Red[i].StartTime[j]);
```

```
    end
    for j in range(size(Output2.Red[i].StartTime)):
      write(Output2File, "%s \t", Output2.Red[i].EndTime[j]);
    end
  end

  # for all traffic lights we first write the number of slowmodes of this traffic light
     to the file 'Output2File'.
  # Herafter we write all the start times of these slowmodes to the file followed by
     all the end times of these slowmodes to the file 'Output2File'.
  for i in range(N):
    if size(Output2.Slowmode[i].StartTime) > size(Output2.Slowmode[i].EndTime): # A
       slowmode is not finished yet
      Output2.Slowmode[i].EndTime = Output2.Slowmode[i].EndTime + [CurrentTime];
    end
    write(Output2File, "%s \t", size(Output2.Slowmode[i].StartTime));
    for j in range(size(Output2.Slowmode[i].StartTime)):
      write(Output2File, "%s \t", Output2.Slowmode[i].StartTime[j]);
    end
    for j in range(size(Output2.Slowmode[i].StartTime)):
      write(Output2File, "%s \t", Output2.Slowmode[i].EndTime[j]);
    end
  end

  # For all traffic lights write the queue lengths to the file 'Output2File' and
      hereafter write the times corresponding to these queue lengths to the file '
      Output2File'
  for i in range(N):
    write(Output2File, "%s \t", size(Output2.X[i]));
    for j in range(size(Output2.X[i])):
      write(Output2File, "%s \t", Output2.X[i][j]);
    end
    for j in range(size(Output2.X[i])):
      write(Output2File, "%s \t", Output2.Time[i][j]);
    end
  end

  # Go to a new line for the next testcase.
  write(Output2File, "\n");

  return Output2
end

# Calculate the traffic light in the set G that is switched to green first (of the
    traffic lights in the set G).
# The traffic light that satisfies tf[i].tfi = 0.0 is the first traffic light in the
    set G to switch to green.
# This because tl[i].tfi seconds after the first traffic light has switched to green
    traffic light i switches to green.
func int CalcServedFirst(set int G; list trafficlight tl):
  int tlServedFirst;

  for i in G:
    if tl[i].tfi  == 0.0:
      tlServedFirst=i;
    end
  end;

  return tlServedFirst
end

```

```
# Calculate the traffic light in the set G that is switched to red first (of the traffic lights in the set G).
# The traffic light in the set G that has the largest value for tf[i].tif is the first traffic light in the set G to switch to red.
# This because tl[i].tif seconds after traffic light i has switched to red the first traffic light in the other set is switched to green.
func int CalcSwitchedFirst(set int G; list trafficlight tl):
  int tlSwitchedFirst;
  real Maxtif = 0.0;

  for i in G:
    if tl[i].tif >= Maxtif:
      tlSwitchedFirst=i;
      Maxtif = tl[i].tif;
    end
  end;

  return tlSwitchedFirst
end

# This function controls which of the traffic lights are green and which of the traffic lights are red
func tlControlInfoType CalcGreenTls(list trafficlight tl; list set int G; real Ctime; tlControlInfoType tlControlInfo):

  if tlControlInfo.Switched: # In this case we already switched the  'tlSwitchedFirst' to red.
    if (Ctime - tlControlInfo.ToLS) >= tl[tlControlInfo.tlSwitchedFirst].tif: # tl[tlControlInfo.tlSwitchedFirst].tif seconds after we switched the trafficlight 'tlSwitchedFirs' to red a trafficlight in the other set is switched to green.
      tlControlInfo.SetServed, tlControlInfo.SetNotServed = tlControlInfo.SetNotServed, tlControlInfo.SetServed;  # We change the set that is currently served
      tlControlInfo.tlSwitchedFirst = CalcSwitchedFirst(G[tlControlInfo.SetServed], tl);             # We calculate the traffic light (in the set that is currently served) that was switched to green the first (of the trafficlights in the set that is currently served).
      tlControlInfo.tlServedFirst = CalcServedFirst(G[tlControlInfo.SetServed], tl);                 # We calculate the traffic light (in the set that is currently served) that will be switched to red the first (of the trafficlights in the set that is currently served).
      tlControlInfo.ToLS = Ctime;     # The last time that the traffic light 'tlSwitchedFirst' was switched to red or the traffic light 'tlServedFirst' was switched to green.
      tlControlInfo.Switched = false;   # We have not yet switched the traffic light 'tlSwitchedFirst' to red.
    end
  else: # If we have not yet switched the traffic light 'tlSwitchedFirst' to red we evaluate whether we should switch the traffic light 'tlSwitchedFirst' to red.
    tlControlInfo.Switched = switch(tl, G, tlControlInfo.SetServed, tlControlInfo.SetNotServed, tlControlInfo.tlServedFirst, tlControlInfo.tlSwitchedFirst, Ctime - tlControlInfo.ToLS);
    if tlControlInfo.Switched:        # If we switch the first traffic light in the set SetServed to red, we change ToLS (Time of Last switch) ot the current time.
      tlControlInfo.ToLS = Ctime;     # The last time that the traffic light 'tlSwitchedFirst' was switched to red or the traffic light 'tlServedFirst' was switched to green.
    end
  end;

  for i in G[tlControlInfo.SetNotServed]: # Whenever a set is not served all the traffic lights in this set are red.
```

```
    tlControlInfo.Green[i] = false;
  end

  for i in G[tlControlInfo.SetServed]:
    if tlControlInfo.Switched:  #  (tl[tlControlInfo.tlSwitchedFirst].tif - tl[i].tif) seconds after'tlSwitchedFirst' switched to red, trafficlight i switches to red
      if (Ctime - tlControlInfo.ToLS) < (tl[tlControlInfo.tlSwitchedFirst].tif - tl[i].tif):
        tlControlInfo.Green[i] = true;
      else:
        tlControlInfo.Green[i] = false;
      end
    else: #  tl[i].tfi seconds after'tlServedFirst' switched to green, trafficlight i switches to green. During the first green time of set 1 (when tlControlInfo.ToLS = 0.0) all traffic lights in set 1 are green.
      if (Ctime - tlControlInfo.ToLS) >=  tl[i].tfi  or tlControlInfo.ToLS <= 0.00001:
        tlControlInfo.Green[i] = true;
      else:
        tlControlInfo.Green[i] = false;
      end
    end
  end;

  return tlControlInfo
end

 # With this function we evaluate whether we should switch the traffic light 'tlSwitchedFirst' to red if we have not yet switched the traffic light 'tlSwitchedFirst' to red.
 # For more information about when we switch the traffic light 'tlSwitchedFirst to red see ??????
func bool switch(list trafficlight tl; list set int G; int SetServed; int SetNotServed; int tlServedFirst; int tlSwitchedFirst; real tstar):
  bool b13 = false;

  # tstar is the time that has elapsed since the traffic light 'tlServedFirst' was switched to green.
  # If the maximum green time is exceeded then we switch the traffic light 'tlSwitchedFirst' to red.
  if tstar >= (tl[tlSwitchedFirst].gmax +  tl[tlSwitchedFirst].tfi):
    return true
  end

  # We switch the traffic light to red (that must be switched first) if otherwise a queue would exceed its maximum queue length (for a hybrid fluid model).
  # If we switch 'tlSwitchedFirst' to red then traffic light j (in the set that is not served) will be green (tl[tlSwitchedFirst].tif + tl[j].tfi) seconds
  for j in G[SetNotServed]:
    if size(tl[j].ArrivalTime) >= tl[j].Xmax - (tl[tlSwitchedFirst].tif + tl[j].tfi)*tl[j].lambda:
      return true
    end
  end

  # We also switch the traffic light 'tlSwitchedFirst' to red whenever conditions 1.1, 1.2 and 1.3 are satisfied

  # Is condition 1.3 satisfied?
  for j in G[SetNotServed]:
    if size(tl[j].ArrivalTime) >= tl[j].Xsharp and b13 == false:
      b13 = true;
```

```
    end
  end

  # If condition 1.3 is not satisfied and condition 2 and 3 are both not satisfied then
      we do not switch traffic light 'tlSwitchedFirst' to red
  if b13 == false:
    return false
  end

  # Condition 1.1 is satisfied whenever all traffic lights j in the set 'SetServed'
     satisfy size(tl[j].ArrivalTime) <= ((tl[tlSwitchedFirst].tif - tl[j].tif)*(tl[j].
     mu - tl[j].lambda) - max((tl[j].tfi -tstar),0.0))
  # Condition 1.2 is satisfied whenever all traffic lights j in the set 'SetServed'
     satisfy tstar > (tl[j].gmin + tl[j].tfi + tl[j].tif - tl[tlSwitchedFirst].tif).
  for j in G[SetServed]:
    if  size(tl[j].ArrivalTime) > ((tl[tlSwitchedFirst].tif - tl[j].tif)*(tl[j].mu - tl
        [j].lambda) - max((tl[j].tfi -tstar),0.0)) or tstar < (tl[j].gmin + tl[j].tfi +
         tl[j].tif - tl[tlSwitchedFirst].tif):
      # If condition 1.1 or 1.2 is not satisfied and condition 2 and 3 are both not
          satisfied then we do not switch traffic light 'tlSwitchedFirst' to red
      return false
    end
  end

  # if condition 1.1, 1.2 and 1.3 are all satisfied then we switch traffic light '
     tlSwitchedFirst' to red
  return true
end

# This Function updates the average delay whenever a vehicle has departed.
# About the input of this function:
# - NumOfDepartures is the number of Departures at the intersection (including the
   vehicle that has just departed)
# - AvgDelay is the average delay of the vehicles (excluding the vehicle that has just
   departed)
func real UpdateAvgDelay(int NumOfDepartures; real AvgDelay; real Delay):
  AvgDelay = (NumOfDepartures-1)/NumOfDepartures*AvgDelay + 1/NumOfDepartures*Delay; #
     updating average delay

  return AvgDelay
end


# This Function updates the fraction of time that a queue exceeded its maximum queue
   length.
func real UpdateFracOverflow(real CurrentTime; real TimeOfPreviousChange; real
   FracOverflow; int X; real Xmax):
  if X > Xmax:
    FracOverflow = (CurrentTime-TimeOfPreviousChange)/CurrentTime*FracOverflow +
        TimeOfPreviousChange/CurrentTime # updating fraction of overflow
  else:
    FracOverflow = (CurrentTime-TimeOfPreviousChange)/CurrentTime*FracOverflow
  end

  return FracOverflow
end

# Function calculating x squared
func real Square(real x):
  return x*x
end
```

```

# Calculating the average of a list
func real mean(list real xs):
  real sum;
  for x in xs:
    sum = sum + x;
  end;

  return sum / size(xs)
end

# Calculating the standard deviation of the elements inside a list.
func real StdDev(list real xs):
  real avgx;
  real sum;
  avgx = mean(xs);
  for x in xs:
    sum = sum + Square(x - avgx);
  end

  return sqrt(1/(size(xs)-1)*sum)
end
```

Appendix C

Proofs (Trajectory Optimization)

C.1 Analytical Solution Optimization Problem 1

In this section we find an analytic solution for the optimization problem posed in Section 4.8. We assume that we can find a solution to this optimization problem. Below we show the objective function and the inequality constraints of the optimization problem:

Objective:

$$J(y_1, y_2) = \frac{k(y_2+1)^2 + (y_1+1)^2}{1+y_1+y_2}. \tag{C.1a}$$

Constraints:

$$b_1 : y_1 \geq \frac{\rho_1}{1-\rho_1}(1+y_2), \tag{C.1b}$$

$$b_2 : y_2 \geq \frac{\rho_2}{1-\rho_2}(1+y_1), \tag{C.1c}$$

$$b_3 : y_1^{min} \leq y_1, \tag{C.1d}$$

$$b_4 : y_2^{min} \leq y_2, \tag{C.1e}$$

$$b_5 : y_1 \leq y_1^{max}, \tag{C.1f}$$

$$b_6 : y_2 \leq y_2^{max}, \tag{C.1g}$$

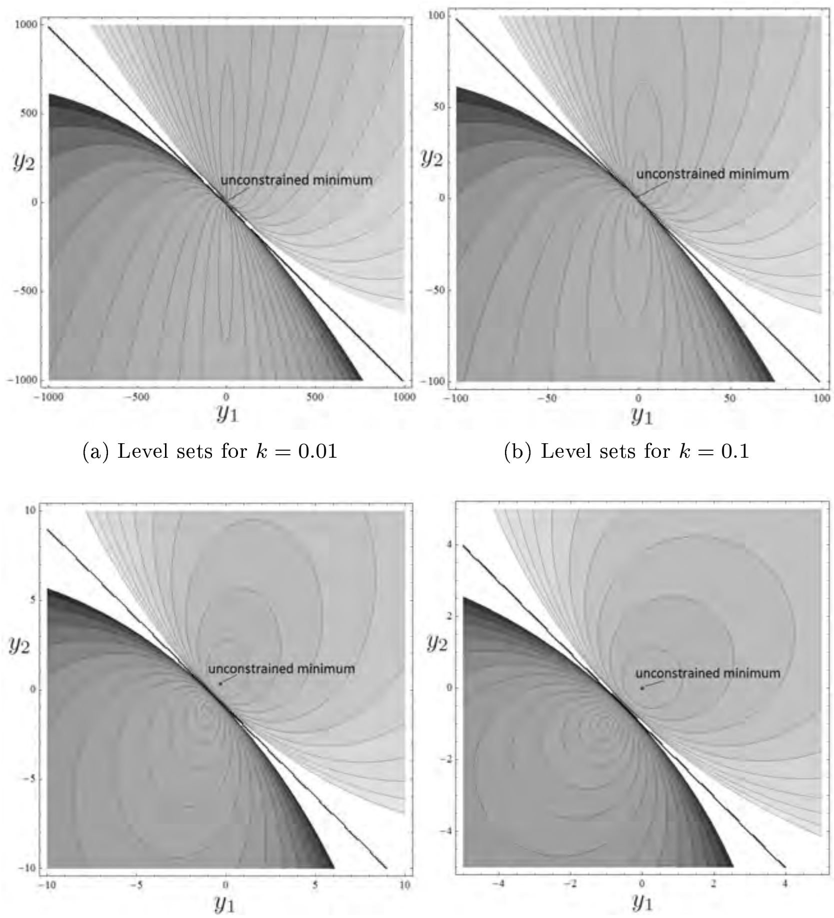
where

$$\begin{aligned}
&0 < k \leq 1, \\
&y_i^{min}, \geq 0 && i = 1, 2, \\
&0 < \rho_i < 1, && i = 1, 2, \\
&\rho_1 + \rho_2 < 1, \\
&\sigma_{1,2,1} > 0.
\end{aligned}$$

Unconstrained optimization problem First we consider the unconstrained problem. By taking the derivative of (C.1a) with respect to y_1 and y_2 and setting them to zero we obtain two stationary points:

$$(y_1, y_2) = (-1, -1),$$
$$(y_1, y_2) = \left(\frac{k-1}{k+1}, \frac{1-k}{k+1}\right).$$

$(y_1, y_2) = (-1, -1)$ is a local maximum and $(y_1, y_2) = \left(\frac{k-1}{k+1}, \frac{1-k}{k+1}\right)$ is a local minimum. In Figure C.1 we have shown some level sets for different values of k. As can be seen in these plots, the level sets are ellipsis. The shape of these ellipses depends on the value of k. Note that the scales on the axes of these figures differ.



(a) Level sets for $k = 0.01$

(b) Level sets for $k = 0.1$

(c) Level sets for $k = 0.5$

(d) Level sets for $k = 1$

Figure C.1: Level sets for different values of k.

Both stationary points (the local maximum as the local minimum) cannot be positioned in the feasible area of the constrained optimization problem. As a result, one of the constraints b_1 until b_6 must be active, i.e. the constrained minimizer is positioned on a boundary of the feasible area.

Feasible area In Figure C.2 we can see all constraints of the optimization problem. In this figure all boundaries of the constraints are positioned in the feasible area.

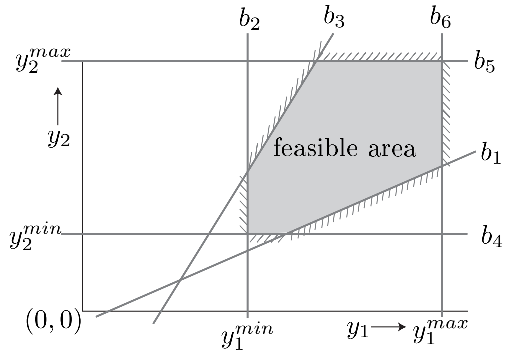


Figure C.2: Feasible area of the optimization problem.

Upper bounds We can derive $\frac{\partial J}{\partial y_1} + \frac{\partial J}{\partial y_2} = \frac{2(y_2+ky_1+(k+1)y_1y_2)}{(1+y_1+y_2)^2}$ which is greater than zero for $y_1 > 0 \wedge y_2 > 0$. Hence, on the boundary of constraint b_5 and b_6 and in the feasible area we can always decrease the linear cost function J by moving towards the $y_1 - axis$ or $y_2 - axis$. As a consequence, the global constrained minimum can only be positioned on a boundary of constraint b_1, b_2, b_3 or b_4.

Unconstrained minimum on a line

Lemma C.1 *On the line $y_2 = ay_1 + b_{y_1}$ were $a, b \in \mathbb{R}^+$, the linear cost function J has two stationary points: a local (unconstrained) minimum and a local (unconstrained) maximum. The y_1-position of the local minimum could be non-negative. The y_1-position of the local maximum could not. The position of the local (unconstrained) minimum on the line $y_2 = ay_1 + b_{y_1}$ is:*

$$y_1^{unc,min} = \frac{-(1+b)}{1+a} + \frac{\sqrt{(a-b)^2 + (1+b)^2 k}}{(1+a)\sqrt{(1+a^2k)}}.$$

Proof. In Figure C.7 we give an overview of what we are about to prove. In this figure we can see the following:

- A local (unconstrained) maximum at $y_1 = root_1 = \frac{-(1+b)}{1+a} - \frac{\sqrt{((a-b)^2+(1+b)^2k)}}{(1+a)\sqrt{(1+a^2k)}} < 0$.
- An asymptote at $\frac{-(1+b)}{1+a} < 0$.
- A local (unconstrained) minimum at $y_1 = root_2 = \frac{-(1+b)}{1+a} + \frac{\sqrt{((a-b)^2+(1+b)^2k)}}{(1+a)\sqrt{(1+a^2k)}}$.

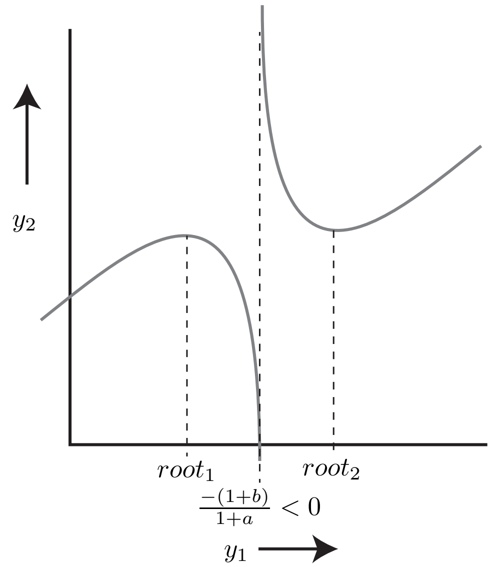


Figure C.3: Overview of this proof.

Using $y_2 = ay_1 + b$ and (C.1a) we obtain the following equation for J:

$$J = \frac{t_C + t_{y_1} y_1 + t_{y_1^2} y_1^2}{n_C + n_{y_1} y_1}, \tag{C.2}$$

where

$$\begin{aligned}
t_C &= 1 + (1+b)^2 k > 0, \\
t_{y_1} &= 2 + 2a(1+b)k > 0, \\
t_{y_1^2} &= 1 + a^2 k > 0, \\
n_C &= 1 + b > 0, \\
n_{y_1} &= 1 + a > 0.
\end{aligned}$$

Note that $|J|$ goes to infinity for $y_1 \to \frac{-(1+b)}{1+a} < 0$, i.e. there is an asymptote at $y_1 = \frac{-(1+b)}{1+a} < 0$.

When taking the derivative with respect to y_1 we obtain:

$$\frac{dJ}{dy_1} = \frac{n_C t_{y_1} - t_C n_{y_1} + 2 n_C t_{y_1^2} y_1 + n_{y_1} t_{y_1^2} y_1^2}{(n_C + n_{y_1} y_1)^2}. \tag{C.3}$$

We can find the y_1-position of the stationary points of (C.2) by setting the denominator of (C.3) equal to zero. By using the abc-equation and after some rewriting we can find expressions for the 2 stationary points: $root_1$ and $root_2$.

$$root_1 = \frac{-(1+b)}{1+a} - \frac{\sqrt{(a-b)^2+(1+b)^2k}}{(1+a)\sqrt{(1+a^2k)}} < 0,$$
$$root_2 = \frac{-(1+b)}{1+a} + \frac{\sqrt{(a-b)^2+(1+b)^2k}}{(1+a)\sqrt{(1+a^2k)}}.$$

Note that these roots are real (not imaginary) since $(a-b)^2+(1+b)^2k > 0$. We can see that $root_1$ is smaller than the asymptote. The other stationary point $root_2$ is larger than the asymptote.

We can easily see that $root_1$ is a maximum and $root_2$ is a minimum because the objective function in (C.2) goes to ∞ for $y_1 \to \infty$ and the objective function in (C.2) goes to $-\infty$ for $y_1 \to -\infty$.

Hence, the root $y_1^{unc,min}$ that corresponds to the unconstrained minimum on the line $y_2 = ay_1 + b$ is equal to:

$$y_1^{unc,min} = \frac{-(1+b)}{1+a} + \frac{\sqrt{(a-b)^2+(1+b)^2k}}{(1+a)\sqrt{1+a^2k}}. \tag{C.4}$$

■

Similarly to the proof of Lemma C.1 we can find that on the line $y_1 = ay_2 + b$, where $a, b \in \mathbb{R}^+$, the linear cost function J has two stationary points: a local (unconstrained) minimum and a local (unconstrained) maximum. The y_2-position of the local minimum could be non-negative. The y_1-position of the local maximum could not. The position of the local (unconstrained) minimum on the line $y_1 = ay_2 + b$ is:

$$y_2^{unc,min} = -\frac{(1+b)}{1+a} + \frac{\sqrt{(1+b)^2+(a-b)^2k}}{(1+a)\sqrt{a^2+k}}. \tag{C.5}$$

Using (C.15) and (C.5) we can find the unconstrained minima on the boundaries of the constraints b_1 until b_4. By writing the boundaries of these constraint in the form $y_1 = ay_2 + b$, $a, b \in \mathcal{R}^+$ or $y_2 = ay_1 + b$, $a, b \in \mathcal{R}^+$ we have obtained the following expressions for $\underline{y}^{b_i}(k) = \left(\underline{y}_1^{b_i}(k), \underline{y}_2^{b_i}(k)\right)$, $i = 1, ..., 4$, which is the $(y_1, y_2)-$position of the minimum on the boundary of constraint b_i:

$$\begin{aligned}
\underline{y}^{b_1}(k) &= \left(\frac{\rho_1}{\sqrt{k(1-\rho_1)^2+\rho_1^2}}, -1 + \frac{1-\rho_1}{\sqrt{k(1-\rho_1)^2+\rho_1^2}}\right), \\
\underline{y}^{b_2}(k) &= \left(-1 + \frac{\sqrt{k}(1-\rho_2)}{\sqrt{(1-\rho_2)^2+k\rho_2^2}}, \frac{\sqrt{k}\rho_2}{\sqrt{(1-\rho_2)^2+k\rho_2^2}}\right), \\
\underline{y}^{b_3}(k) &= \left(y_1^{min}, -(1+y_1^{min}) + \frac{\sqrt{(y_1^{min})^2k+(1+y_1^{min})^2}}{\sqrt{k}}\right), \\
\underline{y}^{b_4}(k) &= \left(-(1+y_2^{min}) + \sqrt{(y_2^{min})^2+(1+y_2^{min})^2k}, y_2^{min}\right).
\end{aligned} \tag{C.6}$$

We can see that $\underline{y}_1^{b_2}(k) = -1 + \frac{\sqrt{k}(1-\rho_2)}{\sqrt{(1-\rho_2)^2+k\rho_2^2}} < 0$. Thus, the unconstrained minimum on the boundary of constraint b_2 cannot be positioned in the feasible area.

As a result, the constrained global minimizer of the optimization problem with objective (C.1a) and constraints (C.1b) until (C.1g) is positioned on the boundary of constraint b_1, b_3 or b_4.

Monotonicity For now we are going to consider the optimization problem with only constraints b_1,b_3 and b_4, i.e. for the moment we forget about constraints b_2, b_5 and b_6 (see Figure C.4). Note that the three lines can intersect in different ways.

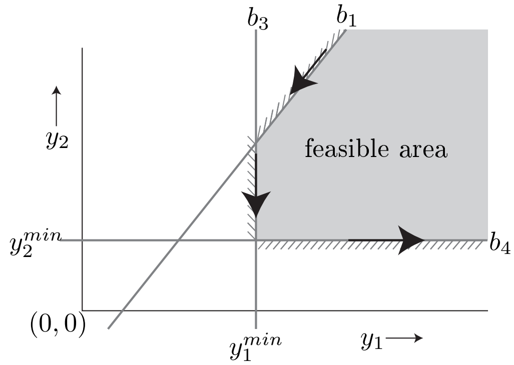


Figure C.4: Whenever we consider only constraints b_1,b_3 and b_4 and we increase k, the position of the constraint minimizer follows the arrows annotated to the boundaries of the 3 constraints.

Lemma C.2 *When we consider the constrained problem with the objective shown in (C.1a) and constraints b_1, b_3 and b_4, the position of the constrained minimum follows the arrows annotated to the boundaries in Figure C.4 when increasing k.*

Proof. We can easily see that the derivatives $\frac{\partial \underline{y}_1^{b_1}}{\partial k}$, $\frac{\partial \underline{y}_2^{b_3}}{\partial k}$ and $\frac{\partial \underline{y}_1^{b_4}}{\partial k}$ are according to the arrows in Figure C.4:We can easily see that the derivatives $\frac{\partial \underline{y}_1^{b_1}}{\partial k}$, $\frac{\partial \underline{y}_2^{b_3}}{\partial k}$ and $\frac{\partial \underline{y}_1^{b_4}}{\partial k}$ are according to the arrows in Figure C.4:

$$
\begin{aligned}
\frac{\partial \underline{y}_1^{b_1}}{\partial k} &= -\frac{(1-\rho_1)^2 \rho_1}{2(k(1-\rho_1)^2 + \rho_1^2)^{1.5}} < 0, \\
\frac{\partial \underline{y}_2^{b_3}}{\partial k} &= -\frac{(1-y_1^{min})^2}{2k^{1.5}\sqrt{(1+y_1^{min})^2 + (y_1^{min})^2 k}} < 0, \\
\frac{\partial \underline{y}_1^{b_4}}{\partial k} &= \frac{(1-y_2^{min})^2}{2\sqrt{k(1+y_2^{min})^2 + (y_2^{min})^2}} > 0.
\end{aligned}
$$

From now on we use $k_{i,j}$, $i, j = 1, ..., 6$ for the value of k for which $\underline{y}^{b_i}$ is positioned at the intersection of the boundary of constraint b_i and the boundaries of constraint b_j.

We can derive:

$$
\begin{aligned}
k_{1,3} &= \frac{\rho_1^2(1 - y_1^{min^2})}{(1-\rho_1)^2 y_1^{min^2}}, \\
k_{1,4} &= \frac{1}{(1+y_2^{min})^2} - \frac{\rho_1^2}{(1-\rho_1)^2}, \\
k_{3,1} &= \frac{\rho_1^2(1+y_1^{min})^2}{(1-\rho_1^2) y_1^{min^2}}, \\
k_{3,4} &= \frac{(1+y_1^{min})^2}{(1+y_2^{min})(1+y_2^{min}+2y_1^{min})}, \\
k_{4,1} &= \frac{1}{(1-\rho_1)^2} - \frac{y_2^{min^2}}{(1+y_2^{min})^2}, \\
k_{4,3} &= \frac{(1+y_1^{min})(1+y_1^{min}+2y_2^{min})}{(1+y_2^{min})^2}.
\end{aligned}
$$

We have to show that:

- $k_{4,3} \geq k_{3,4}$ when the intersection between the boundary of b_4 and the boundary of b_3 is positioned in the feasible area.
- $k_{4,1} \geq k_{1,4}$ when the intersection between the boundary of b_4 and the boundary of b_1 is positioned in the feasible area.
- $k_{3,1} \geq k_{1,3}$ when the intersection between the boundary of b_3 and the boundary of b_1 is positioned in the feasible area.

We can see that always $k_{4,3} \geq k_{3,4}$ because $1 + y_1^{min} \leq 1 + y_1^{min} + 2y_2^{min}$ (hence the numerator of $k_{4,3}$ is larger than the nominator of $k_{3,4}$) and $1 + y_2^{min} \leq 1 + y_2^{min} + 2y_1^{min}$ (hence the denominator of $k_{4,3}$ is smaller than the denominator of $k_{3,4}$). It also holds that $k_{4,1} \geq k_{1,4}$ because we can derive $k_{4,1} - k_{1,4} = \frac{2(\rho_1+y_2^{min})(1+\rho_1 y_2^{min})}{(1-\rho_1)^2(1+y_2^{min})^2} > 0$. We can derive that the y_2-position of the intersection between the boundary of b_3 and the boundary is non-negative for $y_1^{min} \geq \frac{\rho_1}{1-\rho_1}$. When using $y_1^{min} \geq \frac{\rho_1}{1-\rho_1}$ we can find $k_{3,1} - k_{1,3} = \frac{(y_1^{min}-\rho_1)\rho_1^2(1+y_1^{min})}{(1-\rho_1)^2(1+\rho_1)y_1^{min^2}} \geq \frac{\rho_1^4(1+y_1^{min})}{(1-\rho_1)^2(1+\rho_1)y_1^{min^2}} \geq 0$. Hence, it holds that $k_{3,1} \geq k_{1,3}$ when the intersection between the boundary of b_3 and the boundary of b_1 is positioned in the feasible area.

Thus, when we consider the constrained problem with the objective shown in (C.1a) and constraints b_1, b_3 and b_4, the position of the constrained minimum follows the arrows annotated to the boundaries when increasing k. ■

Lets again consider the optimization problem with constraints b_1 until b_6. We know that the constrained minimum is positioned on the boundary of b_1, b_3 or b_4. We can obtain expressions for the constraint minimum on each of these lines. We use k_i^{min} for the smallest value for k such that the unconstrained minimum $\underline{y}^{b_i}(k)$ is positioned in the feasible area. We use k_i^{max} for the largest value for k such that the unconstrained minimum $\underline{y}^{b_i}(k)$ is positioned in the feasible area. Whenever the boundary of constraint b_i is not positioned in the feasible area (and hence there is no constrained minimum on this boundary) it holds that $k_i^{min} > k_i^{max}$. Before we give the expressions for k_i^{min}, $i = 1, 3, 4$ and k_i^{max}, we give the relevant expressions for $k_{i,j}$:

$$
\begin{aligned}
k_{1,2} &= \frac{(1-\rho_2)(1-\rho_2-2\rho_1)}{(1-\rho_1)^2}, \\
k_{1,3} &= \frac{\rho_1^2(1-y_1^{min^2})}{(1-\rho_1)^2 y_1^{min^2}}, \\
k_{1,4} &= \frac{1}{(1+y_2^{min})^2} - \frac{\rho_1^2}{(1-\rho_1)^2}, \\
k_{1,5} &= \frac{1}{(1+y_2^{max})^2} - \frac{\rho_1^2}{(1-\rho_1)^2}, \\
k_{1,6} &= \frac{\rho_1^2(1-y_1^{max^2})}{(1-\rho_1)^2 y_1^{max^2}}, \\
k_{3,1} &= \frac{\rho_1^2(1+y_1^{min})^2}{(1-\rho_1^2) y_1^{min^2}}, \\
k_{3,2} &= \frac{(1-\rho_2)^2(1+y_1^{min})^2}{(1+y_1^{min})^2-(1-\rho_2)^2 y_1^{min^2}}, \\
k_{3,4} &= \frac{(1+y_1^{min})^2}{(1+y_2^{min})(1+y_2^{min}+2y_1^{min})}, \\
k_{3,5} &= \frac{(1+y_1^{min})^2}{(1+y_2^{max})(1+y_2^{max}+2y_1^{min})}, \\
k_{4,1} &= \frac{1}{(1-\rho_1)^2} - \frac{y_2^{min^2}}{(1+y_2^{min})^2}, \\
k_{4,2} &= \frac{(1-\rho_2^2) y_2^{min^2}}{\rho_2^2(1+y_2^{min})^2}, \\
k_{4,3} &= \frac{(1+y_1^{min})(1+y_1^{min}+2y_2^{min})}{(1+y_2^{min})^2}, \\
k_{4,6} &= \frac{(1+y_1^{max})(1+y_1^{max}+2y_2^{min})}{(1+y_2^{min})^2}.
\end{aligned}
$$

We found the following expressions for the constrained minima on the boundaries of constraints b_1, b_3 and b_4.

Constrained minimum on the boundary of b_1:

$$
\begin{aligned}
\underline{y}^{b_1}(k_1^{min}) &\text{ if } k \le k_1^{min} \le k_1^{max}, \\
\underline{y}^{b_1}(k) &\text{ if } k_1^{min} \le k \le k_1^{max}, \\
\underline{y}^{b_1}(k_1^{max}) &\text{ if } k_1^{min} \le k_1^{max} \le k,
\end{aligned}
$$

where

$$
\begin{aligned}
k_1^{min} &= \max\{k_{1,5}, k_{1,6}\}, \\
k_1^{max} &= \min\{k_{1,2}, k_{1,3}, k_{1,4}\}.
\end{aligned}
$$

Constrained minimum on the boundary of b_3:

$$\begin{aligned}\underline{y}^{b_3}(k_3^{min}) &\text{ if } k \le k_3^{min} \le k_3^{max},\\ \underline{y}^{b_3}(k) &\text{ if } k_3^{min} \le k \le k_3^{max},\\ \underline{y}^{b_3}(k_3^{max}) &\text{ if } k_3^{min} \le k_3^{max} \le k,\end{aligned}$$

where

$$\begin{aligned}k_3^{min} &= \max\{k_{3,1}, k_{3,5}\},\\ k_3^{max} &= \min\{k_{3,2}, k_{3,4}\}.\end{aligned}$$

Constrained minimum on the boundary of b_4:

$$\begin{aligned}\underline{y}^{b_4}(k_4^{min}) &\text{ if } k \le k_4^{min} \le k_4^{max},\\ \underline{y}^{b_4}(k) &\text{ if } k_4^{min} \le k \le k_4^{max},\\ \underline{y}^{b_4}(k_4^{max}) &\text{ if } k_4^{min} \le k_4^{max} \le k,\end{aligned}$$

where

$$\begin{aligned}k_4^{min} &= \max\{k_{4,1}, k_{4,3}\},\\ k_4^{max} &= \min\{k_{4,2}, k_{4,6}\}.\end{aligned}$$

Solution Using Lemma C.2 we can find the analytical solution (shown below) for the position of the constrained minimizer $\left(y_1^{con,min}, y_2^{con,min}\right)$.

$$\left(y_1^{con,min}, y_2^{con,min}\right) = \begin{cases} \underline{y}^{b_1}(k_1^{min}) & \text{if } k \le k_1^{min} \le k_1^{max},\\ \underline{y}^{b_1}(k) & \text{else if } k_1^{min} < k < k_1^{max},\\ \underline{y}^{b_1}(k_1^{max}) & \text{else if } k_3^{max} < k_3^{min} \vee k_4^{max} < k_4^{min},\\ \underline{y}^{b_1}(k_3^{min}) & \text{else if } k \le k_3^{min} \le k_3^{max},\\ \underline{y}^{b_1}(k) & \text{else if } k_3^{min} < k < k_3^{max},\\ \underline{y}^{b_1}(k_3^{max}) & \text{else if } k_4^{max} < k_4^{min},\\ \underline{y}^{b_1}(k_4^{min}) & \text{else if } k \le k_4^{min} \le k_4^{max},\\ \underline{y}^{b_1}(k) & \text{else if } k_4^{min} < k < k_4^{max},\\ \underline{y}^{b_1}(k_4^{max}) & \text{otherwise.}\end{cases}$$

where

$$\begin{aligned}\underline{y}^{b_1}(k) &= \left(\frac{\rho_1}{\sqrt{k(1-\rho_1)^2+\rho_1^2}}, -1 + \frac{1-\rho_1}{\sqrt{k(1-\rho_1)^2+\rho_1^2}}\right),\\ \underline{y}^{b_3}(k) &= \left(y_1^{min}, -(1+y_1^{min}) + \frac{\sqrt{(y_1^{min})^2 k + (1+y_1^{min})^2}}{\sqrt{k}}\right),\\ \underline{y}^{b_4}(k) &= \left(-(1+y_2^{min}) + \sqrt{(y_2^{min})^2 + (1+y_2^{min})^2 k}, y_2^{min}\right),\\ k_1^{min} &= \max\{k_{1,5}, k_{1,6}\},\\ k_1^{max} &= \min\{k_{1,2}, k_{1,3}, k_{1,4}\},\end{aligned}$$

$$
\begin{aligned}
k_3^{min} &= \max\{k_{3,1}, k_{3,5}\}, \\
k_3^{max} &= \min\{k_{3,2}, k_{3,4}\}, \\
k_4^{min} &= \max\{k_{4,1}, k_{4,3}\}, \\
k_4^{max} &= \min\{k_{4,2}, k_{4,6}\}, \\
k_{1,2} &= \frac{(1-\rho_2)(1-\rho_2-2\rho_1)}{(1-\rho_1)^2}, \\
k_{1,3} &= \frac{\rho_1^2(1-y_1^{min^2})}{(1-\rho_1)^2 y_1^{min^2}}, \\
k_{1,4} &= \frac{1}{(1+y_2^{min})^2} - \frac{\rho_1^2}{(1-\rho_1)^2}, \\
k_{1,5} &= \frac{1}{(1+y_2^{max})^2} - \frac{\rho_1^2}{(1-\rho_1)^2}, \\
k_{1,6} &= \frac{\rho_1^2(1-y_1^{max^2})}{(1-\rho_1)^2 y_1^{max^2}}, \\
k_{3,1} &= \frac{\rho_1^2(1+y_1^{min})^2}{(1-\rho_1^2) y_1^{min^2}}, \\
k_{3,2} &= \frac{(1-\rho_2)^2(1+y_1^{min})^2}{(1+y_1^{min})^2-(1-\rho_2)^2 y_1^{min^2}}, \\
k_{3,4} &= \frac{(1+y_1^{min})^2}{(1+y_2^{min})(1+y_2^{min}+2y_1^{min})}, \\
k_{3,5} &= \frac{(1+y_1^{min})^2}{(1+y_2^{max})(1+y_2^{max}+2y_1^{min})}, \\
k_{4,1} &= \frac{1}{(1-\rho_1)^2} - \frac{y_2^{min^2}}{(1+y_2^{min})^2}, \\
k_{4,2} &= \frac{(1-\rho_2^2) y_2^{min^2}}{\rho_2^2(1+y_2^{min})^2}, \\
k_{4,3} &= \frac{(1+y_1^{min})(1+y_1^{min}+2y_2^{min})}{(1+y_2^{min})^2}, \\
k_{4,6} &= \frac{(1+y_1^{max})(1+y_1^{max}+2y_2^{min})}{(1+y_2^{min})^2}.
\end{aligned}
$$

C.2 Analytical Solution Optimization Problem 2

In this section we find an analytic solution for the optimization problem posed in Section 7.5.3. Before you read this section we advice you to read Section C.1 first. In Section C.1 we find the analytical solution of a more simple optimization. However, finding the analytical solution is very similar for both optimization problems. Below we show the objective function and the inequality constraints of the optimization problem. We assume that we can find a solution to this optimization problem.

$$
\begin{aligned}
J(y_1, y_2) &= \frac{(1+y_2)^2 + k_1(1+y_1)^2 + k_2(k_3+y_1)^2}{1+y_1+y_2}, && \text{(C.7)}\\
b_1 &: y_2 \geq \frac{\alpha_2}{1-\alpha_2}(1+y_1),\\
b_2 &: y_2 \geq \frac{\alpha_3 y_1 + k_3}{1-\alpha_3} - 1,\\
b_3 &: y_1 \geq \frac{\alpha_1}{1-\alpha_1}(1+y_2),\\
b_4 &: y_1^{min} \leq y_1,\\
b_5 &: y_2^{min} \leq y_2,\\
b_6 &: y_1 \leq y_1^{max},\\
b_7 &: y_2 \leq y_2^{max},
\end{aligned}
$$

where

$$
\begin{aligned}
&0 < k_1 + k_2 \leq 1,\\
&0 < k_3, k_1, k_2 \leq 1,\\
&y_i^{min} \geq 0, && i = 1, 2,\\
&\alpha_i > 0, && i = 1, 2, 3,\\
&\alpha_1 + \alpha_2 < 1,\\
&\alpha_1 + \alpha_3 < 1.
\end{aligned}
\tag{C.8}
$$

Unconstrained optimization problem First we consider the unconstrained problem. By taking the derivative of (C.7) with respect to y_1 and y_2 and setting them to zero we obtain two stationary points:

$$
(y_1, y_2) = \left(-1 - \frac{\sqrt{(k_1+k_2)(k_1(1+k_2(1-k_3)^2) + k_2k_3^2)}}{\sqrt{k_1+k_2}(k_1+k_2k_3)}, \frac{-1-k_2(1-k_3) - \sqrt{(k_1+k_2)(k_1(1+k_2(1-k_3)^2)+k_2k_3^2)}}{1+k_1+k_2}\right), \tag{C.9a}
$$

$$
(y_1, y_2) = \left(-1 + \frac{\sqrt{(k_1+k_2)(k_1(1+k_2(1-k_3)^2) + k_2k_3^2)}}{\sqrt{k_1+k_2}(k_1+k_2k_3)}, \frac{-1-k_2(1-k_3) + \sqrt{(k_1+k_2)(k_1(1+k_2(1-k_3)^2)+k_2k_3^2)}}{1+k_1+k_2}\right). \tag{C.9b}
$$

The coordinate in (C.9a) is a local maximum. It is positioned outside the feasible area since both its y_1-coordinate and its y_2-coordinate are negative. The coordinate in (C.9b) is a local minimum. This point is positioned outside the feasible area because for this coordinate it holds that $y_1 + y_2 \leq 0$. We proof this below:

$$
y_1 + y_2 = \frac{-\sqrt{k_1+k_2} + \sqrt{k_1(1+k_2(1-k_3)^2) + k_2k_3^2}}{\sqrt{k_1+k_2}}.
$$

Thus, $y_1 + y_2 \leq 0$ if $k_1 + k_2 \geq k_1(1 + k_2(1 - k_3)^2) + k_2 k_3^2$. Below we prove that this is indeed the case:

$$
\begin{aligned}
k_1 + k_2 &\geq k_1(1 + k_2(1 - k_3)^2) + k_2 k_3^2, \\
k_2 &\geq k_1 k_2 (1 - k_3)^2 + k_2 k_3^2, \\
1 &\geq k_1(1 - k_3)^2 + k_3^2.
\end{aligned}
$$

Thus, both stationary points (the local maximum as the local minimum) cannot be positioned in the feasible area of the constrained optimization problem. As a result, one of the constraints b_1 until b_7 must be active, i.e. the constrained minimizer is positioned on a boundary of the feasible area.

Feasible area In Figure C.5 we can see all constraints of the optimization problem. In this figure all boundaries of the constraints are positioned in the feasible area. However, this does not have to be the case.

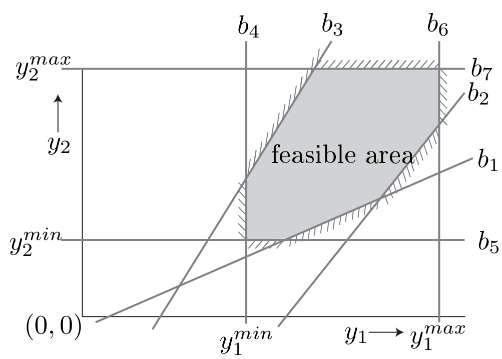


Figure C.5: Feasible area of the optimization problem.

Writing the boundaries of the constraints b_i, $i = 1, 2, 3$ in the form $y_2 = a^i y_1 + b^i$ gives us:

$$
\begin{aligned}
a^1 &= \frac{\alpha_2}{1 - \alpha_2} > 0, \\
a^2 &= \frac{\alpha_3}{1 - \alpha_3} > 0, \\
a^3 &= \frac{1 - \alpha_1}{\alpha_1} > 0.
\end{aligned}
\tag{C.10}
$$

Lemma C.3 *The boundary of constraint b_2, i.e. the line $y_2 = \frac{\alpha_3 y_1 + k_3}{1 - \alpha_3} - 1$, could be positioned in the feasible area only if $\alpha_3 \geq \alpha_2$. In this case it holds that $a^3 \geq a^2$.*

Proof. When we write the constraints b_i, $i = 1, 2$ in the form $y_2 = a^i y_1 + b^i$ we get:

$$a^1 = \frac{\alpha_2}{1-\alpha_2},$$
$$a^2 = \frac{\alpha_3}{1-\alpha_3},$$
$$b^1 = \frac{\alpha_2}{1-\alpha_2},$$
$$b^2 = \frac{k_3}{1-\alpha_3} - 1.$$

When $\alpha_3 < \alpha_2$ it holds that $b^1 - b^2 = \frac{1}{1-\alpha_2} - \frac{k_3}{1-\alpha_3} < 0$. In this case it holds that $a^2 > a^1$ and the boundary of constraint b_2 cannot be positioned in the feasible area. This situation is shown in Figure C.6.

Hence, boundary of constraint b_2, i.e. the line $y_2 = \frac{\alpha_3 y_1 + k_3}{1-\alpha_3} - 1$, could be positioned in the feasible area only if $\alpha_3 \geq \alpha_2$. In this case it holds that $a^3 \geq a^2$.

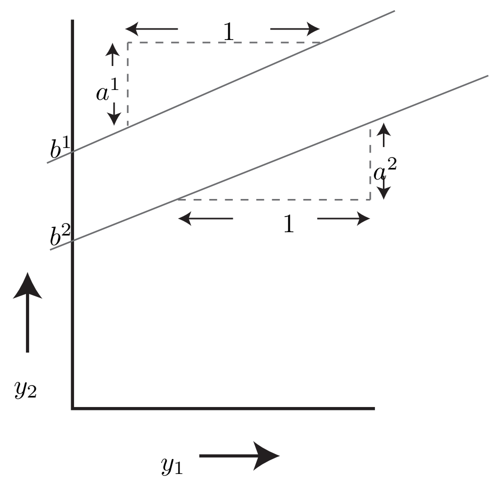


Figure C.6: The boundary of constraint b_2 could only be positioned in the feasible area whenever $\alpha_2 > \alpha_1$.

■

Upper bounds We can derive:

$$\frac{\partial J}{\partial y_1} + \frac{\partial J}{\partial y_2} = \frac{2y_1 + k_2(k_3+y_1)(y_2+(1-k_3)) + (k_1+(k_1+1)y_1)y_2}{(1+y_1+y_2)^2}.$$

We can see that $\frac{\partial J}{\partial y_1} + \frac{\partial J}{\partial y_2} > 0$ if $y_1 > 0 \wedge y_2 \geq 0$. Hence, on the boundary of constraint b_6 and b_7 and in the feasible area we can always decrease the linear cost function J by moving towards the $y_1 - axis$ or $y_2 - axis$. As a consequence, the global constrained minimum can only be positioned on a boundary of constraint b_1, b_2, b_3, b_4 or b_5.

Unconstrained minimum on a line

Lemma C.4 *On the line $y_2 = ay_1 + b_{y_1}$ were $a, b \in \mathbb{R}^+$, the linear cost function J has two stationary points: a local (unconstrained) minimum and a local (unconstrained) maximum. The y_1-position of the local minimum could be non-negative. The y_1-position of the local maximum could not. The position of the local (unconstrained) minimum on the line $y_2 = ay_1 + b_{y_1}$ is:*

$$y_1^{unc,min} = -\frac{(1+b)}{1+a} + \frac{\sqrt{(1+b)^2 + k_1(a-b)^2 + k_2((1+a)k_3 - (1+b))^2}}{(1+a)\sqrt{a^2+k_1+k_2}}. \tag{C.11}$$

Proof. In Figure C.7 we give an overview of what we are about to prove. In this figure we can see the following:

- A local (unconstrained) maximum at $y_1 = root_1 = -\frac{(1+b)}{1+a} - \frac{\sqrt{(1+b)^2+k_1(a-b)^2+k_2((1+a)k_3-(1+b))^2}}{(1+a)\sqrt{a^2+k_1+k_2}} < 0$.
- An asymptote at $\frac{-(1+b)}{1+a} < 0$.
- A local (unconstrained) minimum at $y_1 = root_2 = -\frac{(1+b)}{1+a} + \frac{\sqrt{(1+b)^2+k_1(a-b)^2+k_2((1+a)k_3-(1+b))^2}}{(1+a)\sqrt{a^2+k_1+k_2}}$.

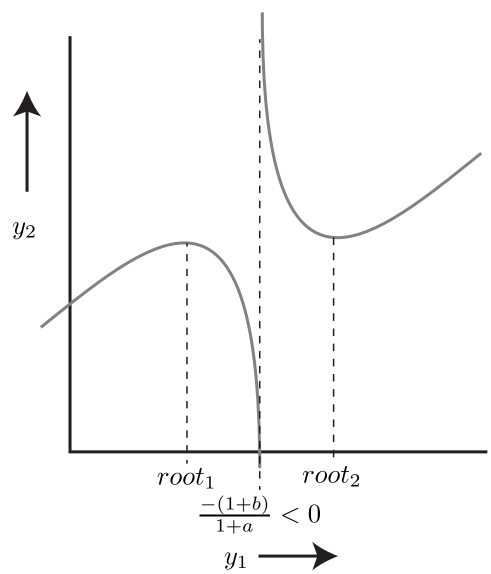


Figure C.7: Overview of this proof.

Using $y_2 = ay_1 + b$ and (C.1a) we obtain the following equation for J:

$$
\begin{aligned}
J &= \frac{t_C + t_{y_1} y_1 + t_{y_1^2} y_1^2}{n_C + n_{y_1} y_1}, &\text{(C.12)}\\
t_C &= (1+b)^2 + k_2 k_3^2,\\
t_{y_1} &= 2(a(1+b) + k_1 + k_2 k_3),\\
t_{y_1^2} &= a^2 + k_1 + k_2,\\
n_C &= 1+b,\\
n_{y_1} &= 1+a.
\end{aligned}
\tag{C.13}
$$

Note that $|J|$ goes to infinity for $y_1 \to \frac{-(1+b)}{1+a} < 0$, i.e. there is an asymptote at $y_1 = \frac{-(1+b)}{1+a} < 0$.

When taking the derivative with respect to y_1 we obtain:

$$
\frac{dJ}{dy_1} = \frac{n_C t_{y_1} - t_C n_{y_1} + 2 n_C t_{y_1^2} y_1 + n_{y_1} t_{y_1^2} y_1^2}{n_C + n_{y_1} y_1}. \tag{C.14}
$$

We can find the y_1-position of the stationary points of (C.12) by setting the denominator of (C.14) equal to zero. By using the abc-equation and after some rewriting we can find expressions for the two stationary points: $root_1$ and $root_2$.

$$
\begin{aligned}
root_1 &= -\frac{(1+b)}{1+a} - \frac{\sqrt{(1+b)^2 + k_1(a-b)^2 + k_2((1+a)k_3 - (1+b))^2}}{(1+a)\sqrt{a^2+k_1+k_2}} < 0,\\
root_2 &= -\frac{(1+b)}{1+a} + \frac{\sqrt{(1+b)^2 + k_1(a-b)^2 + k_2((1+a)k_3 - (1+b))^2}}{(1+a)\sqrt{a^2+k_1+k_2}}.
\end{aligned}
$$

Note that these roots are real (not imaginary) since $(1+b)^2 + k_1(a-b)^2 + k_2((1+a)k_3 - (1+b))^2 > 0$. We can see that $root_1$ is smaller than the asymptote. The other stationary point $root_2$ is larger than the asymptote.

We can easily see that $root_1$ is a maximum and $root_2$ is a minimum because the objective function in (C.12) goes to ∞ for $y_1 \to \infty$ and the objective function in (C.12) goes to $-\infty$ for $y_1 \to -\infty$.

Hence, the root $y_1^{unc,min}$ that corresponds to the unconstrained minimum on the line $y_2 = a y_1 + b$ is equal to:

$$
-\frac{(1+b)}{1+a} + \frac{\sqrt{(1+b)^2 + k_1(a-b)^2 + k_2((1+a)k_3 - (1+b))^2}}{(1+a)\sqrt{a^2+k_1+k_2}}. \tag{C.15}
$$

■

Similarly to the proof of Lemma C.1 we can find that on the line $y_1 = a y_2 + b$, where $a, b \in \mathbb{R}^+$, the linear cost function J has two stationary points: a local (unconstrained) minimum and a local (unconstrained) maximum. The y_2-position of the local minimum could be non-negative. The y_1-position of the local maximum could not. The position of the local (unconstrained) minimum on the line $y_1 = a y_2 + b$ is:

$$
y_2^{\text{unc, min}} = -\frac{(1+b)}{1+a} + \frac{\sqrt{k_1(1+b)^2 + k_2((k_3-1)(1+a) + (1+b))^2 + (a-b)^2}}{(1+a)\sqrt{1 + a^2(k_1+k_2)}}. \tag{C.16}
$$

Using (C.11) and (C.16) we can find the unconstrained minima on the boundaries of the constraints b_1 until b_4. By writing the boundaries of these constraint in the form $y_1 = ay_2 + b$, $a, b \in \mathbb{R}^+$ or $y_2 = ay_1 + b$, $a, b \in \mathbb{R}^+$ we have obtained the following expressions for $\underline{y}^{b_i}(k_1) = \left(\underline{y}_1^{b_i}(k_1), \underline{y}_2^{b_i}(k_1)\right)$, $i = 1, ..., 5$, which is the $(y_1, y_2)-$position of the minimum on the boundary of constraint b_i:

$$\underline{y}^{b_1}(k_1) = \left(-1 + \frac{(1-\alpha_2)\sqrt{1+k_2(1-k_3)^2}}{\sqrt{(k_1+k_2)(1-\alpha_2)^2+\alpha_2^2}}, \frac{\alpha_2\sqrt{1+k_2(1-k_3)^2}}{\sqrt{(k_1+k_2)(1-\alpha_2)^2+\alpha_2^2}}\right), \tag{C.17a}$$

$$\underline{y}^{b_2}(k_1) = \left(-k_3 + \frac{(1-\alpha_3)\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}}, -(1-k_3) + \frac{\alpha_3\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}}\right), \tag{C.17b}$$

$$\underline{y}^{b_3}(k_1) = \left(\frac{\sqrt{k_1+k_2k_3^2}\alpha_1}{\sqrt{(k_1+k_2)\alpha_1^2+(1-\alpha_1)^2}}, -1 + \frac{\sqrt{k_1+k_2k_3^2}(1-\alpha_1)}{\sqrt{(k_1+k_2)\alpha_1^2+(1-\alpha_1)^2}}\right), \tag{C.17c}$$

$$\underline{y}^{b_4}(k_1) = \left(y_1^{min}, -(1+y_1^{min}) + \sqrt{(y_1^{min})^2 + k_1(1+y_1^{min})^2 + k_2(y_1^{min}+k_3)^2}\right), \tag{C.17d}$$

$$\underline{y}^{b_5}(k_1) = \left(-(1+y_2^{min}) + \frac{\sqrt{(1+y_2^{min})^2 + k_1(y_2^{min})^2 + k_2(1+y_2^{min}-k_3)^2}}{\sqrt{k_1+k_2}}, y_2^{min}\right). \tag{C.17e}$$

We can see that $\underline{y}_2^{b_3}(k_1) \leq 0$ because $\sqrt{k_1+k_2k_3^2}(1-\alpha_1) \leq (1-\alpha_1)$ and $\sqrt{(k_1+k_2)\alpha_1^2+(1-\alpha_1)^2} \geq (1-\alpha_1)$. Hence at least one of the boundaries b_i $i = 1, 2, 4, 5$ must be active.

Monotonicity For now we are going to consider the optimization problem with only constraints b_i,$i = 1, 2, 4, 5$, i.e. for the moment we forget about constraints b_3, b_6 and b_7 (see Figure C.8). Note that the three lines can intersect in different ways.

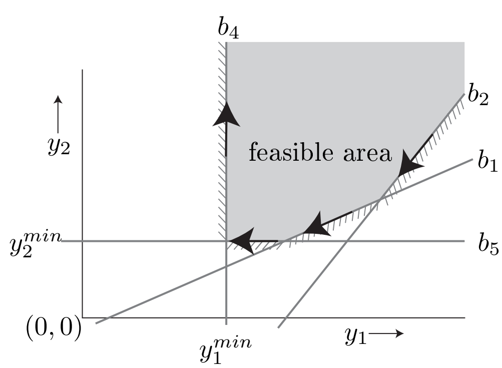


Figure C.8: Whenever we consider only constraints b_1,b_2,b_4 and b_5 and we increase k_1 (in the range $[0, 1]$), the position of the constraint minimizer follows the arrows annotated to the boundaries.

Theorem C.1 *When we consider the constrained problem with the objective shown in (C.7) and constraints b_i, $i = 1, 2, 4, 5$ the arrows annotated to the boundaries of constraints b_i ,$i = 1, 2, 4, 5$ shown in Figure C.8 visualize the direction in which the constrained minimum moves when increasing k_1 (in the range $[0, 1]$).*

Proof. Before we prove Theorem C.1 we first prove some lemma's.

Lemma C.5 *The constraint minimum* $\underline{y}^{b_i}(k_1)$, $i = 1, 4, 5$ *moves in the direction shown in Figure C.8 when increasing* k_1.

Proof. We can derive:

$$\begin{aligned}
\frac{\partial \underline{y}_1^{b_1}}{\partial k_1} &= -\frac{(1-\alpha_2)^2\sqrt{1+k_2(1-k_3)^2}}{2((k_1+k_2)(1-\alpha_2^2)+\alpha_2^2)^{1.5}} < 0,\\
\frac{\partial \underline{y}_2^{b_4}}{\partial k_1} &= \frac{(1+y_1^{min})^2}{2\sqrt{(y_1^{min})^2+k_1(1+y_1^{min})^2+k_2(k_3+y_1^{min})^2}} > 0,\\
\frac{\partial \underline{y}_1^{b_5}}{\partial k_1} &= -\frac{k_2(1-k_3)(1-k_3+2y_2^{min})+(1+y_2^{min})^2}{2(k_1+k_2)^{1.5}\sqrt{k_1 y_2^{min^2}+(1+y_2^{min})^2+k_2(1-k_3+y_2^{min})^2}} < 0.
\end{aligned}$$

Thus, the derivatives $\frac{\partial \underline{y}_1^{b_1}}{\partial k_1}$, $\frac{\partial \underline{y}_2^{b_4}}{\partial k_1}$ and $\frac{\partial \underline{y}_1^{b_5}}{\partial k_1}$ are accordance with the arrows in Figure C.8. ∎

Lemma C.6 *When* $\underline{y}_1^{b_2}(k_1) \geq 0$ *and* $\underline{y}_2^{b_2}(k_1) \geq 0$ *then it must hold that* $k_3 > \alpha_3$.

Proof. Lets assume that $\underline{y}_1^{b_2}(k_1) \geq 0 \wedge \underline{y}_2^{b_2}(k_1) \geq 0$ is also possible when $k_3 \leq \alpha_3$. First we derive that if $\underline{y}_1^{b_2}(k_1) \geq 0 \wedge \underline{y}_2^{b_2}(k_1) \geq 0$ was possible when $k_3 \leq \alpha_3$ then it must hold that $k_3 < 1-\alpha_3$. Hereafter we derive that when $k_3 \leq \alpha_3$ and $k_3 \leq 1-\alpha_3$ it holds that $\underline{y}_2^{b_2} < 0$. Hence, $\underline{y}_1^{b_2}(k_1) \geq 0 \wedge \underline{y}_2^{b_2}(k_1) \geq 0$ could not occur when $k_3 \leq \alpha_3$. In case $\underline{y}_1^{b_2}(k_1) \geq 0 \wedge \underline{y}_2^{b_2}(k_1) \geq 0$ it must hold that:

$$\underline{y}_1^{b_2} + \underline{y}_1^{b_2} = -1 + \frac{\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+(1-\alpha_1)^2}} \geq 1. \tag{C.18}$$

$$\tag{C.19}$$

This because otherwise either $\underline{y}_1^{b_2}$ is negative, $\underline{y}_2^{b_2}$ is negative or both are negative and the unconstrained minimum $\underline{y}^{b_2}(k_1)$ could not be positioned in the feasible area. Lets use:

$$q = \frac{\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_1^2}}. \tag{C.20}$$

We can derive that:

$$\frac{\partial q}{\partial k_2} = -\frac{(1-\alpha_3)^2\sqrt{k_1(1-k_3)^2+k_3^2}}{2(a_3^2+(1-a_3)^2(k_1+k_2))^{1.5}} < 0.$$

Hence, it holds that:

$$\frac{\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{k_1(1-\alpha_3)^2+\alpha_1^2}} < q.$$

For $k_2 = 0$ we can derive:

$$\frac{\partial q}{\partial k_1} = \frac{(1-k_3)^2\alpha_3^2 - (1-\alpha_3)^2 k_3^2}{2(k_1(1-\alpha_3)^2 + a_3^2)^{1.5}\sqrt{k_1(1-k_3)^2 + k_3^2}}.$$

We know $\frac{\partial q}{\partial k_1} \geq 0$ because $k_3 \leq \alpha_3$. Hence it holds that:

$$\frac{\sqrt{(1-k_3)^2 + k_3^2}}{\sqrt{(1-\alpha_3)^2 + \alpha_1^2}} \geq \frac{\sqrt{k_1(1-k_3)^2 + k_3^2}}{\sqrt{k_1(1-\alpha_3)^2 + \alpha_1^2}} > q.$$

Hence, when $q \geq 1$ it must hold that:

$$(1-k_3)^2 + k_3^2 > (1-\alpha_3)^2 + \alpha_1^2. \tag{C.21}$$

In Figure (C.9) we can see the function $f(x) = (1+x)^2 + x^2$ for $0 \leq x \leq 1$. We can see that this function is symmetric around the line $x = 0.5$. Both x', $0 \leq x' \leq 1$ and $1 - x'$ result in the same value $f(x')$. From this figure we can easily see that when $k_3 \leq \alpha_3$ and $\alpha_3 \geq 0.5$ then $f(k_3) > f(\alpha_3)$ could only hold if $k_3 < 1 - \alpha_3$. Hence, when $k_3 \leq \alpha_3$, (C.18) could only be satisfied when it also holds that $k_3 < 1 - \alpha_3$ (for the case where $\alpha_3 < 0.5$ it also holds that $k_3 < 1 - \alpha_3$ because $k_3 \leq \alpha_3$).

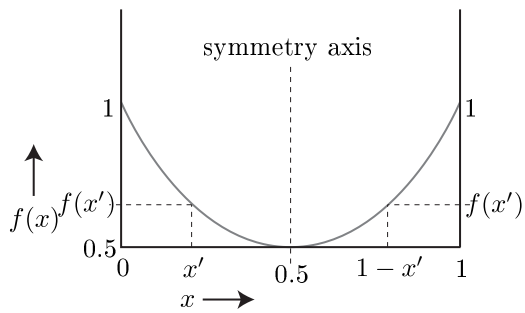


Figure C.9: The function $f(x) = (1-x)^2 + x^2$.

Thus, when $k_3 \leq \alpha_3$ and the unconstrained minimum $\underline{y}^{b_2}(k_1)$ is positioned in the feasible then it must hold that $k_3 < 1 - \alpha_3$. Lets distinguish the two situations below:

situation 1 $k_3 \leq \alpha_3 < 1 - \alpha_3$. In this situation it holds that $\alpha_3 \leq 0.5$.
situation 2 $k_3 < 1 - \alpha_3 \leq \alpha_3$. In this situation it holds that $\alpha_3 > 0.5$.

For both situations we can show that the y_2 coordinate of the unconstrained minimum $\underline{y}^{b_2}(k_1)$ is negative. As a result the unconstrained minimum $\underline{y}^{b_2}(k_1)$ could not be positioned in the feasible area when $k_3 \leq \alpha_3$.

Situation 1 From (C.17b) we can obtain that:

$$\frac{\partial \underline{y}_2^{b_2}}{\partial k_3} = 1 - \frac{\alpha_3}{\alpha_3^2 + (1-\alpha_3)^2 k_2} \frac{k_1(1-k_3) - k_3}{\sqrt{k_2(1-k_3)^2 + k_3^2}} > 0.$$

Hence, we can derive:

$$
\begin{aligned}
\underline{y}_2^{b_2} =& -(1-k_3) + \frac{\alpha_3\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}},\\
\underline{y}_2^{b_2} \leq& -(1-\alpha_3) + \frac{\alpha_3\sqrt{k_1(1-\alpha_3)^2+\alpha_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}},\\
\underline{y}_2^{b_2} \leq& -1+\alpha_3(1+\frac{\sqrt{k_1(1-\alpha_3)^2+\alpha_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}}),\\
\underline{y}_2^{b_2} <& -1+2\alpha_3,\\
\underline{y}_2^{b_2} <& 0.
\end{aligned}
$$

In the first step we used $k_3 = \alpha_3$. Hence, in situation 1 the unconstrained minimum $\underline{y}^{b_2}(k_1)$ could not be positioned in the feasible area.

Situation 2 In situation 2 we can derive:

$$
\begin{aligned}
\underline{y}_2^{b_2} =& -(1-k_3) + \frac{\alpha_3\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}},\\
\underline{y}_2^{b_2} \leq& -\alpha_3 + \frac{\alpha_3\sqrt{k_1\alpha_3^2+(1-\alpha_3)^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}},\\
\underline{y}_2^{b_2} \leq& \alpha_3(\frac{\sqrt{k_1\alpha_3^2+(1-\alpha_3)^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}}-1).
\end{aligned}
$$

Using $k_2(1-\alpha_3)^2 \geq 0$ and $1-\alpha_3 < \alpha_3$ we can derive that $\frac{\sqrt{k_1\alpha_3^2+(1-\alpha_3)^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}} < 1$. Hence, in situation 2 it holds that $\underline{y}_2^{b_2} < 0$ and the unconstrained minimum $\underline{y}^{b_2}(k_1)$ could not be positioned in the feasible area. ■

Lemma C.7 *When the unconstrained minimum $\underline{y}^{b_2}(k_1)$ is in the feasible area, it moves in the direction shown in Figure C.8 when increasing k_1 (in the range $[0,1]$).*

Proof. In this lemma we prove that $\frac{\partial \underline{y}_1^{b_2}}{\partial k_1} < 0$ whenever the unconstrained minimum $\underline{y}^{b_2}$ is positioned in the feasible area.

We do so by proving that $\frac{\partial \underline{y}_1^{b_2}}{\partial k_1} \geq 0$ could only occur if $\underline{y}_1^{b_2} + \underline{y}_2^{b_2} < 0$. If $\underline{y}_1^{b_2} + \underline{y}_2^{b_2} < 0$ then the y_1-coordinate or the y_2-coordinate of $\underline{y}^{b_2}$ is negative (and thus $\underline{y}^{b_2}$ is positioned outside the feasible area).

We can derive the following expression for $\frac{\partial \underline{y}_1^{b_2}}{\partial k_1}$:

$$
\frac{k_2(1-\alpha_3)^2(1-k_3)^2-(k_3-\alpha_3)(\alpha_3(1-k_3)+k_3(1-\alpha_3))}{2(1-\alpha_3)((k_1+k_3)(1-\alpha_3)^2+\alpha_3^2)^{1.5}\sqrt{k_3^2+k_1(1-k_3)^2}}.
$$

We can see that:

$$\frac{\partial \underline{y}_1^{b_2}}{\partial k_1} < 0 \quad \text{if} \quad k_2 < h,$$
$$\frac{\partial \underline{y}_1^{b_2}}{\partial k_1} = 0 \quad \text{if} \quad k_2 = h,$$
$$\frac{\partial \underline{y}_1^{b_2}}{\partial k_1} > 0 \quad \text{if} \quad k_2 > h,$$

where

$$h = \frac{(k_3 - \alpha_3)(\alpha_3(1-k_3) + k_3(1-\alpha_3))}{(1-k_3)^2(1-\alpha_3)^2}.$$

From Lemma C.6 we know that $\underline{y}_1^{b_2} > 0 \wedge \underline{y}_1^{b_2} > 0$ is only possible when $k_3 > \alpha_3$. Hence, it holds that $h > 0$. When $h > 1$ it holds that $\frac{\partial \underline{y}_1^{b_2}}{\partial k_1} < 0$ because $k_2 \leq 1$. We still have to proof that $\frac{\partial \underline{y}_1^{b_2}}{\partial k_1} < 0$ for $h < 0 \leq 1$. We do so by proving that $0 < h \leq 1$ and $k_2 \geq h$ is not possible.

We can derive the following:

$$\frac{\partial(\underline{y}_1^{b_2} + \underline{y}_2^{b_2})}{\partial k_2} = -\frac{\sqrt{k_1(1-k_3)^2 + k_3^2}(1-\alpha_3)^2}{((k_1+k_2)(1-\alpha_3)^2 + \alpha_3^2)^{1.5}} < 0.$$

Therefore, when $0 < h \leq 1$ and $k_2 \geq h$ it holds:

$$\begin{aligned}\underline{y}_1^{b_2} + \underline{y}_1^{b_2} &= -1 + \frac{\sqrt{k_1(1-k_3)^2 + k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2 + (1-\alpha_1)^2}} \\ &\leq -1 + \frac{1-k_3}{1-\alpha_3} < 0.\end{aligned}$$

We used $k_2 = h$ which is the smallest value for k_2 satisfying $0 < h \leq 1$ and $k_2 \geq h$. When $0 < h \leq 1$ and $k_2 \geq h$ this value for k_2 results is the largest value for $\underline{y}_1^{b_2} + \underline{y}_1^{b_2}$ since $\frac{\partial(\underline{y}_1^{b_2} + \underline{y}_2^{b_2})}{\partial k_2} < 0$. In the last step we used $k_3 > \alpha_3$.

Thus, when the unconstrained minimum $\underline{y}^{b_2}(k_1)$ is in the feasible area it holds that $\frac{\partial \underline{y}_1^{b_2}}{\partial k_1} < 0$. Thus, the unconstrained minimum $\underline{y}^{b_2}(k_1)$ it moves in the direction shown in Figure C.8 when increasing k_1 (in the range $[0,1]$). ■

From now on we use $y^{i=j} = \left(y_1^{i=j}, y_2^{i=j}\right)$ for the coordinate of the intersection of the boundary of constraint b_i and the boundaries of constraint b_j. Further we use $k_{i,j}$, $i,j = 1,...,7$ for the value of k_1 for which $\underline{y}^{b_i}$ is positioned at $y^{i=j}$.

Lemma C.8 *Whenever the coordinate $y^{1=2}$ is positioned in the feasible area and $0 \leq k_{2,1} \leq 1$, it holds that $k_{1,2} \geq k_{2,1}$.*

Proof. We can obtain $y^{1,2} = \left(\frac{(1-\alpha_3) - k_3(1-\alpha_2)}{\alpha_3 - \alpha_2}, \frac{(1-k_3)\alpha_2}{\alpha_3 - \alpha_2}\right)$. From $y_2^{2,3} \geq 0$ we can obtain that $\alpha_2 \leq \alpha_3$ and from $y_1^{2,3} \geq 0$ we can obtain that $k_3 \leq \frac{(1-\alpha_3)}{1-\alpha_2}$.

Furthermore, we know from Lemma C.6 that if $\underline{y}_1^{b_2}(k_1) \geq 0 \wedge \underline{y}_2^{b_2}(k_1) \geq 0$ then it must hold that $k_3 > \alpha_3$. Hence, we know that when $y^{1=2}$ is positioned in the feasible area then $0 \leq k_{2,1} \leq 1$ holds only when $k_3 > \alpha_3$.

Thus, we have to prove that $k_{1,2} - k_{2,1} \geq 0$ whenever $\alpha_2 \leq \alpha_3 \leq k_3 \leq \frac{(1-\alpha_3)}{1-\alpha_2}$. Note that this also means that $\alpha_3 \leq \frac{1}{2-\alpha_2}$. We can derive that:

$$
\begin{aligned}
k_{1,2} &= -\frac{k_2(1-\alpha_3)(1-k_3)^2((1-\alpha_2)+(\alpha_3-\alpha_2))+\alpha_2^2(1-k_3)^2-(\alpha_3-\alpha_2)^2}{(1-\alpha_2)^2(1-k_3)^2},\\
k_{2,1} &= -\frac{k_2(1-k_3)^2(1-\alpha_3)^2+(\alpha_3-k_3\alpha_2)((1-k_3)\alpha_3-k_3(\alpha_3-\alpha_2))}{(1-k_3)^2(1-\alpha_2)((1-\alpha_3)-(\alpha_3-\alpha_2))}.
\end{aligned}
$$

We can derive:

$$
\begin{aligned}
k_{1,2}-k_{2,1} = & \frac{2(\alpha_3-\alpha_2)^2(1-\alpha_3)}{(1-\alpha_2)^2((1-\alpha_3)-(\alpha_3-\alpha_2))}k_2,\\
&+\frac{2(\alpha_3-\alpha_2)(\alpha_3-k_3\alpha_2)((1-k_3)-(\alpha_3-\alpha_2))}{(1-k_3)^2(1-\alpha_2)^2((1-\alpha_3)-(\alpha_3-\alpha_2))} \geq 0.
\end{aligned}
$$

We can see that $k_1^{2,3} - k_1^{3,2} \geq 0$ by using $\alpha_3 \geq \alpha_2$ and because:

$$
(1-\alpha_3)-(\alpha_3-\alpha_2) \geq (1-k_3)-(\alpha_3-\alpha_2) \geq \frac{\alpha_2(\alpha_3-\alpha_2)}{1-\alpha_2} \geq 0.
$$

■

We shortly elaborate on the result of Lemma C.8. Lets consider the optimization problem with objective function (C.7) and we want to find the constrained minimum on the boundary of either b_1 or on the boundary of b_2 (see Figure C.10). Thus, we want to solve the following optimization problem:

$$
\begin{aligned}
& \text{Objective function:}\\
J(y_1,y_2) = & \frac{(1+y_2)^2+k_1(1+y_1)^2+k_2(k_3+y_1)^2}{1+y_1+y_2}.\\
& \text{Constraint:}\\
& y_2 \geq \frac{\alpha_2}{1-\alpha_2}(1+y_1),\\
& y_2 \geq \frac{\alpha_3 y_1+k_3}{1-\alpha_3}-1,\\
& y_2 = \frac{\alpha_2}{1-\alpha_2}(1+y_1) \vee y_2 = \frac{\alpha_3 y_1+k_3}{1-\alpha_3}-1.
\end{aligned}
$$

For $0 \leq k_1 \leq 1$ it holds that the constraint minimum is positioned on the line $y_2 = \frac{\alpha_3 y_1+k_3}{1-\alpha_3}-1$ (the boundary of constraint b_2) if $k_1 < k_{2,1}$ for $k_{2,1} \leq k_1 \leq k_{1,2}$ the constrained minimum is positioned on the intersection of the two lines and for $k_1 > k_{1,2}$ the constrained minimum is positioned on the line $y_2 = \frac{\alpha_2}{1-\alpha_2}(1+y_1)$ (the boundary of constraint b_1). Thus, when increasing k_1 the constrained minimum moves along the arrows (see Figure C.10).

Lemma C.9 *Whenever the coordinate $y^{1=5}$ is positioned in the feasible area, it holds that $k_{5,1} \geq k_{1,5}$.*

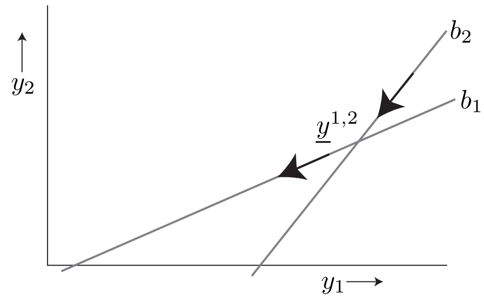


Figure C.10: The constrained minimum moves is on the boundary of b_2 for $k_1 < k_{2,1}$. For $k_{2,1} \leq k_1 \leq k_{1,2}$ the constrained minimum is positioned on the intersection of the two boundaries and for $k_1 > k_{1,2}$ the constrained minimum is positioned on the boundary of b_1.

Proof. We can obtain $y^{1,5} = \left(-1 + \frac{1-\alpha_2}{\alpha_2} y_2^{min}, y_2^{min}\right)$, which could only be positioned in the feasible area if $y_2^{min} \geq \frac{\alpha_2}{1-\alpha_2}$.

$$k_1^{5,1} = \frac{\alpha_2((1+y_2^{min})^2 + k_2(1-k_3+y_2^{min})^2) - k_2 y_2^{min^2}}{(1-\alpha_2^2) y_2^{min^2}},$$
$$k_1^{1,5} = \frac{(1+k_2(1-k_3)^2)\alpha_2^2 - (k_2(1-\alpha_2)^2 + \alpha_2^2) y_2^{min^2}}{(1-\alpha_2)^2 y_2^{min^2}}.$$

Using $y_2^{min} \geq \frac{\alpha_2}{1-\alpha_2}$ we can find:

$$\begin{aligned} k_1^{5,1} - k_1^{1,5} &= \frac{2\alpha_2^2(y_2^{min}(1+k_2(1-k_3)+(1-\alpha_2)y_2^{min}) - \alpha_2(1+k_2(1-k_3)(1-k_3+y_2^{min}))}{(1-\alpha_2)^2(1+\alpha_2)(y_2^{min})^2} \\ &\geq \frac{2\alpha_2^2(y_2^{min}(1+k_2(1-k_3)+(1-\alpha_2)y_2^{min}) - y_2^{min}(1-\alpha_2)(1+k_2(1-k_3)(1-k_3+y_2^{min})))}{(1-\alpha_2)^2(1+\alpha_2)(y_2^{min})^2} \\ &\geq \frac{2\alpha_2^2 y_2^{min}(y_2^{min}(1-\alpha_2)(1-k_2(1-k_3)) + \alpha_2(1+k_2(1-k_3)^2) + k_2(1-k_3)k_3)}{(1-\alpha_2)^2(1+\alpha_2)(y_2^{min})^2} \geq 0. \end{aligned}$$

■

Lemma C.10 *Whenever the coordinate $y^{1=4}$ is positioned in the feasible area, it holds that $k_{4,1} \geq k_{1,4}$.*

We can obtain:

$$\begin{aligned} k_1^{1,4} &= -\frac{\alpha_2^2}{(1-\alpha_2)^2} - \frac{1-k_2(2-k_3-y_1^{min})(k_3+y_1^{min})}{(1+y_1^{min})^2}, \\ k_1^{4,1} &= \frac{1-k_2(1-\alpha_2)^2(k_3+y_1^{min})^2 + y_1^{min}(2+(2-\alpha_2)\alpha_2 y_1^{min})}{(1-\alpha_2)^2(1+y_1^{min})^2}. \end{aligned} \quad \text{(C.22)}$$

we can obtain:

$$k_1^{4,1} - k_1^{1,4} = \frac{2(\alpha_2 + y_1^{min}(1+\alpha_2) + k_2(1-k_2)(1-\alpha_2)^2(k_3+y_1^{min}))(1+\alpha_2 y_1^{min})}{(1-\alpha_2)^2(1+y_1^{min})^2} \geq 0.$$

Lemma C.11 *Whenever the coordinate $y^{2=5}$ is positioned in the feasible area and $0 \le k_{2,5} \le 1$, it holds that $k_{5,2} \ge k_{2,5}$.*

From Lemma C.6 we know that the minimum on the boundary of b_2 can only be positioned in the feasible area whenever $k_3 \ge \alpha_3$. We can obtain $y^{5,2} = \left(\frac{-k_3+(1-\alpha_3)(1+y_2^{min})}{\alpha_3}, y_2^{min}\right)$, which could only be positioned in the feasible area whenever $k_3 \le (1+\alpha_3)(1+y_2^{min})$.

Thus, we have to prove that $k_{5,2} - k_{2,5} \ge 0$ whenever $\alpha_3 \ge \alpha_3 \ge (1+\alpha_3)(1+y_2^{min})$. Note that this also means that $\alpha_3 \le \frac{1}{2-\alpha_2}$. We can derive that $\alpha_3 \le \frac{1+y_2^{min}}{2+y_2^{min}}$:

$$
\begin{aligned}
k_1^{3,4} &= -\frac{(1-\alpha_3)^2(1-k_3+y_2^{min})^2}{(1-k_3+(1-\alpha_3)y_2^{min})(-k_3(1-\alpha_3)-\alpha_3(1-k_3)+(1+y_2^{min})(1-\alpha_3))}k_2 \\
&\quad + -\frac{\alpha_3^2(1-2k_3+y_2^{min})(1+y_2^{min})}{(1-k_3+(1-\alpha_3)y_2^{min})(-k_3(1-\alpha_3)-\alpha_3(1-k_3)+(1+y_2^{min})(1-\alpha_3))}, \\
k_1^{4,3} &= -\frac{(1-\alpha_3^2)(1-k_3+y_2^{min})^2}{(1-k_3+(1-\alpha_3)y_2^{min})(1-k_3+(1+\alpha_3)y_2^{min})} \le 0k_2 \\
&\quad \frac{\alpha_3^2(1+y_2^{min})^2}{(1-k_3+(1-\alpha_3)y_2^{min})(1-k_3+(1+\alpha_3)y_2^{min})}.
\end{aligned}
$$

We can prove $k_1^{5,2} - k_1^{2,5} \ge 0$ by using:

$$
1-k_3-\alpha_3+y_2^{min} \ge 1-2\alpha_3+y_2^{min} \ge 1-\frac{2(1+y_2^{min})}{(2+y_2^{min})}+y_2^{min} \ge \frac{y_2^{min}(1+y_2^{min})}{2+y_2^{min}} \ge 0.
$$

and using:

$$
\begin{aligned}
1-k_3+(1-\alpha_3)y_2^{min} &\ge (1-k_3)(1-\alpha_3)-\alpha_3(1-k_3)+(1-\alpha_3)y_2^{min} \\
&\ge -k_3(1-\alpha_3)-\alpha_3(1-k_3)+(1-\alpha_3)(1+y_2^{min}) \\
&\ge -(1-\alpha_3)^2(1+y_2^{min})-\alpha_3(1-\alpha_3)_3)+(1-\alpha_3)(1+y_2^{min}) \\
&\ge \alpha_3(1-\alpha_3)y_2^{min} \ge 0.
\end{aligned}
$$

we can find:

$$
\begin{aligned}
k_1^{5,2} - k_1^{2,5} = &\left(\frac{(1-\alpha_3)^2(1-k_3+y_2^{min})^2}{(1-k_3+(1-\alpha_3)y_2^{min})(-k_3(1-\alpha_3)-\alpha_3(1-k_3)+(1+y_2^{min})(1-\alpha_3))}\right. \\
&\left. -\frac{(1-\alpha_3^2)(1-k_3+y_2^{min})^2}{(1-k_3+(1-\alpha_3)y_2^{min})(1-k_3+(1+\alpha_3)y_2^{min})}k_2\right) k_2 \\
&+ \frac{2\alpha_3^2(1-k_3+y_2^{min})(1-k_3-\alpha_3+y_2^{min})(1+y_2^{min})}{(1-k_3+(1-\alpha_3)y_2^{min})(1-k_3+(1+\alpha_3)y_2^{min})(-k_3(1-\alpha_3)-\alpha_3(1-k_3)+(1+y_2^{min})(1-\alpha_3))} \ge 0.
\end{aligned}
$$

Lemma C.12 *Whenever the coordinate $y^{2=4}$ is positioned in the feasible area $0 \le k_{2,4} \le 1$, it holds that $k_{4,2} \ge k_{2,4}$.*

From Lemma C.6 we know that the minimum on the boundary of b_2 can only be positioned in the feasible area whenever $k_3 \geq \alpha_3$. We can obtain $y^{2,4} = \left(y_1^{min}, \frac{\alpha_3(1+y_1^{min})-(1-k_3)}{1-\alpha_3}\right)$, which could only be positioned in the feasible area for $k_3 \geq 1-\alpha_3(1+y_1^{min})$.

Thus, we have to prove that $k_1^{4,2} - k_1^{2,4} \geq 0$ whenever $\alpha_3 \leq k_3 \wedge 1-\alpha_3(1+y_1^{min}) \leq 0$.

$$\begin{aligned}
k_1^{2,4} &= -\frac{(k_3+y_1^{min})^2}{(1+y_1^{min})(-1+2k_3+y_1^{min})}k_2 - \frac{(k_3+\alpha_3 y_1^{min})(\alpha_3 y_1^{min} - k_3(1-2\alpha_3))}{(1-\alpha_3)^2(1+y_1^{min})(-1+2k_3+y_1^{min})},\\
k_1^{4,2} &= -\frac{(k_3+y_1^{min})^2}{(1+y_1^{min})^2} + \frac{(k_3+(2-\alpha_3)y_1^{min})(k_3+\alpha_3 y_1^{min})}{(1-\alpha_3)^2(1+y_1^{min})(-1+2k_3+y_1^{min})}.
\end{aligned}$$

Using:

$$\begin{aligned}
-1+\alpha_3+k_3+y_1^{min} &\geq -1+2\alpha_3+y_1^{min}\\
&\geq (1-\alpha_3)y_1^{min} \geq 0.
\end{aligned}$$

we can find that:

$$\begin{aligned}
k_1^{4,2} - k_1^{2,4} = &\frac{2(1-k_3)(k_3+y_1^{min})^2}{(1+y_1^{min})^2(1+2k_3+y_1^{min})}k_2\\
&+ \frac{2(k_3+y_1^{min})(-1+k_3+\alpha_3+y_1^{min})(k_3+\alpha_3 y_1^{min})}{(1-\alpha_3)^2(1+y_1^{min})^2(-1+2k_3+y_1^{min})} \geq 0.
\end{aligned}$$

Lemma C.13 *Whenever the coordinate $y^{4=5}$ is positioned in the feasible area, it holds that $k_{4,2} \geq k_{2,4}$.*

We can find:

$$\begin{aligned}
k_1^{4,5} &= \frac{-k_2(k_3+y_1^{min})^2 + (1+y_2^{min})(1+2y_1^{min}+y_2^{min})}{(1+y_1^{min})^2},\\
k_1^{5,4} &= \frac{k_2(k_3+y_1^{min})(k_3-2(1+y_2^{min})-y_1^{min}) + (1+y_2^{min})^2}{(1+y_1^{min})(1+y_1^{min}+2y_2^{min})}.
\end{aligned}$$

From these expressions for $k_1^{4,5}$ and $k_1^{5,4}$ we can derive:

$$k_1^{4,5} - k_1^{5,4} = \frac{2(1+y_1^{min}+y_2^{min})(k_2(1-k_3)(k_3+y_1^{min}) + (1+y_2^{min})(y_1^{min}+y_2^{min}))}{(1+y_1^{min})(1+y_1^{min}+2y_2^{min})}. \geq 0$$

Combining Lemma C.5 until Lemma C.13 we can see that the arrows annotated to the boundaries of constraints b_i , $i = 1,2,4,5$ shown in Figure C.8 visualize the direction in which the constrained minimum moves when increasing k_1 (in the range $[0,1]$) when we consider the constrained problem with the objective shown in (C.7) and constraints b_i, $i = 1,2,4,5$. ■

Lets again consider the optimization problem with constraints b_1 until b_7. We know that the constrained minimum is positioned on the boundary of b_1, b_2 ,b_4 or b_5. We can obtain expressions for the constraint minimum on each of these lines. We use k_i^{min} for the smallest value for k such that the unconstrained minimum $\underline{y}^{b_i}(k)$ is positioned in the feasible area. We use k_i^{max} for the largest value for k

such that the unconstrained minimum $\underline{y}^{b_i}(k)$ is positioned in the feasible area. Whenever the boundary of constraint b_i is not positioned in the feasible area (and hence there is no constrained minimum on this boundary) it holds that $k_i^{min} > k_i^{max}$. Below we give the expressions for k_i^{min}, $i = 1, 2, 4, 5$ and k_i^{max}. Note that in these expressions z_i^{min}, $i = 1, 2, 4, 5$ and z_i^{max}, $i = 1, 2, 4, 5$ actually refer to positions where b_i intersects with another boundary. From the expressions for z_i^{min} and z_i^{max} we can calculate k_i^{min}. Recall that for $\alpha_3 \leq \alpha_2$ the boundary of b_3 is always positioned below the boundary of b_2. Thus, the boundary of $\alpha_3 \leq \alpha_2$ is not positioned in the feasible area.

Constrained minimum on the boundary of b_1:

$$\begin{aligned}
\underline{y}^{b_2}(k_1^{min}) &\text{ if } k_1 \leq k_1^{min} \leq k_1^{max},\\
\underline{y}^{b_2}(k_1) &\text{ if } k_1^{min} \leq k_1 \leq k_1^{max},\\
\underline{y}^{b_2}(k_1^{max}) &\text{ if } k_1^{min} \leq k_1^{max} \leq k_1,
\end{aligned}$$

where

$$\begin{aligned}
k_1^i &= -\frac{\alpha_2^2}{(1-\alpha_2)^2} + \frac{1 - k_2(2 - k_3 + z_1^i)(k_3 + z_1^i)}{(1 + z_1^i)^2}, \qquad i \in \{min, max\},\\
z_1^{min} &= \begin{cases} \min\{y_1^{max}, \frac{1-\alpha_2}{\alpha_2} y_2^{max} - 1, \frac{1-k_3(1-\alpha_2)-\alpha_3}{\alpha_3-\alpha_2}\} & \text{if } \alpha_3 > \alpha_2,\\ \min\{y_1^{max}, \frac{1-\alpha_2}{\alpha_2} y_2^{max} - 1\} & \text{if } \alpha_3 \leq \alpha_2,\end{cases}\\
z_1^{max} &= \max\{y_1^{min}, \frac{1-\alpha_2}{\alpha_2} y_2^{min} - 1, \frac{\alpha_1}{1-\alpha_1-\alpha_2}\}.
\end{aligned}$$

Constrained minimum on the boundary of b_2:

$$\begin{aligned}
\underline{y}^{b_1}(k_2^{min}) &\text{ if } k_1 \leq k_2^{min} \leq k_2^{max},\\
\underline{y}^{b_1}(k_1) &\text{ if } k_2^{min} \leq k_1 \leq k_2^{max},\\
\underline{y}^{b_1}(k_2^{max}) &\text{ if } k_2^{min} \leq k_2^{max} \leq k_1,
\end{aligned}$$

where

$$\begin{aligned}
k_2^i &= \begin{cases} \frac{k_3^2(1-\alpha_3)^2 - (z_2^i + k_3)^2(k_2(1-\alpha_3)^2 + \alpha_3^2)}{(1+z_2^i)(-1+z_2^i+2k_3)(1-\alpha_3)^2} & \text{if } \alpha_3 > \alpha_2,\\ -\infty & \text{if } \alpha_3 \leq \alpha_2,\end{cases} \qquad i \in \{min, max\},\\
z_2^{min} &= \min\{y_1^{max}, \frac{-k_3 + (1-\alpha_3)(1 + y_2^{max})}{\alpha_3}\},\\
z_2^{max} &= \max\{y_1^{min}, \frac{-k_3 + (1-\alpha_3)(1 + y_2^{min})}{\alpha_3}, \frac{k_3\alpha_1}{1-\alpha_1-\alpha_3}, \frac{1 - k_3(1-\alpha_2) - \alpha_3}{\alpha_3 - \alpha_2}\}.
\end{aligned}$$

Constrained minimum on the boundary of b_4:

$$\begin{aligned}
\underline{y}^{b_4}(k_4^{min}) &\text{ if } k_1 \leq k_4^{min} \leq k_4^{max},\\
\underline{y}^{b_4}(k_1) &\text{ if } k_4^{min} \leq k_1 \leq k_4^{max},\\
\underline{y}^{b_4}(k_4^{max}) &\text{ if } k_4^{min} \leq k_4^{max} \leq k_1,
\end{aligned}$$

where

$$\begin{aligned}
k_4^i =& \frac{-k_2(k_3 + y_1^{min})^2 + (1 + z_4^i)(1 + 2y_1^{min} + z_4^i)}{(1 + y_1^{min})^2}, \qquad i \in \{min, max\}, \\
z_4^{min} =& \min\{y_2^{min}, \frac{\rho_2}{1-\rho_2}(1 + y_1^{min}), \frac{\rho_3 y_1^{min} + k_3}{1-\rho_3} - 1\}, \\
z_4^{max} =& \max\{y_2^{max}, -1 + \frac{1-\rho_1}{\rho_1} y_1^{min}\}.
\end{aligned}$$

Constrained minimum on the boundary of b_5:

$$\begin{aligned}
\underline{y}^{b_5}(k_5^{min}) &\text{ if } k_1 \le k_5^{min} \le k_5^{max}, \\
\underline{y}^{b_5}(k_1) &\text{ if } k_5^{min} \le k_1 \le k_5^{max}, \\
\underline{y}^{b_5}(k_5^{max}) &\text{ if } k_5^{min} \le k_5^{max} \le k_1,
\end{aligned}$$

where

$$\begin{aligned}
k_5^i =& \frac{(1 + y_2^{min})^2 + k_2(k_3 - 2(1 + y_2^{min}) - z_5^i)(k_3 + z_5^i)}{(1 + z_5^i)(1 + 2y_2^{min} + z_5^i)}, \qquad i \in \{min, max\}, \\
z_5^{min} =& \min\{y_1^{max}, \frac{1-\alpha_2}{\alpha_2} y_2^{min} - 1, \frac{-k_3 + (1-\alpha_3)(1 + y_2^{min})}{\alpha_3}\}, \\
z_5^{max} =& \max\{y_1^{min}, \frac{\alpha_1}{1-\alpha_1}(1 + y_2^{min})\}.
\end{aligned} \tag{C.23}$$

Using Lemma C.2 we can find an analytical solution for the position of the constrained minimizer $\left(y_1^{con,min}, y_2^{con,min}\right)$:

$$\left(y_1^{con,min}, y_2^{con,min}\right) = \begin{cases}
\underline{y}^{b_2}(k_2^{min}) & \text{if } k_1 \le k_2^{min} \le k_2^{max}, \\
\underline{y}^{b_2}(k_1) & \text{else if } k_2^{min} < k_1 < k_2^{max}, \\
\underline{y}^{b_2}(k_2^{max}) & \text{else if } k_1^{max} < k_1^{min} \wedge k_5^{max} < k_5^{min} \wedge k_4^{max} < k_4^{min}, \\
\underline{y}^{b_1}(k_1^{min}) & \text{else if } k_1 \le k_1^{min} \le k_1^{max}, \\
\underline{y}^{b_1}(k_1) & \text{else if } k_1^{min} < k_1 < k_1^{max}, \\
\underline{y}^{b_1}(k_1^{max}) & \text{else if } k_5^{max} < k_5^{min} \wedge k_4^{max} < k_4^{min}, \\
\underline{y}^{b_5}(k_5^{min}) & \text{else if } k_1 \le k_5^{min} \le k_5^{max}, \\
\underline{y}^{b_5}(k_1) & \text{else if } k_5^{min} < k_1 < k_5^{max}, \\
\underline{y}^{b_5}(k_5^{max}) & \text{else if } k_4^{max} < k_4^{min}, \\
\underline{y}^{b_4}(k_4^{min}) & \text{else if } k_1 \le k_4^{min} \le k_4^{max}, \\
\underline{y}^{b_4}(k_1) & \text{else if } k_4^{min} < k_1 < k_4^{max}, \\
\underline{y}^{b_4}(k_4^{max}) & \text{otherwise.}
\end{cases}$$

where

$$\underline{y}^{b_1}(k_1) = \left(-1 + \frac{(1-\alpha_2)\sqrt{1 + k_2(1-k_3)^2}}{\sqrt{(k_1 + k_2)(1-\alpha_2)^2 + \alpha_2^2}}, \frac{\alpha_2\sqrt{1 + k_2(1-k_3)^2}}{\sqrt{(k_1 + k_2)(1-\alpha_2)^2 + \alpha_2^2}}\right),$$

$$
\begin{aligned}
\underline{y}^{b_2}(k_1) &= \left(-k_3 + \frac{(1-\alpha_3)\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}}, -(1-k_3) + \frac{\alpha_3\sqrt{k_1(1-k_3)^2+k_3^2}}{\sqrt{(k_1+k_2)(1-\alpha_3)^2+\alpha_3^2}}\right),\\
\underline{y}^{b_4}(k_1) &= \left(y_1^{min}, -(1+y_1^{min}) + \sqrt{(y_1^{min})^2 + k_1(1+y_1^{min})^2 + k_2(y_1^{min}+k_3)^2}\right),\\
\underline{y}^{b_5}(k_1) &= \left(-(1+y_2^{min}) + \frac{\sqrt{(1+y_2^{min})^2 + k_1(y_2^{min})^2 + k_2(1+y_2^{min}-k_3)^2}}{\sqrt{k_1+k_2}}, y_2^{min}\right),\\
k_1^i &= -\frac{\alpha_2^2}{(1-\alpha_2)^2} + \frac{1-k_2(2-k_3+z_1^i)(k_3+z_1^i)}{(1+z_1^i)^2}, \qquad i \in \{min, max\},\\
z_1^{min} &= \begin{cases} \min\{y_1^{max}, \frac{1-\alpha_2}{\alpha_2}y_2^{max}-1, \frac{1-k_3(1-\alpha_2)-\alpha_3}{\alpha_3-\alpha_2}\} & \text{if } \alpha_3 > \alpha_2\\ \min\{y_1^{max}, \frac{1-\alpha_2}{\alpha_2}y_2^{max}-1\} & \text{if } \alpha_3 \le \alpha_2 \end{cases},\\
z_1^{max} &= \max\{y_1^{min}, \frac{1-\alpha_2}{\alpha_2}y_2^{min}-1, \frac{\alpha_1}{1-\alpha_1-\alpha_2}\}\\
k_2^i &= \begin{cases} \frac{k_3^2(1-\alpha_3)^2-(z_2^i+k_3)^2(k_2(1-\alpha_3)^2+\alpha_3^2)}{(1+z_2^i)(-1+z_2^i+2k_3)(1-\alpha_3)^2} & \text{if } \alpha_3 > \alpha_2\\ -\infty & \text{if } \alpha_3 \le \alpha_2 \end{cases}, \quad i \in \{min, max\},\\
z_2^{min} &= \min\{y_1^{max}, \frac{-k_3+(1-\alpha_3)(1+y_2^{max})}{\alpha_3}\},\\
z_2^{max} &= \max\{y_1^{min}, \frac{-k_3+(1-\alpha_3)(1+y_2^{min})}{\alpha_3}, \frac{k_3\alpha_1}{1-\alpha_1-\alpha_3}, \frac{1-k_3(1-\alpha_2)-\alpha_3}{\alpha_3-\alpha_2}\},\\
k_4^i &= \frac{-k_2(k_3+y_1^{min})^2 + (1+z_4^i)(1+2y_1^{min}+z_4^i)}{(1+y_1^{min})^2}, \qquad i \in \{min, max\},\\
z_4^{min} &= \min\{y_2^{min}, \frac{\rho_2}{1-\rho_2}(1+y_1^{min}), \frac{\rho_3 y_1^{min}+k_3}{1-\rho_3}-1\},\\
z_4^{max} &= \max\{y_2^{max}, -1+\frac{1-\rho_1}{\rho_1}y_1^{min}\},\\
k_5^i &= \frac{(1+y_2^{min})^2 + k_2(k_3-2(1+y_2^{min})-z_5^i)(k_3+z_5^i)}{(1+z_5^i)(1+2y_2^{min}+z_5^i)}, \qquad i \in \{min, max\},\\
z_5^{min} &= \min\{y_1^{max}, \frac{1-\alpha_2}{\alpha_2}y_2^{min}-1, \frac{-k_3+(1-\alpha_3)(1+y_2^{min})}{\alpha_3}\},\\
z_5^{max} &= \max\{y_1^{min}, \frac{\alpha_1}{1-\alpha_1}(1+y_2^{min})\}.
\end{aligned}
$$

C.3 Proof of Lemma 7.2

Below we give the proof of Lemma 7.2. This lemma is given on page 74.

Proof. Lets consider a trajectory defined on the time interval $[0,\infty)$ where a queue is not emptied at least once or where the duration of the green periods is not always the same for a signal. Lets call this trajectory the 'original trajectory'. In Figure C.11a we can see an example of the original trajectory.

We introduce the following notation for the average duration of g_i^k, r_i^k, $g_i^{\lambda,k}$ and $g_i^{\mu,k}$:

$$\bar{g}_i = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_i^k}{M}, \ \ i \in \mathcal{N}, \tag{C.24a}$$

$$\bar{r}_i = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{r_i^k}{M}, \ \ i \in \mathcal{N}, \tag{C.24b}$$

$$\bar{g}_i^{\lambda} = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_i^{\lambda,k}}{M}, i \in \mathcal{N}, \tag{C.24c}$$

$$\bar{g}_i^{\mu} = \lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_i^{\mu,k}}{M}, i \in \mathcal{N}. \tag{C.24d}$$

$$\tag{C.24e}$$

We can propose an alternative trajectory where a queue is always emptied during a green period and where the green time of a signal is always the same (see Figure C.11b). For this alternative trajectory we take the green times and red times of signal $i \in \mathcal{N}$ equal to respectively $\bar{g}_i$ and $\bar{r}_i$. We serve signal $i_1 \in \mathcal{G}_1$ during the red period of the signals in $\mathcal{G}_2$ and we serve $i_2 \in \mathcal{G}_2$ during the red period of the signals in $\mathcal{G}_1$.

We can show that the costs J related to this alternative trajectory are not greater than the costs related to the original trajectory.

First we prove that the limits in (C.24) exist. We assume that the limits $\lim_{M\to\infty} \sum_{k=1}^{M} \frac{g_i^k}{M}$, $i \in \mathcal{N}$ exist and that the limits $\lim_{M\to\infty} \sum_{k=1}^{M} \frac{r_i^k}{M}$, $i \in \mathcal{N}$ exist (see Section 7.2.3).

Whenever signal $i \in \mathcal{N}$ satisfies $\lambda_i \bar{r}_i \neq (\mu_i - \lambda_i)\bar{g}_i^{\mu}$ for the original trajectory, this means that the queue length of queue i would go to ∞ or $-\infty$ because:

$$\lim_{t\to\infty} x_i(t) = \lim_{M\to\infty} \sum_{k=1}^{M} (\lambda_i r_i^k - (\mu_i - \lambda_i) g_i^{\mu,k}) = \lim_{M\to\infty} M, (\lambda_i \bar{r}_i - (\mu_i - \lambda_i)\bar{g}_i^{\mu}).$$

Note that we have used that each green time of signal i is finite. A queue length must be a non-negative number and therefore a trajectory where a queue length goes to $-\infty$ is not feasible. Further, whenever a queue length goes to ∞, the costs calculated with (7.1) are infinite. Hence, it must hold that:

$$\lambda_i \bar{r}_i = (\mu_i - \lambda_i)\bar{g}_i^{\mu}, i = 1, 2. \tag{C.25}$$

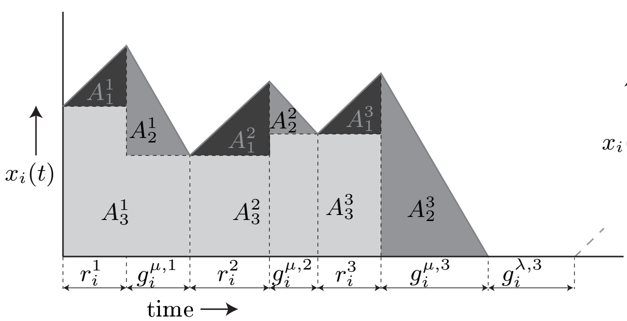
Thus, the amount of traffic that arrives during a red period of signal $i \in \mathcal{N}$ is equal to $\lambda_i \bar{r}_i$ and we can let this amount of traffic depart during a period equal to exactly $\bar{g}_i^{\mu}$. As a result, from $\bar{g}_i = \bar{g}_i^{\mu} + \bar{g}_i^{\lambda}$ we can obtain that for the alternative policy the length of the slow mode is equal to $\bar{g}_i^{\lambda}$ during each green period. From (C.25) we can see that $\bar{g}_i^{\mu}$ exists (because $\bar{r}_i$ exists) and from $\bar{g}_i = \bar{g}_i^{\mu} + \bar{g}_i^{\lambda}$ we know that $\bar{g}_i^{\lambda}$ exists. Hence, $\bar{g}_i$, $\bar{r}_i$, $\bar{g}_i^{\lambda}$ and $\bar{g}_i^{\mu}$ all exist.

Also note that the alternative trajectory is always feasible. First of all, the green periods of the alternative trajectory (with duration $\bar{g}_i$) always take longer than the shortest green period of the original trajectory. Second of all, the green periods of the alternative trajectory (with duration $\bar{g}_i$) always take shorter than the longest green period of the original trajectory. Furthermore, the maximum queue length are less for the alternative trajectory than for original trajectory. As a result, whenever the original trajectory satisfies (7.3d) until (7.3k), the alternative trajectory does as well.

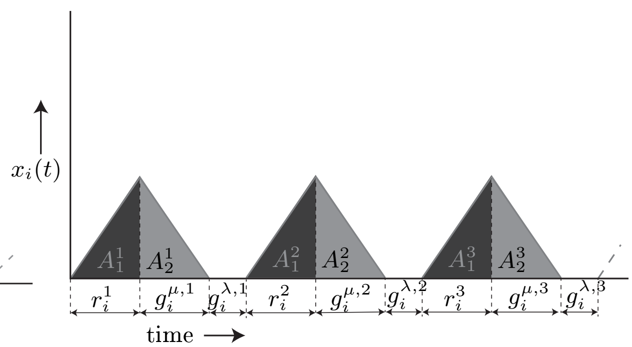
Now we prove that the costs related to the alternative trajectory are not bigger than the costs of the original trajectory. We use $b_{g_i^{\mu,k}}$, $k \geq 1$ and $b_{r_i^k}$, $k \geq 1$ for the time at which the green period g_i^k starts respectively the time at which the red period r_i^k starts. Further, we use $e_{g_i^{\mu,k}}$, $k \geq 1$ for the time at which queue i is emptied during g_i^k and we use $e_{r_i^k}$, $k \geq 1$ for the time at which r_i^k ends. We distinguish three different areas (see Figure C.11): A_1^k,$k \geq 1$, A_2^k, $k \geq 1$ and A_3^k, $k \geq 1$.

$$
\begin{aligned}
A_1^k &= \int_{b_{g_i^{\mu,k}}}^{e_{g_i^{\mu,k}}} (x_i(t) - x_i(b_{g_i^{\mu,k}}))dt, && k \geq 1,\\
A_2^k &= \int_{b_{r_i^k}}^{e_{r_i^k}} (x_i(t) - x_i(e_{r_i^k}))dt, && k \geq 1,\\
A_3^k &= x_i(b_{g_i^{\mu,k}})(e_{g_i^{\mu,k}} - b_{g_i^{\mu,k}}) + x_i(e_{r_i^k})(e_{r_i^k} - b_{r_i^k}), && k \geq 1.
\end{aligned}
$$

In Figure C.11, A_1^k is visualized in dark gray, A_2^k is visualized in medium gray and A_3^k is visualized in light gray.



(a) Queue length of signal i for an example of the original trajectory.



(b) Queue length of signal i for the alternative trajectory.

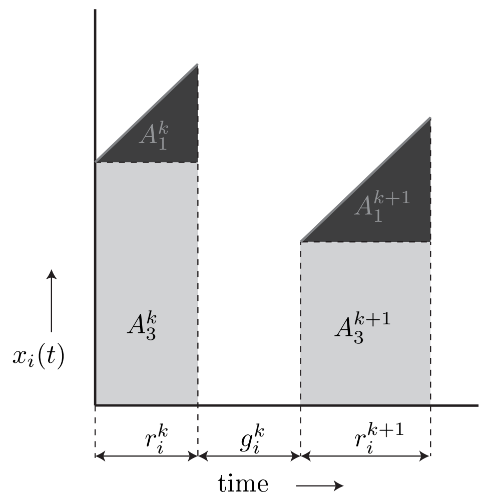
Figure C.11: Visualization of the original trajectory and the alternative trajectory.

Because the queues are always emptied for the alternative trajectory, it holds that $A_3^k = 0$, $k \geq 1$ for this trajectory.

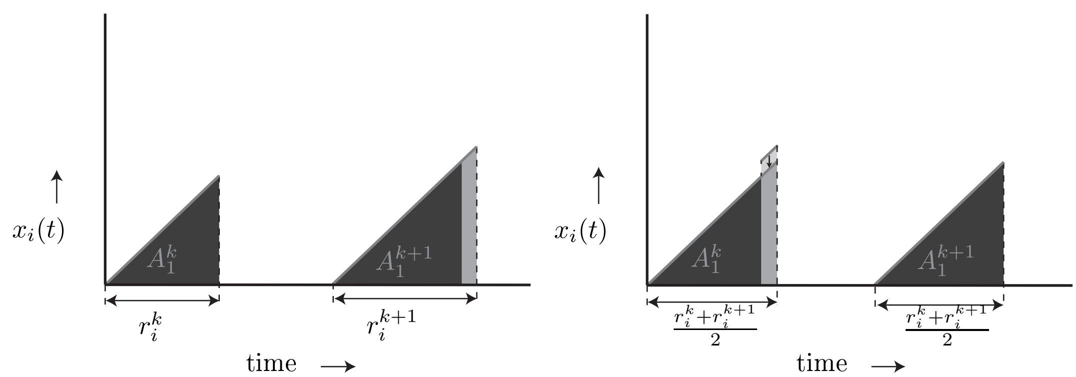
Now we prove that the costs related to signal i and made during only the red periods are not bigger for the alternative trajectory than for original trajectory. Thus, we only consider the signal during the red periods of signal $i \in \mathcal{N}$, i.e. we cut out the parts where signal i is green (see Figure C.12a).

Now we can shift each and every red period towards the time axis for the original trajectory, i.e. removing the areas A_3^k. Note that since f_i is strictly increasing, shifting the red periods of the original trajectory towards the time axis cannot increase the costs related to the red periods of signal i.

On the left side of Figure C.12b we can see A_1^k and A_1^{k+1} plotted for the shifted original trajectory. Without loss of generality we can assume that the first red period r_i^k is longer than the second red period r_i^{k+1} for two adjacent red periods. When we take both green times equal to $\frac{r_i^k + r_i^{k+1}}{2}$ we get the areas A_1^k and A_1^{k+1} as can be seen on the right side of Figure C.12b. We can see that the dark gray areas are the same and that the medium gray areas differ (the difference is the light gray area). Since f_i is strictly increasing, taking the red time of two adjacent red periods equal to each other cannot increase the costs related to the red periods of signal i. Hence, taking all red periods equal to each other cannot increase the costs related to the red periods of signal i. Note, that the costs, of this shifted trajectory where all red periods are equal to each other, are exactly the costs made during the red periods of the alternative trajectory. Thus, the costs related to the red periods of the alternative trajectory cannot be bigger than the costs related to the red periods of the original trajectory.



(a) Visualization of only the red periods of the original trajectory.



(b) Left: visualization of the shifted red periods of the original trajectory, right: 2 equal red periods instead of 2 unequal red periods.

Figure C.12: Comparing the costs made during the red periods for both trajectories.

In exactly the same way we can show that the costs related to the green periods of signal $i \in \mathcal{N}$ cannot be bigger for the alternative trajectory than for the original trajectory. Hence, the costs of the alternative trajectory are not bigger than the costs of the original trajectory.

Thus, whenever we are given a trajectory that does not satisfy the property given in this lemma, we can always give an alternative trajectory that does satisfy this property and that works at least as good. Hence, there must be an optimal trajectory that satisfies the property given in this lemma. ■

Appendix D

Proof of Proposition 8.1 (Regulation)

Below we give the proof of Proposition 8.1. This proposition is given on page 95. Before reading this proof we advice you to read the overview of this proof given on page 8.1.

In this overview we used five different reasons to switch signal $i_c^{r,f}$,$c = 1, 2$ to red: *switch*.1*a*, *switch*.1*b*, *switch*.2, *switch*.3*a* and *switch*.3*b*. In Section D.1 we elaborate on these different reasons to switch a signal from green to red. In Section D.2 we present some notation and definitions used in the proof of Proposition 8.1. All lemmas used in this proof are shown in Section D.4.

D.1 Different reasons to switch

In this section we show 5 different reasons to switch a signal to red.

Recall that τ_i is used for the time that has elapsed since the last mode change of signal $i \in \mathcal{N}$.

When signal $i_c^{g,f}$ is green, we use $\tau^C_{i_c^{g,f}}$ for the smallest value of $\tau_{i_c^{g,f}}$ for which condition $C \in \{1.1, 1.2, 1.3, 2, 3\}$ is satisfied during this green time. See Section 8.3.1 and Section 8.3.2 for more information about these conditions. Further we use $\tau^1_{i_c^{g,f}}$ for the smallest value of $\tau_{i_c^{g,f}}$ for which conditions 1.1, 1.2 and 1.3 are all satisfied:

$$\tau^1_{i_c^{g,f}} = \max\{\tau^{1.1}_{i_c^{g,f}}, \tau^{1.2}_{i_c^{g,f}}, \tau^{1.3}_{i_c^{g,f}}\}.$$

We switch the signals in the set $\mathcal{G}_c$ for the following reasons:

switch.1 We switch because of the reason *switch*.1 whenever we switch signal $\tau_{i_c}^{r,f}$ to red when $\tau^1_{i_c^{g,f}} \leq \tau^2_{i_c^{g,f}} \wedge \tau^1_{i_c^{g,f}} \leq \tau^3_{i_c^{g,f}}$. We split *switch*.1 into *switch*.1*a* and *switch*.1*b*:

- *switch*.1*a* We switch because of the reason *switch*.1*a* whenever we switch signal $\tau_{i_c}^{r,f}$ to red when $\tau^1_{i_c^{g,f}} \leq \tau^2_{i_c^{g,f}} \wedge \tau^1_{i_c^{g,f}} \leq \tau^3_{i_c^{g,f}}$ and $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} \left(x_{i_{\overline{c}}}(t) > x^\sharp_{i_{\overline{c}}} \right)$.
- *switch*.1*b* We switch because of the reason *switch*.1*b* whenever we switch signal $\tau_{i_c}^{r,f}$ to red when $\tau^1_{i_c^{g,f}} \leq \tau^2_{i_c^{g,f}} \wedge \tau^1_{i_c^{g,f}} \leq \tau^3_{i_c^{g,f}}$ and $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}} \left(x_{i_{\overline{c}}}(t) \leq x^\sharp_{i_{\overline{c}}} \right)$.

switch.2 We switch because of the reason *switch*.2 whenever we switch signal $\tau_{i_c}^{r,f}$ to red when $\tau^2_{i_c^{g,f}} < \tau^1_{i_c^{g,f}} \wedge \tau^2_{i_c^{g,f}} \leq \tau^3_{i_c^{g,f}}$. Thus, we switch because otherwise the maximum green time would be exceeded.

switch.3 We switch because of the reason *switch*.3whenever $\tau^3_{i_c^{g,f}} < \tau^1_{i_c^{g,f}} \wedge \tau^3_{i_c^{g,f}} < \tau^2_{i_c^{g,f}}$. Thus, we switch because otherwise a queue would overflow. We split *switch*.3 into *switch*.3*a* and *switch*.3*b*:

switch.3*a* We switch because of the reason *switch*.3*a* whenever we switch signal $\tau^{r,f}_{i_c}$ to red when $\tau^3_{i_c^g,f} < \tau^1_{i_c^g,f} \wedge \tau^3_{i_c^g,f} < \tau^2_{i_c^g,f}$ and the queue(s) that would overflow first if we do not switch, was (where) not emptied during its (their) previous green time. In Figure D.1 we show the situation where we switch signal 1 to red because of the reason *switch*.3*a*. The queue of signal $i_2 \in \mathcal{G}_2$ would overflow if we did not switch signal 1 to red. We can see that during the previous green time of signal i_2, queue i_2 was not emptied.

switch.3*b* We switch because of the reason *switch*.3*a* whenever we switch signal $\tau^{r,f}_{i_c}$ to red when $\tau^3_{i_c^g,f} < \tau^1_{i_c^g,f} \wedge \tau^3_{i_c^g,f} < \tau^2_{i_c^g,f}$ and a queue that would overflow first if we did not switch, was emptied during its previous green time. In Figure D.2 we show the situation where we switch signal 1 to red because of the reason *switch*.3*b*. The queue of signal $i_2 \in \mathcal{G}_2$ would overflow if we did not switch the signal 1 to red. We can see that during the previous green time of signal i_2, queue i_2 was emptied (points 2 and 3 could also overlap).

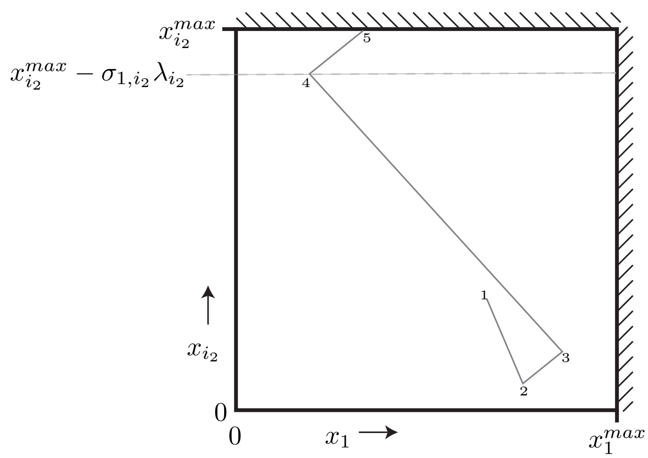


Figure D.1: Visualization of the switch reason *switch*.3*a*.

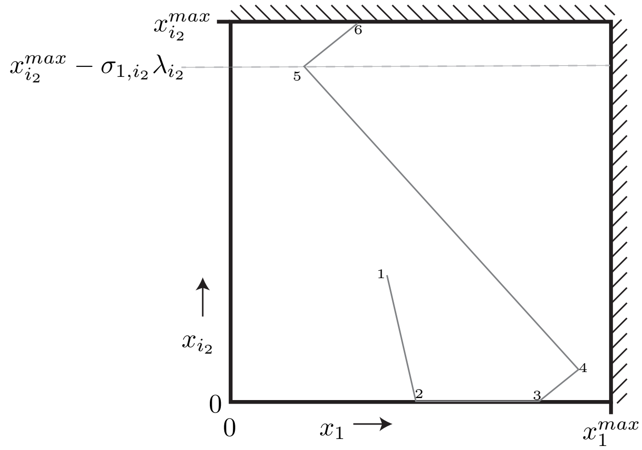


Figure D.2: Visualization of the switch reason *switch.3b*.

D.2 Notation and definitions

In this section we introduce some notations and definitions.

D.2.1 Referring to Signals and Sets

With i_1 and l_1 we refer to two different signals in the signal group $\mathcal{G}_1$:

$$i_1 \in \mathcal{G}_1, \qquad l_1 \in \mathcal{G}_1, \qquad i_1 \neq l_1.$$

With i_2 and l_2 we refer to two different signals in the signal group $\mathcal{G}_2$:

$$i_2 \in \mathcal{G}_2, \qquad l_2 \in \mathcal{G}_2, \qquad i_2 \neq l_2.$$

Further we use:

$$c = 1, 2,$$
$$\overline{c} = \begin{cases} 1 & \text{if c=2,} \\ 2 & \text{if c=1.} \end{cases}$$

Using c and $\overline{c}$ we are able to express to cases at once: the case where $c = 1$ and $\overline{c} = 2$ and the case where $c = 2$ and $\overline{c} = 1$.

Further, we use i_c^k to refer to a signal in the set $\mathcal{G}_c$ which queue length is equal to $x_{i_c}^{max}$ at the start of $g_{i_c}^k$. Queue i_c is active whenever a queue i_c reaches its maximum queue length.

D.2.2 Green times and Red times

We use g_i^k, $i \in \mathcal{N}$ to refer to the kth green time of signal $i \in \mathcal{N}$. We assume without loss of generality that we start serving the signals in signal group 1. Further we use r_i for the red period of signal i that comes between the kth and $(k+1)$th green period of signal i:

$$r_{i_1}^k = g_{i_2}^k + \sigma_{i_1,i_2,i_1}, \qquad i_1 \in \mathcal{G}_1, i_2 \in \mathcal{G}_2, \tag{D.1a}$$
$$r_{i_2}^k = g_{i_1}^{k+1} + \sigma_{i_1,i_2,i_1}, \qquad i_1 \in \mathcal{G}_1, i_2 \in \mathcal{G}_2. \tag{D.1b}$$

From D.1 we can obtain that the green periods are related according to:

$$g_{i_1}^k + \sigma_{i_1,i_2,i_1} = g_{i_1}^k + \sigma_{l_1,i_2,l_1} \quad i_1, l_1 \in \mathcal{G}_1, \quad i_2 \in \mathcal{G}_2, \tag{D.2a}$$
$$g_{i_2}^k + \sigma_{i_1,i_2,i_1} = g_{l_2}^k + \sigma_{i_1,l_2,i_1} \qquad i_1 \in \mathcal{G}_1, \quad i_2, l_2 \in \mathcal{G}_2. \tag{D.2b}$$

When using c and $\overline{c}$ we use that $g_{i_{\overline{c}}}^{f(k)}$ comes between $g_{i_c}^k$ and $g_{i_c}^{k+1}$. We can find:

$$f(k) = \begin{cases} \text{k} & \text{if c=1,} \\ \text{k+1} & \text{if c=2.} \end{cases}$$

Further, we split up g_i^k, $i \in \mathcal{N}$ in $g_i^{\mu,k}$ and $g_i^{\lambda,k}$. We use $g_i^{\lambda,k}$ for the length of the slow mode at signal i during the kth green period of signal i and we use $g_i^{\mu,k}$ for the length of the interval during the kth green period of signal i during which the queue of signal i is not empty.

Further, we use g_{i_1,i_2}^{pbt} to refer to the green time of signal i_1 of the pure bow tie curve in the (i_1, i_2)-plane and we use g_{i_2,i_1}^{pbt} to refer to the green time of signal i_2 of the pure bow tie curve in the (i_1, i_2)-plane:

$$g_{i_1,i_2}^{pbt} = \frac{\sigma_{i_1,i_2,i_1}\rho_1}{1-\rho_{i_1}-\rho_{i_2}}, \tag{D.3a}$$
$$g_{i_2,i_1}^{pbt} = \frac{\sigma_{i_1,i_2,i_1}\rho_2}{1-\rho_{i_1}-\rho_{i_2}}. \tag{D.3b}$$

Further we use:

$$g_{i_1}^{pbt} = \max_{i_2 \in \mathcal{G}_2} g_{i_1,i_2}^{pbt}, \tag{D.4a}$$
$$g_{i_2}^{pbt} = \max_{i_1 \in \mathcal{G}_1} g_{i_2,i_1}^{pbt}. \tag{D.4b}$$
$$\tag{D.4c}$$

From Lemma D.17 we know that the queues in the set $\mathcal{S}_c$ are the only queues in $\mathcal{G}_c$ that could go from empty to their maximum queue lengths without exceeding a maximum queue length and that a queue $i_s \in \mathcal{S}_c$ can only go from empty to its maximum queue length whenever all queues in the set $\mathcal{S}_c$ go from empty to their maximum queue lengths.

It must hold that $r_{i_s}^k \leq \frac{x_{i_s}^{max}}{\lambda_{i_s}}$, $i_s \in \mathcal{S}_c$, $\forall k \geq 1$ because otherwise the maximum queue length of queue i_s (and the maximum queue lengths of all other signals in the set $\mathcal{G}_c$) would be exceeded. Hence, using (D.1) we can see that the green time of signal i_c cannot be larger than $\frac{x_{i_s}^{max}}{\lambda_{i_s}} - \sigma_{i_c,i_s,i_c}$, $i_s \in \mathcal{S}_{\overline{c}}$. Hence, we can find that:

$$g_{i_c}^k \le \tilde{g}_{i_c}^{max}, \qquad i_c \in \mathcal{G}_c, \tag{D.5a}$$
$$r_{i_c}^k \le \tilde{r}_{i_c}^{max}, \qquad i_c \in \mathcal{G}_c, \tag{D.5b}$$

where

$$\tilde{g}_{i_c}^{max} = \min\{g_{i_c}^{max}, \frac{x_{i_s}^{max}}{\lambda_{i_s}} - \sigma_{i_c,i_s,i_c}\}, \qquad i_c \in \mathcal{G}_c, \quad i_s \in \mathcal{S}_{\overline{c}}, \tag{D.5c}$$
$$\tilde{r}_{i_c}^{max} = \min\{g_{i_{\overline{c}}}^{max}, \frac{x_{i_s}^{max}}{\lambda_{i_s}} - \sigma_{i_{\overline{c}},i_s,i_{\overline{c}}}\} + \sigma_{i_{\overline{c}},i_c,i_{\overline{c}}}, \qquad i_c \in \mathcal{G}_c, \quad i_{\overline{c}} \in \mathcal{G}_{\overline{c}}, \quad i_s \in \mathcal{S}_c. \tag{D.5d}$$

Note that $g_{i_c}^k = \tilde{g}_{i_c}^{max} = g_{i_c}^{max}$ whenever $s_c^k = switch.2$ and that $g_{i_c}^k = \tilde{g}_{i_c}^{max} = \frac{x_{i_s^{max}}}{\lambda_{i_s}} - \sigma_{i_c,i_s,i_c}$ whenever $s_c^k = switch.3b$. Further, note that:

$$\tilde{r}_{i_c}^{max} = \tilde{g}_{i_{\overline{c}}}^{max} + \sigma_{i_c,i_{\overline{c}},i_c}, \forall i_c \in \mathcal{G}_c, \forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}. \tag{D.6}$$

D.2.3 Definition of Cycle 1 and Cycle 2

In this section we introduce two different cycles. Cycle 1 starts when signal $i_1^{g,f}$ switches to green and ends when signal $i_1^{g,f}$ switches to green. Cycle 2 starts when signal $i_2^{g,f}$ switches to green and ends when signal $i_2^{g,f}$ switches to green.

We can distinguish the following phases for the kth cycle 1.

phase 1 finish the setup σ_{i_2,i_1}, which still has a duration of $\sigma_{i_1}^{res}$
phase 2 perform $g_{i_1}^k$.
phase 3 perform the setup σ_{i_1,i_2}
phase 4 perform $g_{i_2}^k$
phase 5 Perform the setup σ_{i_2,i_1} until the signal $i_1^{g,f}$ switches to green, which has a duration of $\sigma_{i_2,i_1^{g,f}}$.

We can distinguish the following phases for the kth cycle 2.

phase 1 finish the setup σ_{i_1,i_2}, which still has a duration of $\sigma_{i_2}^{res}$
phase 2 perform $g_{i_2}^k$.
phase 3 perform the setup σ_{i_2,i_1}
phase 4 perform $g_{i_2}^{k+1}$
phase 5 Perform the setup σ_{i_1,i_2} until the signal $i_2^{g,f}$ switches to green, which has a duration of $\sigma_{i_1,i_2^{g,f}}$.

We use x_{1,i_c}^k for the queue length at queue $i_c \in \mathcal{G}_c$ at the beginning of the kth cycle 1. Similarly, we use x_{2,i_c}^k for the queue length at queue $i_c \in \mathcal{G}_c$ at the beginning of the kth cycle 2. We can find the following expressions:

$$\Delta x_{1,i_1}^{k+1} = x_{1,i_1}^{k+1} - x_{1,i_1}^k = (g_{i_2}^k + \sigma_{i_1,i_2,i_1})\lambda_{i_1} - g_{i_1}^{\mu,k}(\mu_{i_1} - \lambda_{i_1}), \tag{D.7a}$$
$$\Delta x_{1,i_2}^{k+1} = x_{1,i_2}^{k+1} - x_{1,i_2}^k = (g_{i_1}^k + \sigma_{i_1,i_2,i_1})\lambda_{i_2} - g_{i_2}^{\mu,k}(\mu_{i_2} - \lambda_{i_2}), \tag{D.7b}$$
$$\Delta x_{2,i_1}^{k+1} = x_{2,i_1}^{k+1} - x_{2,i_1}^k = (g_{i_2}^k + \sigma_{i_1,i_2,i_1})\lambda_{i_1} - g_{i_1}^{\mu,k+1}(\mu_{i_1} - \lambda_{i_1}), \tag{D.7c}$$
$$\Delta x_{2,i_2}^{k+1} = x_{2,i_2}^{k+1} - x_{2,i_2}^k = (g_{i_1}^{k+1} + \sigma_{i_1,i_2,i_1})\lambda_{i_2} - g_{i_2}^{\mu,k}(\mu_{i_2} - \lambda_{i_2}). \tag{D.7d}$$

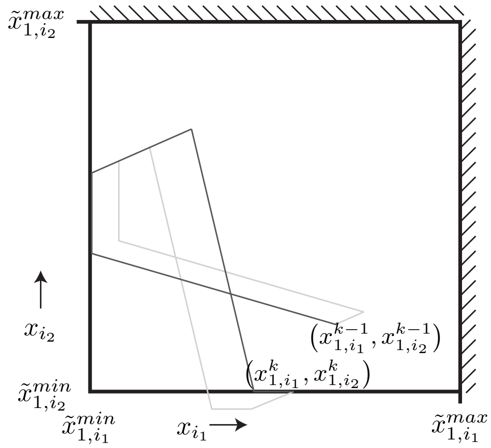


Figure D.3: Mapping of cycle 1. The mapping is shown in dark gray. We show actual queue length in light gray whenever it differs from the mapping.

Further we can derive:

$$x^{k+1}_{1,i_1} = x^{k}_{2,i_1} + (\sigma^{res}_{i_2} + g^{k}_{i_2} + \sigma_{i_2,i_1^{g,f}})\lambda_{i_1}, \tag{D.8a}$$

$$x^{k}_{2,i_2} = x^{k}_{1,i_2} + (\sigma^{res}_{i_1} + g^{k}_{i_1} + \sigma_{i_1,i_2^{g,f}})\lambda_{i_2}. \tag{D.8b}$$

D.2.4 Mappings

Instead of using the actual evolution of the queue length we often use mappings from the queue lengths at the beginning of a cycle (either cycle 1 or cycle 2) to the queue lengths at the end of this cycle.

In the previous section we showed the phases that we distinguish for the kth cycle 1. For the mapping from $\left(x^{k-1}_{1,i_1}, x^{k-1}_{1,i_2}\right)$, $i_1 \in \mathcal{G}_1$, $i_2 \in \mathcal{G}_2$ $k > 0$ to $\left(x^{k}_{1,i_1}, x^{k}_{1,i_2}\right)$ we change the order of the phases to the following order: phase 2, phase 1, phase 3, phase 5, phase 4 (see Figure D.3). Note that changing the order of these phases does not change the queue length at the end of a cycle.

In Figure D.3 we show the mapping from $\left(x^{k-1}_{1,i_1}, x^{k-1}_{1,i_2}\right)$, $i_1 \in \mathcal{G}_1$, $i_2 \in \mathcal{G}_2$ $k \geq 1$ to $\left(x^{k}_{1,i_1}, x^{k}_{1,i_2}\right)$. The actual queue length can range from zero to the maximum queue length. Hence, for this mapping the feasible range of x_{i_1} is equal to:

$$\left[-t^{f}_{i_1}\lambda_{i_1}, x^{max}_{i_1} - t^{f}_{i_1}\lambda_{i_1}\right] = \left[\tilde{x}^{min}_{1,i_1}, \tilde{x}^{max}_{1,i_1}\right].$$

For this mapping the feasible range of x_{i_2} is equal to:

$$\left[\sigma_{i_2,i_1^f}\lambda_{i_2}, x^{max}_{i_2} + \sigma_{i_2,i_1^f}\lambda_{i_2}\right] = \left[\tilde{x}^{min}_{1,i_2}, \tilde{x}^{max}_{1,i_2}\right].$$

.

In the previous section we showed the phases that we distinguish for the kth cycle 2. For the mapping from $\left(x^{k-1}_{2,i_1}, x^{k-1}_{2,i_2}\right)$, $i_1 \in \mathcal{G}_1$, $i_2 \in \mathcal{G}_2$ $k > 0$ to $\left(x^{k}_{2,i_1}, x^{k}_{2,i_2}\right)$ we change the order of the phases to the following order: phase 2, phase 1, phase 3, phase 5, phase 4 (see Figure D.4).

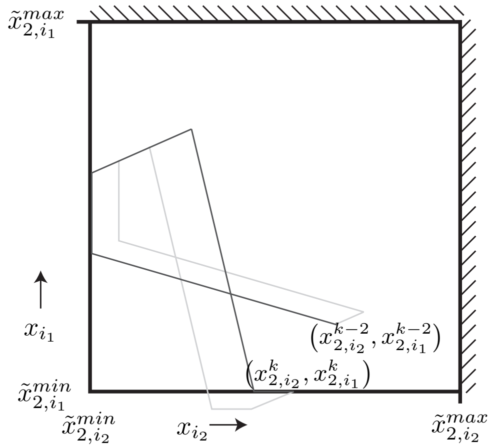


Figure D.4: Mapping of cycle 2. The mapping is shown in dark gray. We show actual queue length in light gray whenever it differs from the mapping.

For this mapping the feasible range of x_{i_1} is equal to:

$$\left[\sigma_{i_1,i_2^f}\lambda_{i_1}, x^{max}_{i_1} + \sigma_{i_1,i_2^f}\lambda_{i_1}\right] = \left[\tilde{x}^{min}_{2,i_1}, \tilde{x}^{max}_{2,i_1}\right].$$

and for this mapping the feasible range of x_{i_2} is equal to:

$$\left[-t^f_{i_2}\lambda_{i_2}, x^{max}_{i_2} - t^f_{i_2}\lambda_{i_2}\right] = \left[\tilde{x}^{min}_{2,i_2}, \tilde{x}^{max}_{2,i_2}\right].$$

Using some of the introduced notation we can summarize (D.8) in one equation:

$$x^{f(k)}_{\bar{c},i_{\bar{c}}} = x^k_{c,i_{\bar{c}}} + r^{f(k)-1}_{i_{\bar{c}}}\lambda_{i_{\bar{c}}} + \tilde{x}^{min}_{\bar{c},i_{\bar{c}}} - \tilde{x}^{min}_{c,i_{\bar{c}}}. \tag{D.9}$$

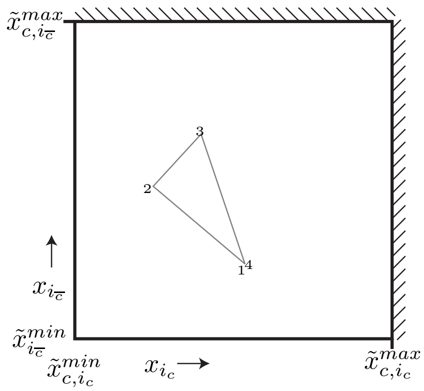
Also note, that queue $i_c \in \mathcal{G}_c$ is full at the beginning of $g^k_{i_c}$ iff $x^k_{c,i_c} = \tilde{x}^{max}_{c,i_c}$ and that queue $i_c \in \mathcal{G}_c$ is empty at the end of $g^k_{i_c}$ iff $x^{f(k)}_{\bar{c},i_c} = \tilde{x}^{min}_{\bar{c},i_c}$.

In Figure D.5a we show the mapping of the pure bow-tie curve. In Figure D.5b we can see the case where the green time of signal i_c exceeds $g^{pbt}_{i_c,i_{\bar{c}}}$ and the green time of signal $i_{\bar{c}}$ exceeds $g^{pbt}_{i_{\bar{c}},i_c}$ and both signals do not have a slow mode during these green times.

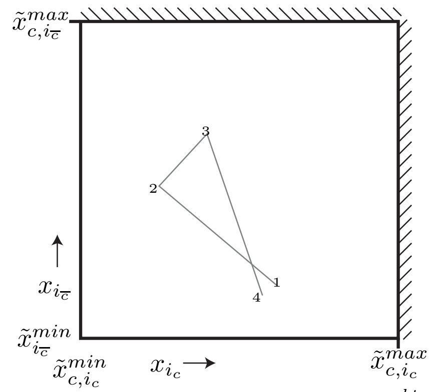
D.3 Proof of the Policy

In this section we prove policy proposed in Section 8.1 makes sure that a trajectory converges to the desired trajectory. We use s^k_c for the reason why we stopped the kth green period of the signals in the set $\mathcal{G}_c$.

We want to prove that every switch reason s^k_c, $k > 1$ is part of a combination in the set C_i, $i = 1, \ldots, n_c$. Further, we want to prove that when s^k_c, $k \geq n^{\mathcal{C}_1} + 1$ (where $n^{\mathcal{C}_1}$ is a finite integer) is part of a combination in the set $\mathcal{C}_i$, $i = 1, \ldots, n_s$ then $s^{f(k)}_{\bar{c}}$ cannot be part of a combination $\mathcal{C}_j$, $1 \leq j < i$. We explain later what the exact definition of $n^{\mathcal{C}_1}$ is. Further, we want to prove that only a finite number of adjacent switch reasons can be part of a combination in the set $\mathcal{C}_i$, $1 \leq i \leq n_s - 2$ (either with the previous or the next switch reason).



(a) Mapping of the pure bow-tie curve.



(b) Mapping when the green time of signal i_c exceeds $g^{pbt}_{i_c,i_{\overline{c}}}$, the green time of signal $i_{\overline{c}}$ exceeds $g^{pbt}_{i_{\overline{c}},i_c}$ and both signals do not have a slow mode during these green times.

First of all we want to prove that $s_c^k, >1$ is a part of combination C_i, $i = 1, \dots, n_s$. In the simplified overview of the proof (on page 95) we told that we consider combinations of 2 subsequent green periods. However, this is not entirely true. In fact we consider combinations of 2 subsequent green periods in combination with some knowledge about queue lengths. In Table D.1 we have shown the combinations that we consider. Further, in this table we have shown these combinations are partitioned in the sets $\mathcal{C}_i$, $i = 1, \dots, n_s$. Note that we consider all combinations of two subsequent switch reasons shown in Table D.2. In Table D.3 we can see all possible combinations of three subsequent switch reasons. In this table we can see that s_c^k is always part of a combination, except for the cases 37, 53, 60 and 61.

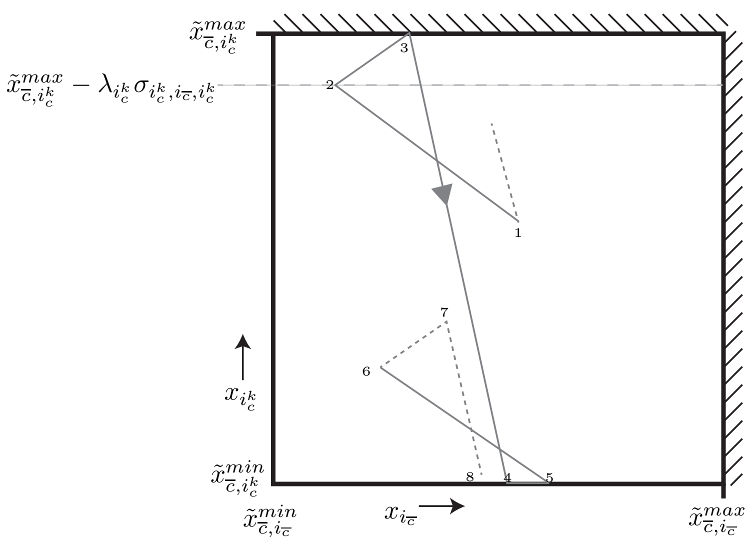


Figure D.5: When $s_{\overline{c}}^{f(k)-1} \to s_c^k \to s_{\overline{c}}^{f(k)} = switch.3a \to switch.1a \to switch.2$ then it holds that $s_c^{k-1} = switch.3a$ if $k > 1$

Combination	$s_c^k \to s_{\overline{c}}^{f(k)}$	Δ	In the set
C_1	$switch.1a \to switch.1a$	-	$\mathcal{C}_{11}$
C_2	$switch.1a \to switch.1b$	-	$\mathcal{C}_{12}$
	$switch.1a \to switch.3a$	-	not possible [1]
C_3	$switch.1a \to switch.3b$	-	$\mathcal{C}_{10}$
C_4	$switch.1b \to switch.1b$	-	$\mathcal{C}_{12}$
C_5	$switch.2 \to switch.1a$	-	$\mathcal{C}_9$
C_6	$switch.2 \to switch.1b$	-	$\mathcal{C}_{12}$
C_7	$switch.2 \to switch.2$	$\max_{i_{\overline{c}} \in \mathcal{G}_c} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$	$\mathcal{C}_7$
C_8	$switch.2 \to switch.2$	$\max_{i_{\overline{c}} \in \mathcal{G}_c} \Delta x_{c,i_{\overline{c}}}^{k+1} > 0 \vee \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} > 0$	$\mathcal{C}_2$
C_9	$switch.2 \to switch.3a$	$\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$	$\mathcal{C}_8$
	$switch.2 \to switch.3a$	$\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} > 0$	not possible [2]
C_{10}	$switch.2 \to switch.3b$	$\max_{i_{\overline{c}} \in \mathcal{G}_c} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$	$\mathcal{C}_6$
C_{11}	$switch.2 \to switch.3b$	$\max_{i_{\overline{c}} \in \mathcal{G}_c} \Delta x_{c,i_{\overline{c}}}^{k+1} > 0$	$\mathcal{C}_4$
C_{12}	$switch.3a \to switch.1b$	-	$\mathcal{C}_{12}$
C_{13}	$switch.3a \to switch.3a$	-	$\mathcal{C}_1$
C_{14}	$switch.3a \to switch.3b$	$\max_{i_{\overline{c}} \in \mathcal{G}_c} \Delta x_{c,i_{\overline{c}}}^{k+1} < 0$	$\mathcal{C}_3$
C_{15}	$switch.3a \to switch.3b$	$\max_{i_{\overline{c}} \in \mathcal{G}_c} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$	$\mathcal{C}_4$
C_{16}	$switch.3b \to switch.1b$	-	$\mathcal{C}_{12}$
	$switch.3b \to switch.3b$	$\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} < 0 \vee \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} < 0$	not possible [3]
C_{17}	$switch.3b \to switch.3b$	$\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} = 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} = 0$	$\mathcal{C}_5$
C_{18}	$switch.3b \to switch.3b$	$\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} > 0$	$\mathcal{C}_4$
C_{19}	$switch.3b \to switch.3b$	$\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} > 0$	$\mathcal{C}_4$

[1] From Lemma D.1 we know that this combination is not possible.

[2] Whenever $g_{i_c}^k$ is stopped for the reason $switch.2$ and $g_{i_{\overline{c}}}^{f(k)}$ is stopped for the reason of $switch.3a$ then $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c^{k+1}}^{k+1} \geq 0$ (because a queue in the set $\mathcal{G}_c$ has reached its maximum queue length at the start of $g_{i_c}^{k+1}$), $g_{i_c^{k+1}}^k = g_{i_c^{k+1}}^{max} > g_{i_c^{k+1}}^{pbt}$ (because of inequalities (8.1h) and (8.1k)) and $g_{i_c^{k+1}}^{\mu,k} = g_{i_c^{k+1}}^k$ (because otherwise we stopped because of $switch.3b$ instead of $switch.3a$). Now we can see from Lemma D.18 that when $s_c^k \to s_{\overline{c}}^{f(k)} = switch.2 \to switch.3a$ then it holds that $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^k \leq 0$.

[3] Whenever $g_{i_c}^k$ is stopped for the reason $switch.3b$ and $g_{i_{\overline{c}}}^{f(k)}$ is stopped for the reason $switch.3b$ then $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} < 0 \vee \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} < 0$ cannot occur. First of all, there is a queue in the set $i_c \in \mathcal{G}_c$ that has a maximum queue length at the start of $g_{i_c}^{k+1}$. Hence, $\Delta x_{c,i_c^{k+1}}^{k+1} \geq 0$. Furthermore, because we stop $g_{i_c}^k$ for the reason $switch.3b$, there is at least one signal $i_{\overline{c}}$ that is emptied during $g_{i_{\overline{c}}}^{f(k)-1}$. This signal satisfies $x_{c,i_{\overline{c}}}^k = \tilde{x}_{c,i_{\overline{c}}}^{min}$. In general it must hold that $x_{c,i_c}^k \geq \tilde{x}_{c,i_{\overline{c}}}^{min}$, $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$, $k \geq 1$ ($x_{c,i_c}^k < \tilde{x}_{c,i_{\overline{c}}}^{min}$ implies a negative queue length). Hence it holds that the signal that satisfies $x_{c,i_{\overline{c}}}^k = \tilde{x}_{c,i_{\overline{c}}}^{min}$ also satisfies $\Delta x_{c,i_{\overline{c}}}^{k+1} \geq 0$. Therefore $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \geq 0$.

Table D.1: The combinations that we consider and how we partition these combinations into sets.

	$s_c^k \to s_{\bar{c}}^{f(k)}$
1	$switch.1a \to switch.1a$
2	$switch.1a \to switch.1b$
3	$switch.1a \to switch.3a$
4	$switch.1a \to switch.3b$
5	$switch.1b \to switch.1b$
6	$switch.2 \to switch.1a$
7	$switch.2 \to switch.1b$
8	$switch.2 \to switch.2$
9	$switch.2 \to switch.3a$
10	$switch.2 \to switch.3b$
11	$switch.3a \to switch.1b$
12	$switch.3a \to switch.3a$
13	$switch.3a \to switch.3b$
14	$switch.3b \to switch.1b$
15	$switch.3b \to switch.3b$

Table D.2: We consider all combinations of two subsequent switch reasons shown in this table.

case	$s_{\bar{c}}^{f(k)-1} \to s_c^k \to s_{\bar{c}}^{f(k)}$	comment
1	$\mathrlap{\overbrace{\phantom{\mathit{switch.1b} \to \mathit{switch.1b}}}}\mathit{switch.1b} \to \underbrace{\mathit{switch.1b} \to \mathit{switch.1b}}$	
2	$\mathrlap{\overbrace{\phantom{* \to \mathit{switch.1b}}}}* \to \underbrace{\mathit{switch.1b} \to \mathit{switch.1b}}$	
3	$* \to \overbrace{* \to \mathit{switch.1b}}$	
4	$\mathrlap{\overbrace{\phantom{\mathit{switch.1a} \to \mathit{switch.1a}}}}\mathit{switch.1a} \to \underbrace{\mathit{switch.1a} \to \mathit{switch.1a}}$	
5	$\overbrace{\mathit{switch.1a} \to \mathit{switch.1a}} \to \mathit{switch.2}$	
6	$\mathrlap{\overbrace{\phantom{\mathit{switch.1a} \to \mathit{switch.1a}}}}\mathit{switch.1a} \to \underbrace{\mathit{switch.1a} \to \mathit{switch.3a}}$	
7	$\mathrlap{\overbrace{\phantom{\mathit{switch.1a} \to \mathit{switch.1a}}}}\mathit{switch.1a} \to \underbrace{\mathit{switch.1a} \to \mathit{switch.3b}}$	
8	$\mathit{switch.1a} \to \underbrace{\mathit{switch.2} \to \mathit{switch.1a}}$	
9	$\mathit{switch.1a} \to \underbrace{\mathit{switch.2} \to \mathit{switch.2}}$	
10	$\mathit{switch.1a} \to \underbrace{\mathit{switch.2} \to \mathit{switch.3a}}$	
11	$\mathit{switch.1a} \to \underbrace{\mathit{switch.2} \to \mathit{switch.3b}}$	
12	$\overbrace{\mathit{switch.1a} \to \mathit{switch.3a}} \to \mathit{switch.1a}$	
13	$\overbrace{\mathit{switch.1a} \to \mathit{switch.3a}} \to \mathit{switch.2}$	
14	$\mathrlap{\overbrace{\phantom{\mathit{switch.1a} \to \mathit{switch.3a}}}}\mathit{switch.1a} \to \underbrace{\mathit{switch.3a} \to \mathit{switch.3a}}$	
15	$\mathrlap{\overbrace{\phantom{\mathit{switch.1a} \to \mathit{switch.3a}}}}\mathit{switch.1a} \to \underbrace{\mathit{switch.3a} \to \mathit{switch.3b}}$	
16	$\overbrace{\mathit{switch.1a} \to \mathit{switch.3b}} \to \mathit{switch.1a}$	
17	$\overbrace{\mathit{switch.1a} \to \mathit{switch.3b}} \to \mathit{switch.2}$	
18	$\overbrace{\mathit{switch.1a} \to \mathit{switch.3b}} \to \mathit{switch.3a}$	
19	$\mathrlap{\overbrace{\phantom{\mathit{switch.1a} \to \mathit{switch.3b}}}}\mathit{switch.1a} \to \underbrace{\mathit{switch.3b} \to \mathit{switch.3b}}$	
20	$\mathrlap{\overbrace{\phantom{\mathit{switch.2} \to \mathit{switch.1a}}}}\mathit{switch.2} \to \underbrace{\mathit{switch.1a} \to \mathit{switch.1a}}$	
21	$\overbrace{\mathit{switch.2} \to \mathit{switch.1a}} \to \mathit{switch.2}$	
22	$\mathrlap{\overbrace{\phantom{\mathit{switch.2} \to \mathit{switch.1a}}}}\mathit{switch.2} \to \underbrace{\mathit{switch.1a} \to \mathit{switch.3a}}$	
23	$\mathrlap{\overbrace{\phantom{\mathit{switch.2} \to \mathit{switch.1a}}}}\mathit{switch.2} \to \underbrace{\mathit{switch.1a} \to \mathit{switch.3b}}$	
24	$\overbrace{\mathit{switch.2} \to \mathit{switch.2}} \to \mathit{switch.1a}$	
25	$\mathrlap{\overbrace{\phantom{\mathit{switch.2} \to \mathit{switch.2}}}}\mathit{switch.2} \to \underbrace{\mathit{switch.2} \to \mathit{switch.2}}$	
26	$\overbrace{\mathit{switch.2} \to \mathit{switch.2}} \to \mathit{switch.3a}$	
27	$\overbrace{\mathit{switch.2} \to \mathit{switch.2}} \to \mathit{switch.3b}$	
28	$\overbrace{\mathit{switch.2} \to \mathit{switch.3a}} \to \mathit{switch.1a}$	
29	$\overbrace{\mathit{switch.2} \to \mathit{switch.3a}} \to \mathit{switch.2}$	
30	$\mathrlap{\overbrace{\phantom{\mathit{switch.2} \to \mathit{switch.3a}}}}\mathit{switch.2} \to \underbrace{\mathit{switch.3a} \to \mathit{switch.3a}}$	
31	$\mathrlap{\overbrace{\phantom{\mathit{switch.2} \to \mathit{switch.3a}}}}\mathit{switch.2} \to \underbrace{\mathit{switch.3a} \to \mathit{switch.3b}}$	
32	$\overbrace{\mathit{switch.2} \to \mathit{switch.3b}} \to \mathit{switch.1a}$	

Table D.3: All possible combinations of 3 subsequent switch reasons. When $s_c^k \to s_{\bar{c}}^{f(k)}$ is in Table D.1 this is visualized with braces. * means that this switch reason can be either one of the five switch reasons except for switch.1b. Note that for case 3, $s_{\bar{c}}^{f(k)-1} \to s_c^k$ could also be a combination from Table D.1.

case	$s_{\bar{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\bar{c}}^{f(k)}$	comment
33	$\overbrace{switch.2 \rightarrow switch.3b} \rightarrow switch.2$	
34	$\overbrace{switch.2 \rightarrow switch.3b} \rightarrow switch.3a$	
35	$\rlap{\overbrace{}}switch.2 \rightarrow \underbrace{switch.3b \rightarrow switch.3b}$	
36	$switch.3a \rightarrow \underbrace{switch.1a \rightarrow switch.1a}$	
37	$switch.3a \rightarrow switch.1a \rightarrow switch.2$	for more information see footnote [1]
38	$switch.3a \rightarrow \underbrace{switch.1a \rightarrow switch.3a}$	
39	$switch.3a \rightarrow \underbrace{switch.1a \rightarrow switch.3b}$	
40	$switch.3a \rightarrow \underbrace{switch.2 \rightarrow switch.1a}$	
41	$switch.3a \rightarrow \underbrace{switch.2 \rightarrow switch.2}$	
42	$switch.3a \rightarrow \underbrace{switch.2 \rightarrow switch.3a}$	
43	$switch.3a \rightarrow \underbrace{switch.2 \rightarrow switch.3b}$	
44	$\overbrace{switch.3a \rightarrow switch.3a} \rightarrow switch.1a$	
45	$\overbrace{switch.3a \rightarrow switch.3a} \rightarrow switch.2$	
46	$\rlap{\overbrace{}}switch.3a \rightarrow \underbrace{switch.3a \rightarrow switch.3a}$	
47	$\rlap{\overbrace{}}switch.3a \rightarrow \underbrace{switch.3a \rightarrow switch.3b}$	
48	$\overbrace{switch.3a \rightarrow switch.3b} \rightarrow switch.1a$	
49	$\overbrace{switch.3a \rightarrow switch.3b} \rightarrow switch.2$	
50	$\rlap{\overbrace{}}switch.3a \rightarrow \underbrace{switch.3b \rightarrow switch.3a}$	
51	$\rlap{\overbrace{}}switch.3a \rightarrow \underbrace{switch.3b \rightarrow switch.3b}$	
52	$switch.3b \rightarrow \underbrace{switch.1a \rightarrow switch.1a}$	
53	$switch.3b \rightarrow switch.1a \rightarrow switch.2$	not possible. See footnote [2]
54	$switch.3b \rightarrow \underbrace{switch.1a \rightarrow switch.3a}$	
55	$switch.3b \rightarrow \underbrace{switch.1a \rightarrow switch.3b}$	
56	$switch.3b \rightarrow \underbrace{switch.2 \rightarrow switch.1a}$	
57	$switch.3b \rightarrow \underbrace{switch.2 \rightarrow switch.2}$	
58	$switch.3b \rightarrow \underbrace{switch.2 \rightarrow switch.3a}$	
59	$switch.3b \rightarrow \underbrace{switch.2 \rightarrow switch.3b}$	
60	$switch.3b \rightarrow switch.3a \rightarrow switch.1a$	not possible. See footnote [3]
61	$switch.3b \rightarrow switch.3a \rightarrow switch.2$	not possible. See footnote [2]
62	$switch.3b \rightarrow \underbrace{switch.3a \rightarrow switch.3a}$	
63	$switch.3b \rightarrow \underbrace{switch.3a \rightarrow switch.3b}$	
64	$\overbrace{switch.3b \rightarrow switch.3b} \rightarrow switch.1a$	
65	$\overbrace{switch.3b \rightarrow switch.3b} \rightarrow switch.2$	
66	$\overbrace{switch.3b \rightarrow switch.3b} \rightarrow switch.3a$	
67	$\rlap{\overbrace{}}switch.3b \rightarrow \underbrace{switch.3b \rightarrow switch.3b}$	

[1] In Figure D.5 we show the case where $s_{\bar{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\bar{c}}^{f(k)} = switch.3a \rightarrow switch.1a \rightarrow switch.2$. We use i_c^k to refer to the queue that is full at the start of $g_{i_c}^k$. From $s_{\bar{c}}^{f(k)-1} \rightarrow s_c^k = switch.3a \rightarrow switch1a$ we know that we could always empty queue i_c^k during a green time of $\tilde{g}_{i_c}^{max}$ (because we where able to empty queue i_c^k during $g_{i_c}^k$). Furthermore, from $s_{\bar{c}}^{f(k)-1} = switch.3a$, we know that queue i_c^k was not emptied during $g_{i_c}^{k-1}$. Therefore $s_c^{k-1} = switch.1a$, $s_c^{k-1} = switch.1b$ are not possible (if $k > 1$). Further, $s_c^{k-1} = switch.2$ and $s_c^{k-1} = switch.3b$ are not possible if $k > 1$ because we could always empty queue i_c^k during a green time of $\tilde{g}_{i_c}^{max}$. Hence, it must hold that $s_c^{k-1} = switch.3a$.

[2] From Lemma D.2 we know that this sequence of switch reasons cannot occur.

[3] From Lemma D.3 we know that this sequence of switch reasons cannot occur.

Table D.3: All possible combinations of 3 subsequent switch reasons. When $s_c^k \rightarrow s_{\bar{c}}^{f(k)}$ is in Table D.1 this is visualized with an braces. * means that this switch reason can be either one of the five switch reasons (except for switch.1b). Note that for case 3, $s_{\bar{c}}^{f(k)-1} \rightarrow s_c^k$ could also be a combination from Table D.1.

We can prove that cases 53, 60 and 61 cannot occur (see the footnotes of Table D.3). For case 37 we can prove that when this combination of three subsequent switch reasons occurs then $s_c^{k-1} = switch.3a$ if $k > 1$ (see the first footnote of Table D.1). Thus, if case 37 occurs and $k > 1$ then $s_{\overline{c}}^{f(k)-1}$ is part of a combination in the set $\mathcal{C}_1$. Further, $s_{\overline{c}}^{f(k)}$ is part of a combination C_j, $j = 8, 9, \ldots, 17$ (see Table D.1). Hence, $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_i$, $i = 2, 3, 4, 5, 6, 7, 9, 11$. Whenever $s_{\overline{c}}^{f(k)-1} \to s_c^k \to s_{\overline{c}}^{f(k)} = switch.3a \to switch.1a \to switch.2$ occurs then we define s_c^k to be part of a combination in the set $\mathcal{C}_2$ (however according to Table D.1 it is not). Because of this definition, it still holds that every switch reason s_c^k, $k > 1$ is part of a combination C_i, $i = 1, 2, \ldots, 25$.

The first problem is that we have to prove that when s_c^k, $k \geq n^{\mathcal{C}_1} + 1$ is part of a combination in the set $\mathcal{C}_i$, $i = 1, \ldots, n_s$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set$\mathcal{C}_j$, $1 \leq j < i$. Note that this holds for $s_{\overline{c}}^{f(k)-1} \to s_c^k \to s_{\overline{c}}^{f(k)} = switch.3a \to switch.1a \to switch.2$ (case 37). We use $n^{\mathcal{C}_1}$ for the smallest number $k > 0$ such that s_1^k is not part of a combination in the set $\mathcal{C}_1$. We can prove (see Section D.3.1) that $n^{\mathcal{C}_1}$ is finite and that s_2^k, $k \geq n^{\mathcal{C}_1}$ and s_1^k, $k \geq n^{\mathcal{C}_1}$ cannot be part of a combination in the set $\mathcal{C}_1$ anymore.

The second problem is that we have to prove that only a finite sequence of switch reasons s_c^k can be part of a combination in the set $\mathcal{C}_i$, $1 \leq i \leq n_s - 2$.

In sections D.3.1-D.3.12 we consider these two problems for the sets $\mathcal{C}_1$ until $\mathcal{C}_{12}$.

D.3.1 s_c^k is part of a combination in the set $\mathcal{C}_1$

For the set $\mathcal{C}_1$ we have to show that an infinite number of adjacent switch reasons s_c^k that are all part of a combination in the set $\mathcal{C}_1$ is not possible.

In this section we assume an infinite sequence $s_1^1 \to s_2^{f(1)} \to s_1^2 \to s_2^{f(2)} \to \cdots = switch.3a \to switch.3a \to switch.3a \to switch.3a \to \ldots$. We prove that $g_{i_c}^k < g_{i_c}^{k+1}$, $\forall k > 0$ $\forall i_c \in \mathcal{G}_1 \cup \mathcal{G}_2$. Hence, an infinite sequence $s_c^k \to s_{\overline{c}}^{f(k)} \to \cdots = switch.3a \to switch.3a \to \ldots$ is not possible because eventually the green times are too big for $switch.3a$ to occur.

Increasing green times

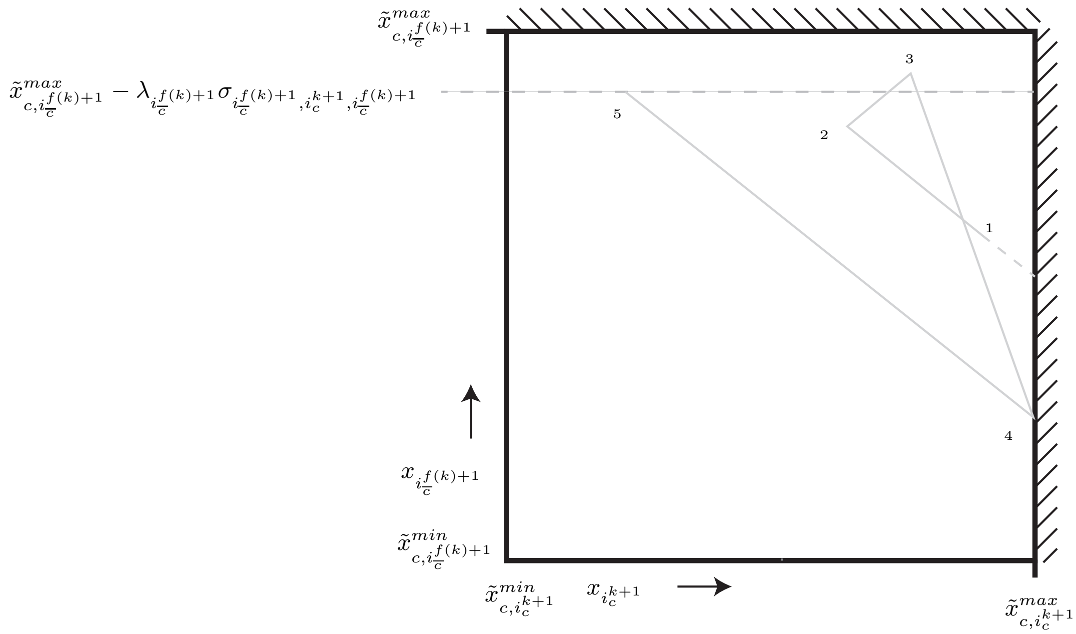
We show that $g_{i_1}^k < g_{i_1}^{k+1}$, $\forall i_1 \in \mathcal{G}_1$, $\forall k > 0$ and that $g_{i_2}^k < g_{i_2}^{k+1}$, $\forall i_2 \in \mathcal{G}_2$, $\forall k > 0$ by distinguishing the following two situations:

situation 1 $g_{i_{\overline{c}}^{f(k)+1}}^{f(k)} > g_{i_{\overline{c}}^{f(k)+1}, i_c^{k+1}}^{pbt}$.

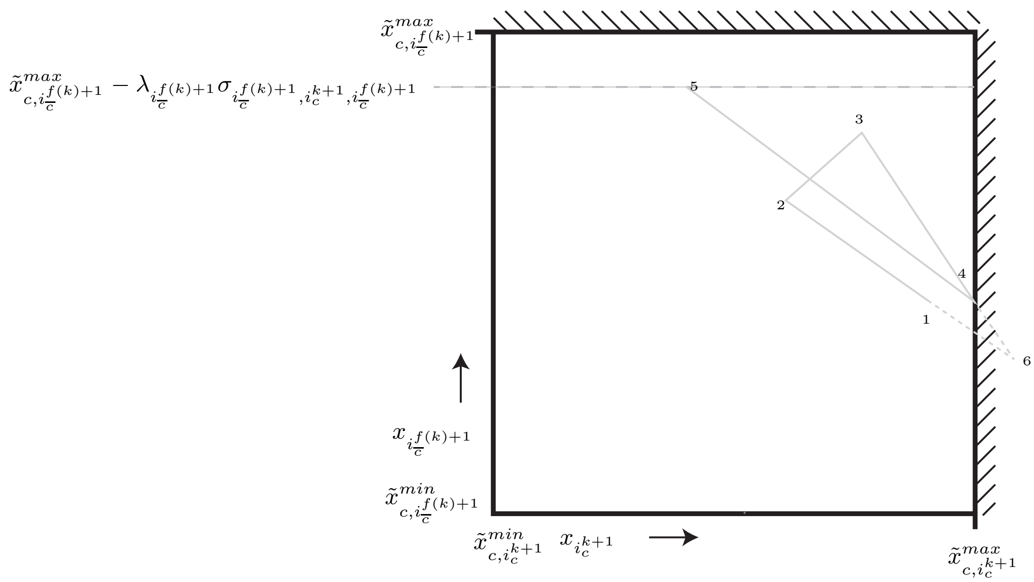
situation 2 $g_{i_{\overline{c}}^{f(k)+1}}^{f(k)} \leq g_{i_{\overline{c}}^{f(k)+1}, i_c^{k+1}}^{pbt}$.

We prove that in situation 1 it holds that $g_{i_c}^k < g_{i_c}^{k+1}$, $\forall i_c \in \mathcal{G}_c$ and that in situation 2 it holds that $g_{i_c}^k < g_{i_c}^{k+1}$, $\forall i_c \in \mathcal{G}_c$. Hence, it follows that $g_{i_1}^k < g_{i_1}^{k+1}$, $\forall i_1 \in \mathcal{G}_1$, $\forall k > 0$ and that $g_{i_2}^k < g_{i_2}^{k+1}$, $\forall i_2 \in \mathcal{G}_2$, $\forall k > 0$.

We have visualized situation 1 and situation 2 in Figure D.6.



(a) Situation 1



(b) Situation 2

Figure D.6: Situation 1 and situation 2

We can distinguish the following sections:

1 between point 1 and point 2, $g^k_{i_c^{k+1}}$ is performed.

2 between point 2 and point 3,the setup $\sigma_{i_{\overline{c}}^{f(k)+1}, i_c^{k+1}, i_{\overline{c}}^{f(k)+1}}$ is performed.

3 between point 3 and point 4, $g^{f(k)}_{i^{f(k)+1}_{\overline{c}}}$ is performed.

4 between point 4 and point 5, $g^{k+1}_{i^{k+1}_c}$ is performed.

Using (D.7a) and (D.7d) we can find that in situation 1 point 1 is positioned above point 4 because $g^{f(k)}_{i^{f(k)+1}_{\overline{c}}} > g^{pbt}_{i^{f(k)+1}_{\overline{c}},i^{k+1}_c}$ (see definition of situation 1), $g^{\mu,f(k)}_{i^{f(k)+1}_{\overline{c}}} = g^{f(k)}_{i^{f(k)+1}_{\overline{c}}}$ (queue $i^{f(k)+1}_{\overline{c}}$ is not emptied during $g^{f(k)}_{i^{f(k)+1}_{\overline{c}}}$ since $s^{k+1}_c = switch.3a$) and $\Delta x^{k+1}_{c,i^{k+1}_c} \geq 0$ (because queue i^{k+1}_c is full at the start of $g^{k+1}_{i^{k+1}_c}$). Further, point 2 cannot be positioned above point 5 (otherwise the maximum queue capacity would be exceeded). Hence, $g^{k}_{i^{k+1}_c} < g^{k+1}_{i^{k+1}_c}$. The green periods are related according to D.2. Hence, $g^{k}_{i^{k+1}_c} < g^{k}_{i^{k+1}_c}$ also means $g^{k}_{i_c} < g^{k+1}_{i_c}$, $\forall i_c \in \mathcal{G}_c$.

In situation 2, it holds that $g^{f(k)}_{i^{f(k)+1}_{\overline{c}}} \leq g^{pbt}_{i^{f(k)+1}_{\overline{c}},i^{k+1}_c}$ which could happen only if $g^{k}_{i^{k+1}_c} \leq g^{pbt}_{i^{k+1}_c,i^{f(k)+1}_{\overline{c}}}$ (Between the points 2,3 and 6 we shows the mapping of the pure bow-tie curve). We can see this using (D.7a) and (D.7b), $g^{f(k)}_{i^{f(k)+1}_{\overline{c}}} \leq g^{pbt}_{i^{f(k)+1}_{\overline{c}},i^{k+1}_c}$ (see definition of situation 2), $g^{\mu,f(k)}_{i^{f(k)+1}_{\overline{c}}} = g^{f(k)}_{i^{f(k)+1}_{\overline{c}}}$ (queue $i^{f(k)+1}_{\overline{c}}$ is not emptied during $g^{f(k)}_{i^{f(k)+1}_{\overline{c}}}$ since $s^{k+1}_c = switch.3a$) and $\Delta x^{k+1}_{c,i^{k+1}_c} \geq 0$ (because queue i^{k+1}_c is full at the start of $g^{k+1}_{i^{k+1}_c}$). Thus, it holds that $g^{k}_{i^{k+1}_c} \leq g^{pbt}_{i^{k+1}_c,i^{f(k)+1}_{\overline{c}}}$. However, it must hold that $g^{k+1}_{i^{k+1}_c} > g^{pbt}_{i^{k+1}_c,i^{f(k)+1}_{\overline{c}}}$ because otherwise no convergence is possible (see Section 8.2). Hence, $g^{k}_{i^{k+1}_c} < g^{k+1}_{i^{k+1}_c}$. The green periods are related according to D.2. Hence, $g^{k}_{i^{k+1}_c} < g^{k}_{i^{k+1}_c}$ also means $g^{k}_{i_c} < g^{k+1}_{i_c}$, $\forall i_c \in \mathcal{G}_c$.

Hence, we know that $g^{k}_{i_1} < g^{k+1}_{i_1}$, $\forall i_1 \in \mathcal{G}_1$, $\forall k > 0$ and that $g^{k}_{i_2} < g^{k+1}_{i_2}$, $\forall i_2 \in \mathcal{G}_2$, $\forall k > 0$.

Now we know that for an infinite sequence $s^k_c \to s^{f(k)}_{\overline{c}} \to \dots = switch.3a \to switch.3a \to \dots$ the green times keep increasing. We want to show that eventually the green times are too large for *switch*.3*a* to occur. However, when the green times of all signals increase, it can be the case that these green times converge to an asymptote. For example the series $1, 1+\frac{1}{2}, 1+\frac{1}{2}+\frac{1}{4}, 1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}, 1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}, \dots$. is increasing. However, this series has an asymptote at 2.

We want to exclude that the green times converges to an asymptote. To this end we first show that for each set $\mathcal{G}_1$ and $\mathcal{G}_2$, the queue that is active can only change finitely many times, i.e. the queue that reaches its maximum queue length can only change finitely many times. Hereafter, we prove that when the queue that is active does not change anymore for both sets (see Figure D.7) then $\Delta g^{k+1}_{i_c} = g^{k+1}_{i_c} - g^{k}_{i_c}$, $\forall i_c \in \mathcal{G}_c$ increases for increasing $g^{k}_{i_c}$. As a result we cannot converge to a asymptote.

The Buffer That is Active in $\mathcal{G}_c$ Can Only Change a Finite Number of Times

Now we show when the active queue in the set $\mathcal{G}_c$ can change. When, $x^{k}_{c,i_c} = \tilde{x}^{max}_{c,i_c} \wedge x^{k}_{c,j_c} < \tilde{x}^{max}_{c,j_c}$, $i_c \in \mathcal{G}_c, j_c \in \mathcal{G}_c$ and $x^{k+1}_{c,i_c} < \tilde{x}^{max}_{c,i_c} \wedge x^{k+1}_{c,j_c} = \tilde{x}^{max}_{c,j_c}$, $i_c, j_c \in \mathcal{G}_c$ (the queue that is active changes from i_c to j_c), then it must hold that $\Delta x^{k+1}_{c,i_c} < 0$ and $\Delta x^{k+1}_{c,j_c} > 0$. Hence, using $g^{\mu,k}_{i_c} \leq g^{k}_{i_c}$, $g^{\mu,k}_{j_c} = g^{k}_{j_c}$ (because if j_c becomes active when $g^{\mu,k}_{j_c} < g^{k}_{j_c}$ then the signals would switch for the reason *switch*.3*b* and not for the reason *switch*.3*a*), (D.7a), (D.7d) we can find that for an infinite sequence of *switch*.3*a* switch reasons the active buffer can change from i_c to j_c whenever we satisfy the following strict inequalities:

$$(g^{f(k)}_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_c} < g^{k}_{i_c}(\mu_{i_c} - \lambda_{i_c}), \tag{D.10a}$$

$$(g^{f(k)}_{i_{\overline{c}}} + \sigma_{j_c,i_{\overline{c}},j_c})\lambda_{j_c} > g^{k}_{j_c}(\mu_{j_c} - \lambda_{j_c}). \tag{D.10b}$$

Using (D.2) we can see that we can find a value for $g_{i_{\overline{c}}}^{f(k)}$ that satisfies the inequalities in (D.10) if and only if:

$$g_{i_{\overline{c}}}^{f(k)} a_{i_c,j_c} - b_{i_c,j_c} > 0, \tag{D.11}$$

where

$$\begin{aligned} a_{i_c,j_c} &= \frac{\rho_{j_c}}{1-\rho_{j_c}} - \frac{\rho_{i_c}}{1-\rho_{i_c}}, \\ b_{i_c,j_c} &= \frac{\sigma_{i_c,i_{\overline{c}},i_c}}{1-\rho_{i_c}} - \frac{\sigma_{j_c,i_{\overline{c}},j_c}}{1-\rho_{j_2}}. \end{aligned}$$

Thus whenever signal i_c is active then j_c can become active if (and only if) the inequality in (D.11) is satisfied. We can represent which signals could become active using a transition system. This transition system has the states $1, 2, \ldots, N_c$, where N_c is the number of signals in the set $\mathcal{G}_c$. The state represents which of the queues is active. The transitions between these states represent which signals could become active. We could make a transition from state i_c to the state j_c whenever the inequality (D.11) holds. Note that the following holds:

$$a_{i_c,j_c} = -a_{j_c,i_c}, \tag{D.12a}$$

$$b_{i_c,j_c} = -b_{j_c,i_c}. \tag{D.12b}$$

Hence, it holds that whenever $g_{i_{\overline{c}}}^{f(k)} a_{i_c,j_c} - b_{i_c,j_c} > 0$ then $g_{i_{\overline{c}}}^{f(k)} a_{j_c,i_c} - b_{j_c,i_c} < 0$. Thus whenever a transition from i_c to j_c is possible then a transition j_c to i_c is not possible

We distinguish two different transitions.

type 1 we make a transition from the state i_c to the state j_c and $a_{i_c,j_c} \geq 0$.
type 2 we make a transition from the state i_c to the state j_c and $a_{i_c,j_c} < 0$.

First we consider the first type of transitions. Because $g_{i_{\overline{c}}}^{f(k)}$ increases for increasing k and because $a_{j_2,i_2} = -a_{j_2,i_2}$, we can see that whenever we make a type 1 transition from i_c to j_c, from that moment on it holds that $g_{i_{\overline{c}}}^{f(k)} a_{j_c,i_c} - b_{j_c,i_c} < 0$. Hence, we cannot make a direct transition from state j_c to the state i_c anymore.

Lets consider an indirect path from the state j_c to the state i_c via the path $l^1 \to l^2 \to \cdots \to l^n$ where $l^1 = j_2$ and $l^n = i_2$. Thus, from the state j_c we make a transition to the state l^2 and from the state l^2 we make a transition to the state l^3 etcetera. Using (D.12) we can derive:

$$g_{i_{\overline{c}}}^{f(k)} a_{j_c,i_c} - b_{j_c,i_c} = \sum_{m=1}^{m=n} (g_{i_{\overline{c}}}^{f(k)} a_{l^m,l^{m+1}} - b_{l^m,l^{m+1}}) < 0. \tag{D.13}$$

Hence, $\exists m : g_{i_{\overline{c}}}^{f(k)} a_{l^m,l^{m+1}} - b_{l^m,l^{m+1}} < 0$ and thus every path from j_c to i_c contains a transition that is not possible. In conclusion whenever we make a type 1 transition out of the state i_c, we can never reach the state i_c again. In other words, the signal i_c can never become active anymore (when considering the infinite sequence $s_c^k \to s_{\overline{c}}^{f(k)} \to \cdots = switch.3a \to switch.3a \to \ldots$).

A type 2 transition from the state i_c to the state j_c is only possible whenever $b_{i_c,j_c} < 0$, i.e. when $\frac{\sigma_{j_2,i_{\overline{c}},j_c}}{1-\rho_{j_c}} > \frac{\sigma_{i_c,i_{\overline{c}},i_c}}{1-\rho_{i_c}}$. Thus, when there are $N_{reachable}$ reachable states then we can make maximally $N_{reachable} - 1$ type 2 transitions in a row.

Thus, we can make maximally $N_{reachable} - 1$ type 2 transitions in a row (and then the next transition has to be a type 1 transition). Whenever we make a type 1 transition out of a state we can never reach this state again (number of reachable states decreases with at least one). Hence, we cannot make more than $\sum_{i=1}^{N_c} i = \frac{N_c(N_c+1)}{2}$ transitions (and thus the active queue in the set $\mathcal{G}_c$ cannot change more than $\frac{N_c(N_c+1)}{2}$ times), which is finite when we assume N_c is finite.

The Green Times Cannot Converge to a Asymptote

Now we prove that when the queue that is active does not change anymore for both sets (see Figure D.7) then $\Delta g_{i_c}^{k+1} = g_{i_c}^{k+1} - g_{i_c}^{k}, \forall i_c \in \mathcal{G}_c \ k > 0$ increases for increasing $g_{i_c}^{k}$. Lets assume queue $i_c^k \in \mathcal{G}_c$ is the active queue in $\mathcal{G}_c$ and assume that queue $i_{\overline{c}}^{f(k)} \in \mathcal{G}_{\overline{c}}$ is the active queue in $\mathcal{G}_{\overline{c}}$ and

We can distinguish the following sections:

1 between point 1 and point 2, $g_{i_c^k}^{k}$ is performed.

2 between point 2 and point 3,the setup $\sigma_{i_{\overline{c}}^{f(k)}, i_c^k, i_{\overline{c}}^{f(k)}}$ is performed.

3 between point 3 and point 4, $g_{i_{\overline{c}}^{f(k)}}^{f(k)}$ is performed.

4 between point 4 and point 5, $g_{i_c^k}^{k+1}$ is performed.

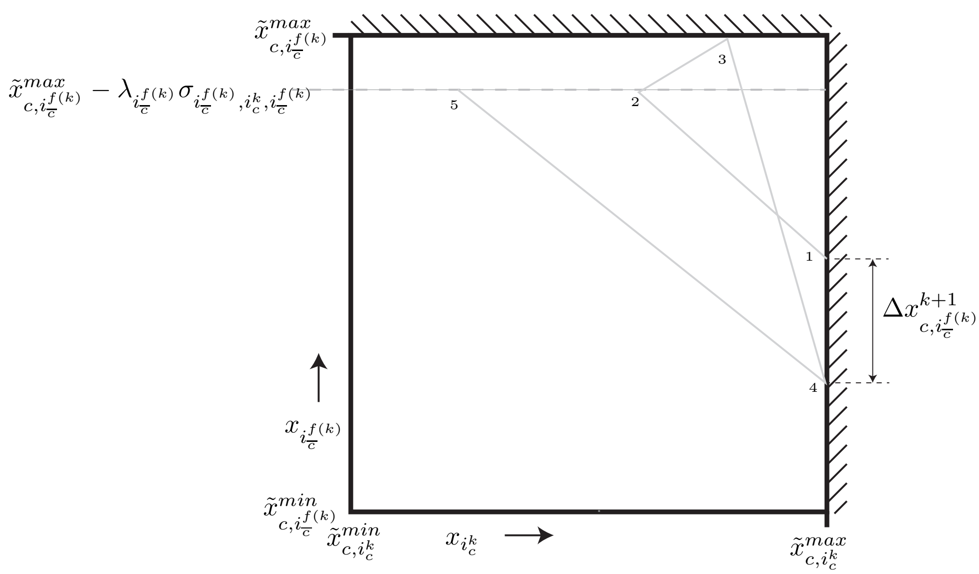


Figure D.7: when the queue that is active does not change anymore for both sets

Using $\Delta x_{c,i_c^k}^{k+1} = 0$ we can derive $\Delta x_{c,i_{\overline{c}}^{f(k)}}^{k+1} = g_{i_c^k}^{k+1} \frac{1-\rho_{i_c^k} - \rho_{i_{\overline{c}}^{f(k)}}}{\rho_{i_c^k}} \mu_{i_{\overline{c}}^{f(k)}} + \sigma_{i_c^k i_{\overline{c}}^{f(k)} i_c^k} \mu_{i_{\overline{c}}^{f(k)}}$. It is easy to see that $\Delta g_{i_c^k}^{k+1} = \frac{\Delta x_{c,i_{\overline{c}}^{f(k)}}^{k+1}}{\lambda_{i_{\overline{c}}^{f(k)}}} = g_{i_c^k}^{k+1} \frac{1-\rho_{i_c^k} - \rho_{i_{\overline{c}}^{f(k)}}}{\rho_{i_c^k} \rho_{i_{\overline{c}}^{f(k)}}} + \frac{\sigma_{i_c^k i_{\overline{c}}^{f(k)} i_c^k}}{\rho_{i_{\overline{c}}^{f(k)}}}$.

The green periods are related according to D.2. As a result the green periods cannot converge to a asymptote.

Eventually $s^k = switch.3a$ is not possible because eventually we satisfy eventually the green period is long enough to empty all queues (with a finite maximum queue capacity) during their green period (and therefore no queue in set $\mathcal{G}_2$ can be active anymore):

$$\forall i_1 \in \mathcal{G}_1 \left(x_{i_1}^{max} = \infty \vee g_{i_1}^k > \frac{x_{i_1}^{max}}{\mu_{i_1} - \lambda_{i_1}} \right).$$

Hence, an infinite sequence $s_c^k \to s_{\overline{c}}^{f(k)} \to \dots = switch.3a \to switch.3a \to \dots$ is not possible.

Thus, $n^{\mathcal{C}_1}$ (which is the smallest number for k such that s_1^k is not element of a combination in the set $\mathcal{C}_1$) is finite. From Lemma D.4 and Lemma D.1 we know that whenever s_c^k is not part of a combination in the set $\mathcal{C}_1$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_1$. Hence, s_c^k, $k \geq n^{\mathcal{C}_1}$ cannot be part of a combination in the set $\mathcal{C}_1$. Thus for $\forall k \geq n^{\mathcal{C}_1}$ two subsequent switch reasons $s_c^k \to s_c^{f(k)}$ cannot be both equal to $switch.3a$.

D.3.2 s_c^k is part of a combination in the set $\mathcal{C}_2$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_2$.

Finite Sequence Using Lemma D.14 we can see that we cannot have an infinite sequence of switch reasons s_c^k where each switch reason is part of a combination in the set $\mathcal{C}_2$.

Restricting Combinations We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_2 = \{C_{14}\}$, then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_1$.

$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set:	Because:
$\mathcal{C}_1$	If $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_1$ then it must hold that $s_{\overline{c}}^{f(k)} = switch.3a$. From Lemma D.4 we know that $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.2 \to switch.3a \to switch.3a$ is not possible.

D.3.3 s_c^k is part of a combination in the set $\mathcal{C}_3$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_3 = \{C_{14}\}$

Finite Sequence From Lemma D.20 we know that whenever $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3a \to switch.3b \to switch.3a \to switch.3b$ then it holds that $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+2} = x_{c,i_{\overline{c}}}^{k+2} - x_{c,i_{\overline{c}}}^{k+1} \geq 0$. Hence, a maximum of 2 subsequent switch reasons can be part of a combination in the set $\mathcal{C}_3 = \{C_{14}\}$.

Restricting Combinations We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_3 = \{C_{14}\}$, i.e. $s_{\overline{c}}^{f(k)-1} \to s_c^k = switch.3a \to switch.3b$ or $s_c^k \to s_{\overline{c}}^{f(k)} = switch.3a \to switch.3b$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_i$, $i < 3$.

$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set:	Because:
$\mathcal{C}_1$	If $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_1$ then it must hold that $s_{\overline{c}}^{f(k)} = switch.3a$. From Lemma D.4 we know that $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.2 \rightarrow switch.3a \rightarrow switch.3a$ is not possible.
$\mathcal{C}_2$	From Lemma D.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_2 = \{C_8\}$.

D.3.4 s_c^k is part of a combination in the set $\mathcal{C}_4$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_4 = \{C_{11}, C_{15}, C_{18}, C_{19}\}$.

Finite Sequence Lets assume that an infinite sequence of switch reasons exists where each switch reason is part of a combination in the set $\mathcal{C}_4$.

From Lemma D.2 we know that combination C_{11} can only occur in this infinite sequence if the combinations C_{18} and C_{19} do not occur in this infinite sequence. Further, from Lemma D.14 we know that an infinite sequence of switch reasons where each switch reason is part of C_{11} is not possible and that an infinite sequence of switch reasons where each switch reason is part of either C_{18} or C_{19} is not possible. Hence, eventually for an infinite sequence of switch reasons that are all part of a combination in the set $\mathcal{G}_4$, the combination C_{15} must occur. From Lemma D.5 we know that whenever combination C_{15} occurs (after another combination in the set $\mathcal{C}_4$) then combination C_{11}, C_{18} and C_{19} can never occur again.

Hence, if an infinite sequence of switch reasons where each switch reason is part of a combination in the set $\mathcal{C}_4$ is possible then an infinite sequence of the combination C_{15} must be possible:

$$s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} \rightarrow \dots = switch.3a \rightarrow switch.3b \rightarrow switch.3a \rightarrow switch.3b \rightarrow \dots$$

From Lemma D.20 we know that for this infinite sequence it holds that $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \Delta x_{c,i_{\overline{c}}}^{h+1} \geq 0 \wedge g_{i_{\overline{c}}}^{\mu,f(h)} = g_{i_{\overline{c}}}^{f(h)} \wedge g_{i_{\overline{c}}}^{f(h)} > g_{i_{\overline{c}}}^{pbt}$ for all $h > k$.

Using Lemma D.19 we can see that each queue $i_c \in \mathcal{G}_c$ either goes empty, i.e. the queue length is zero at the end of $g_{i_c}^h$, $h \geq k$ or its queue length decreases minimally $\Delta_c(g_{i_{\overline{c}}}^h) > 0$. Note that $\Delta_c(g_{i_{\overline{c}}}^h) = \Delta_c(g_{i_{\overline{c}}}^{h+1})$, $h \geq k$ because $g_{i_c}^h = g_{i_c}^{h+1} = \tilde{g}_{i_c}^{max}$, $h \geq k$. As a result, for an infinite sequence of switch reasons that are all part of combination C_{15}, the queues in the set $\mathcal{G}_c$ are eventually emptied (and we switch because of the reason $switch.1a$ or $switch.1b$). Hence, an infinite sequence sequence of switch reasons that are all part of combination C_{15} is not possible.

Restricting Combinations We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_4$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set$\mathcal{C}_i$, $i < 4$.

$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set:	Because:
$\mathcal{C}_1$	If $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_1$ then it must hold that $s_{\overline{c}}^{f(k)} = switch.3a$ is needed. However, from Lemma D.4 we know that $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.3b \rightarrow switch.3a \rightarrow switch.3a$ is not possible.
$\mathcal{C}_2$	From Lemma D.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_2$.
$\mathcal{C}_3$	From Lemma D.20 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_3$.

D.3.5 s_c^k is part of a combination in the set $\mathcal{C}_5$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_5 = \{C_{17}\}$.

Finite Sequence From Lemma D.15 we know that an infinite sequence is not possible.

Restricting Combinations We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_5$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_i$, $i < 5$.

$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set:	Because:
$\mathcal{C}_1$	if $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_1$ then it must hold that $s_{\overline{c}}^{f(k)} = switch.3a$. However, from Lemma D.4 we know that $s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.3b \rightarrow switch.3a \rightarrow switch.3a$ is not possible.
$\mathcal{C}_2$	From Lemma D.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_2$.
$\mathcal{C}_3$	From Lemma D.20 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_3$.
$\mathcal{C}_4$	We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_5$ then $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{11},C_{15},C_{18} and C_{19}. From Lemma D.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{11}. From Lemma D.20 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{15} and from Lemma D.21 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{18} or C_{19}.

D.3.6 s_c^k is part of a combination in the set $\mathcal{C}_6$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_6 = \{C_{10}\}$.

Finite Sequence From Lemma D.15 we know that an infinite sequence of switch reasons, where each switch reason is part of combination C_{10} is not possible.

Restricting Combinations We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_6$ then $s_{\overline{c}}^{f(k)}$ cannot be part of combination $\mathcal{C}_i$, $i < 6$.

$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set:	Because:
$\mathcal{C}_1$	If $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_1$ then it must hold that $s_{\overline{c}}^{f(k)} = switch.3a$. However, from Lemma D.4 we know that $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.3b \rightarrow switch.3a \rightarrow switch.3a$ is not possible.
$\mathcal{C}_2$	From Lemma D.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_2$.
$\mathcal{C}_3$	From Lemma D.20 we know that$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_3$.
$\mathcal{C}_4$	We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_6$ then $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{11},C_{15},C_{18} and C_{19}. From Lemma D.15 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{11}. From Lemma D.9 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{18} and combination C_{15} because and from Lemma D.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{18} and combination C_{19}.
$\mathcal{C}_5$	From Lemma D.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_5$.

D.3.7 s_c^k is part of a combination in the set $\mathcal{C}_7$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_7 = \{C_7\}$.

Finite Sequence From Lemma D.15 we know that an infinite sequence of switch reasons where each switch reason is part of combination C_7 is not possible.

Restricting Combinations We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_7$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_i$, $i < 8$.

$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set:	Because:
$\mathcal{C}_1$	If $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_1$ then it must hold that $s_{\overline{c}}^{f(k)} = switch.3a$. However, from Lemma D.4 we know that $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.2 \rightarrow switch.3a \rightarrow switch.3a$ is not possible.
$\mathcal{C}_2$	From Lemma D.15 and Lemma D.21 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_2$.
$\mathcal{C}_3$, $\mathcal{C}_4$, $\mathcal{C}_5$, $\mathcal{C}_6$	From LemmaD.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in these set.

D.3.8 s_c^k is part of a combination in the set $\mathcal{C}_8$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_8 = \{C_9\}$.

Finite Sequence Lets assume that an infinite sequence of switch reasons where each switch reason is part of a combination in the set $\mathcal{C}_8 = \{C_9\}$ exists:

$$s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} \rightarrow \dots = switch.2 \rightarrow switch.3a \rightarrow switch.2 \rightarrow switch.3a \rightarrow \dots, \quad \text{(D.14)}$$

where $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} x_{c,i_{\overline{c}}}^{h+1} \leq 0$, $\forall h \geq k$.

Since, we stop $g_{i_{\overline{c}}}^h$ $h \geq k$ for the reason $switch.3a$, the signal i_c that reaches its maximum queue (at the start of $g_{i_c}^{h+1}$, $h \geq k$) satisfies $g_{i_c}^{\mu,h} = g_{i_c}^h$, $\Delta x_{c,i_c}^{h+1} \geq 0$ (because $\Delta x_{c,i_c}^{k+1} < 0$ would imply a queue length that exceeds the maximum queue length) and $g_{i_c}^k > g_{i_c}^{pbt}$ (because of inequalities (8.1h), (8.1k)).

Hence, using Lemma D.18 we can see that each queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ either goes empty, i.e. the queue length is zero at the end of $g_{i_{\overline{c}}}^{f(h)}$, $h \geq k$ or its queue length decreases minimally $\Delta_{\overline{c}}(g_{i_c}^h) > 0$. Note that $g_{i_c}^h = g_{i_c}^{h+1} = g_{i_c}^{max}$, $\forall h \geq k$. Hence, eventually all queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ are emptied (and we do not stop serving the signals in the set $\mathcal{G}_{\overline{c}}$ for the reason $switch.3a$ but for the reason $switch.1a$ or $switch.1b$). Thus, an infinite sequence where each switch reason is part of combination C_9 is not possible.

Restricting Combinations We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_8$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_i$, $i < 8$.

$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set:	Because:
$\mathcal{C}_1$	If $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_1$ then it must hold that $s_{\overline{c}}^{f(k)} = switch.3a$. However, from Lemma D.4 we know that $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.2 \rightarrow switch.3a \rightarrow switch.3a$ is not possible.
$\mathcal{C}_2$	From Lemma D.5 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_2$.
$\mathcal{C}_3$	From Lemma D.2 we know that the sequence $s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} = switch.2 \rightarrow switch.3a \rightarrow switch.3b$ and the sequence $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.2 \rightarrow switch.3a \rightarrow switch.3b$ are not possible. From Lemma D.4 we know that the sequence $s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.2 \rightarrow switch.3a \rightarrow switch.3a \rightarrow switch.3b$ is not possible.
$\mathcal{C}_4$	We can prove that when s_c^k is part of combination C_{10} then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in $\mathcal{C}_4 = \{C_{11}, C_{15}, C_{18}, C_{19}\}$. From Lemma D.5 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination C_{11}. Further, $s_{\overline{c}}^{f(k)}$ cannot be part of combination C_{15} for the same reason why $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_3$. From Lemma D.2 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination C_{18} or C_{19}.
$\mathcal{C}_5$	From Lemma D.2 we know that$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_5$.
$\mathcal{C}_6$, $\mathcal{C}_7$	From Lemma D.5 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_6$ or a combination in the set $\mathcal{C}_7$.

D.3.9 s_c^k is part of a combination in the set $\mathcal{C}_9$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_9 = \{C_5\}$.

Finite Sequence From Lemma D.16 we know that an infinite sequence where each switch reason is part of a combination in the set $\mathcal{C}_9$ is not possible.

Restricting Combinations We can prove that when s_c^k is part of a combination in the set $\mathcal{C}_9$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_i$, $i < 9$.

$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set:	Because:
$\mathcal{C}_1$	If $s_{\overline{c}}^{f(k)}$ is part of a combination in the set $\mathcal{C}_1$ then it must hold that $s_{\overline{c}}^{f(k)} = switch.3a$. However, from Lemma D.4 we know that $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.3b \rightarrow switch.3a \rightarrow switch.3a$ is not possible.
$\mathcal{C}_2$	From Lemma D.7 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_2$.
$\mathcal{C}_3$	From Lemma D.1 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_2$.
$\mathcal{C}_4$	We can prove that when s_c^k is part of combination C_{10} then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in $\mathcal{C}_4 = \{C_{11}, C_{15}, C_{18}, C_{19}\}$. From Lemma D.7 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination C_{11}. From Lemma D.1 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination C_{15} and from Lemma D.7 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination C_{18} or C_{19}.
$\mathcal{C}_5$	From Lemma D.7 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_5$.
$\mathcal{C}_6$	From Lemma D.2 and Lemma D.10 we know that$s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $C_3 \in \mathcal{C}_6$. Further, from Lemma D.7 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $C_{10} \in \mathcal{C}_6$.
$\mathcal{C}_7$	From Lemma D.7 we know that $s_{\overline{c}}^{f(k)}$ cannot be part of a combination in the set $\mathcal{C}_7$.

D.3.10 s_c^k is part of a combination in the set $\mathcal{C}_{10}$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_{10} = \{C_6\}$.

Finite Sequence From Lemma D.16 we know that an infinite sequence of switch reasons, where each switch reason is part of combination C_6 is not possible.

Restricting Combinations From Lemma D.7 we know that when s_c^k is part of a combination in the set $\mathcal{C}_{10}$ then $s_{\overline{c}}^{f(k)}$ cannot be part of a combination $\mathcal{C}_i$, $i < 10$ whenever $k \geq n^{\mathcal{C}_1}$.

D.3.11 s_c^k is part of a combination in the set $\mathcal{C}_{11}$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_{11} = \{C_1\}$.

Assume an infinite sequence where each switch reason is part of a combination in the set $\mathcal{C}_{11}$:

$$s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} \rightarrow \cdots = switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow \ldots$$

From Lemma D.22 we know that:

$$
\begin{aligned}
g_{i_{\overline{c}}}^{f(h)} &> g_{i_{\overline{c}}}^{pbt}, \quad \forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}, \quad \forall h \geq k, \\
g_{i_c}^{h} &> g_{i_{\overline{c}}}^{pbt}, \quad \forall i_c \in \mathcal{G}_c, \quad \forall h > k.
\end{aligned}
$$

Further, we know from Lemma D.11 that:

$$
\begin{aligned}
\Delta g_{i_{\overline{c}}}^{f(h)+1} = g_{i_{\overline{c}}}^{f(h)+1} - g_{i_{\overline{c}}}^{f(h)} &\leq \max_{l_c \in \mathcal{G}_c, l_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \frac{\rho_{l_c}(1-\rho_{l_c}-\rho_{l_{\overline{c}}})}{1-\rho_{l_{\overline{c}}}} (g_{l_{\overline{c}},l_c}^{pbt} - g_{l_{\overline{c}}}^{f(h)}) < 0, \quad \forall h \geq k, \\
\Delta g_{i_c}^{h+1} = g_{i_c}^{h+1} - g_{i_{\overline{c}}}^{h} &\leq \max_{l_c \in \mathcal{G}_c, l_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \frac{\rho_{l_{\overline{c}}}(1-\rho_{l_c}-\rho_{l_{\overline{c}}})}{1-\rho_{l_c}} (g_{l_c,l_{\overline{c}}}^{pbt} - g_{l_c}^{h}) \quad < 0, \quad \forall h > k.
\end{aligned}
$$

From these two lemmas we can see that for $k \rightarrow \infty$ the green times (note that the green times of the signals in the set $\mathcal{G}_1$ are related to each other and that the green times of the signals in the set $\mathcal{G}_2$ are related to each other) converge to the smallest green times that satisfy $g_{i_1}^{k} \geq g_{i_1}^{pbt}$, $\forall i_1 \in \mathcal{G}_1$ and satisfy $g_{i_2}^{k} \geq g_{i_2}^{pbt}$, $\forall i_2 \in \mathcal{G}_2$. Note that for $k \rightarrow \infty$, the green times converge to the smallest green and red times that satisfy the inequalities in (7.14a). Hence, the green times converge to green times that are smaller than (or equal to) the green times of the trajectory that we want to follow (for an infinite sequence of switch reasons equal to $switch.1a$).

When the green time $g_{i_c}^{k}$ is smaller than the green times for the desired trajectory then we switch for the reason $switch.1b$ (see section D.1). If the green times that we converge to are equal to the green times of the desired trajectory we converge to the desired trajectory.

Note that we only have an infinite sequence where each switch reason is part of a combination in the set $\mathcal{C}_{11}$ whenever the green times of the desired trajectory are the green times that we converge to.

Restricting Combinations When s_c^k is part of combination $\mathcal{C}_{11}$ then $s_{\overline{c}}^{f(k)}$ cannot be part of combination $\mathcal{C}_i$, $i < 11$ because of Lemma D.12 and Lemma D.1.

D.3.12 s_c^k is part of a combination in the set $\mathcal{C}_{12}$

In this section we consider the case where s_c^k is part of a combination in the set $\mathcal{C}_{12}$.

From Lemma D.23 we know that whenever $s_c^k = switch.1b$ then we follow the desired trajectory from the start of the $k+1$th cycle c. Whenever $s_c^k = switch.1b$ it holds that:

$$
s_c^k = switch.1b, \quad \forall h \geq k, \tag{D.15}
$$

$$
s_{\overline{c}}^{f(h)} = switch.1b, \quad \forall h \geq k. \tag{D.16}
$$

D.4 Lemmas

In this section we show the different lemmas that we use in the proof of the policy.

D.4.1 Lemmas Excluding Sequences of Switch Reasons

Lemma D.1 *The sequence $s_c^k \rightarrow s_{\overline{c}}^{f(k)} = switch.1a \rightarrow switch.3a$ is not possible.*

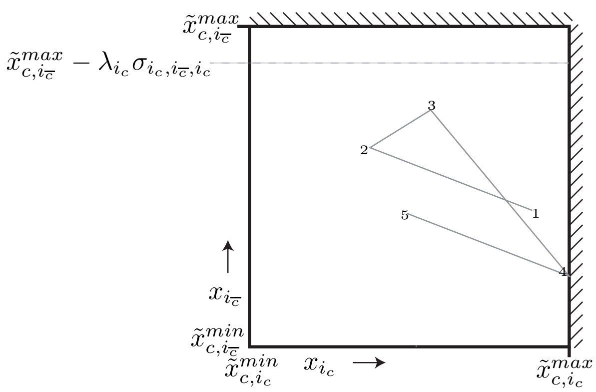
Proof. Whenever $s_{\overline{c}}^{f(k)} = switch.3a$ then there is a queue $i_c \in \mathcal{G}_c$ that was not emptied during $g_{i_c}^k$ (see Section D.1 for the definition of the switch reason $switch.3a$). However, when $s_c^k = switch.1a$ this means that all signals in the set $\mathcal{G}_c$ are emptied during their green time $g_{i_c}^k$. Hence, the sequence $s_c^k \rightarrow s_{\overline{c}}^{f(k)} = switch.1a \rightarrow switch.3a$ is not possible. ■

Lemma D.2 *$s_c^k = switch.3b$ can occur iff $s_c^h \neq switch.2$, $\forall h \geq 1$.*

Proof. From Lemma D.17 we know that whenever a queue in the set $\mathcal{G}_{\overline{c}}$ goes from empty to the maximum queue length then all queues in the set $\mathcal{S}_{\overline{c}}$ will go from empty to the maximum queue length. When $s_c^k = switch.3b$ then this means that a queue goes from empty to the maximum queue length before the maximum green time is reached, i.e. $\frac{x_{i_s^{max}}}{\lambda_{i_s}} - \sigma_{i_c,i_s,i_c} < g_{i_c}^{max}$, $i_c \in \mathcal{G}_c$, $i_s \in \mathcal{S}_{\overline{c}}$ (see the definitions of $switch.2$ and $switch.3b$ in Section D.1). Moreover, when $s_c^k = switch.2$ this means that a queue cannot go from empty to the maximum queue length before the maximum green time is reached, i.e. $\frac{x_{i_s^{max}}}{\lambda_{i_s}} - \sigma_{i_c,i_s,i_c} \geq g_{i_c}^{max}$, $i_c \in \mathcal{G}_c$, $i_s \in \mathcal{S}_{\overline{c}}$. Hence $s_c^k = switch.3b$ can occur iff $s_c^h \neq switch.2$, $\forall h \geq 1$. ■

Lemma D.3 *The sequences $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.3b \rightarrow switch.3a \rightarrow switch.1a$ and $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.2 \rightarrow switch.3a \rightarrow switch.1a$ are not possible.*

Proof. The visualization of this proof can be seen in Figure D.8a. In this Figure $i_c \in \mathcal{G}_c$ is the signal that causes $s_{\overline{c}}^{f(k)} = switch.3a$, i.e. the queue that is full at the beginning of $g_{i_c}^{k+1}$. The signal $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ could refer to any signal in the set $\mathcal{G}_{\overline{c}}$.



(a) Visualization of Lemma D.3

In this figure we can see the following sections:

1 Between point 1 and point 2, $g_{i_c}^k = \tilde{g}_{i_{\overline{c}}}^{max}$ is performed.
2 between point 2 and point 3, the setup $\sigma_{i_{\overline{c}},i_c,i_{\overline{c}}}$ is performed.
3 between point 3 and point 4, $g_{i_{\overline{c}}}^{f(k)}$ is performed.

4 Between point 4 and point 5, $g_{i_c}^{k+1}$ is performed.

We were not able empty queue i_c during $g_{i_c}^k = \tilde{g}_{i_{\overline{c}}}^{max}$ (since $s_{\overline{c}}^{f(k)} = switch.3a$) and the queue length at the beginning of $g_{i_c}^{k+1}$ cannot be less than the queue length at the beginning of $g_{i_c}^k$ (queue i_c is full at the beginning of $g_{i_c}^{k+1}$). Hence, we cannot empty queue i_c during $g_{i_c}^{k+1} \leq \tilde{g}_{i_{\overline{c}}}^{max}$. Thus, the sequences $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3b \to switch.3a \to switch.1a$ and $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.2 \to switch.3a \to switch.1a$ are not possible. ■

Lemma D.4 *$s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.2 \to switch.3a \to switch.3a$ and $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.3b \to switch.3a \to switch.3a$ are not possible*

Proof. We can see the case where $s_c^k \to s_{\overline{c}}^{f(k)} = switch.2 \to switch.3a$ or $s_c^k \to s_{\overline{c}}^{f(k)} = switch.3b \to switch.3a$ in Figure D.8. We use i_c^{k+1} for a signal that has a queue length that is equal to the maximum queue length at the beginning of $g_{i_c}^{k+1}$ (which exists because $s_{\overline{c}}^{f(k)} = switch.3a$). We use $i_{\overline{c}}$ for a signal (could be any sign) in the set $\mathcal{G}_{\overline{c}}$.

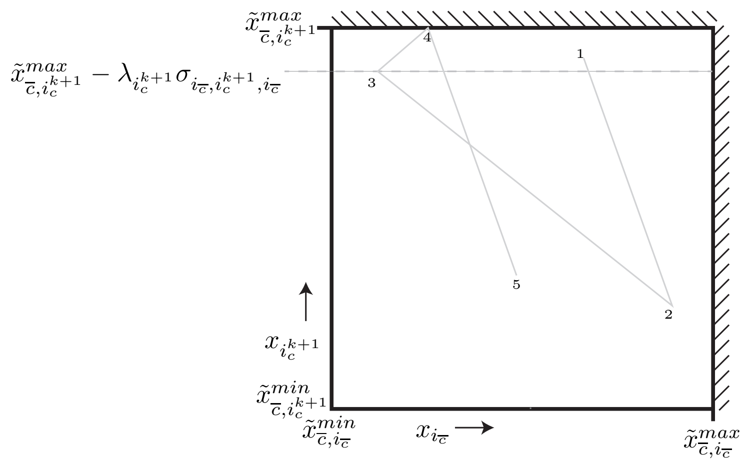


Figure D.8: Visualization of Lemma D.4

We can distinguish the following sections in this figure:

1 between point 1 and point 2, $g_{i_c^{k+1}}^k$ is performed.

2 between point 2 and point 3, $g_{i_{\overline{c}}}^{f(k)}$ is performed.

3 between point 3 and point 4, the setup $\sigma_{i_{\overline{c}},i_c^{k+1},i_{\overline{c}}}$ is performed.

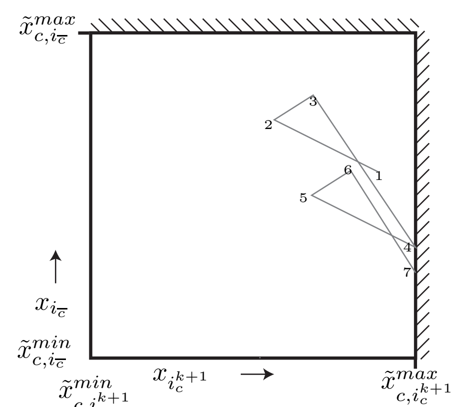
4 between point 4 and point 5, $g_{i_c^{k+1}}^{k+1}$ is performed.

Because $\Delta x_{c,i_c^{k+1}}^{k+1} \geq 0$ (because queue i_c^{k+1} is full at the beginning of $g_{i_c^{k+1}}^{k+1}$), $g_{i_c^{k+1}}^{\mu,k} = g_{i_c^{k+1}}^k$ (because buffer i_c^{k+1} was not emptied during $g_{i_c^{k+1}}^k$ since $s_{\overline{c}}^k = switch.3a$) and $g_{i_c^{k+1}}^k > g_{i_c^{k+1}}^{pbt}$ (because of the inequalities in (8.1h), (8.1k), (8.1n) and (8.1q)) we can use Lemma D.18. From Lemma D.18 it follows that each queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ either empties, i.e. the queue length is zero at the end of its green time $g_{i_{\overline{c}}}^{f(k)}$, or its queue length decreases minimally $\Delta_{\overline{c}}(g_{i_c}^k) > 0$ during the kth cycle c.

As a result, for each signal $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ it holds that the queue length at the start of $g^{k+1}_{i^{k+1}_c}$ is not larger than the queue length at the start of $g^{k}_{i^{k+1}_c}$. Further, because it holds that $g^{k+1}_{i^{k+1}_c} \leq \tilde{g}^{max}_{i_c}$ (because of inequality (D.5a)) and $g^{k}_{i^{k+1}_c} = \tilde{g}^{max}_{i_c}$ (see the definition of $\tilde{g}^{max}_{i_c}$ in Section D.2) the queue length of queue $i_{\overline{c}}$ cannot be larger at the end of $g^{k+1}_{i^{k+1}_c}$ than it was at the end of $g^{k}_{i^{k+1}_c}$. As a result $s^k_c \to s^{f(k)}_{\overline{c}} \to s^{k+1}_c = switch.2 \to switch.3a \to switch.3a$ and $s^k_c \to s^{f(k)}_{\overline{c}} \to s^{k+1}_c = switch.3b \to switch.3a \to switch.3a$ are not possible. ■

Lemma D.5 *If $s^k_c \to s^{f(k)}_{\overline{c}} = switch.2 \to switch.3a$ or $s^k_c \to s^{f(k)}_{\overline{c}} = switch.3b \to switch.3a$ then $s^{f(k)+1}_{\overline{c}} = switch.3b$ is not possible and $s^{f(k)+1}_{\overline{c}} = switch.2$ is not possible.*

Proof. The proof of this lemma is shown in Figure D.9a



(a) Visualization of Lemma D.5

In this figure we use $i^{k+1}_c \in \mathcal{G}_c$ to refer to a signal in the set $\mathcal{G}_c$ which queue is full at the start of $g^{k+1}_{i_c}$ (which exists because $s^{f(k)}_{\overline{c}} = switch.3a$) and $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ could refer to any signal in the set $\mathcal{G}_{\overline{c}}$. In this figure we can see the following sections:

1 Between point 1 and point 2, $g^k_{i_c} = \tilde{g}^{max}_{i_{\overline{c}}}$ is performed.
2 between point 2 and point 3, the setup $\sigma_{i_{\overline{c}}, i^{k+1}_c, i_{\overline{c}}}$ is performed.
3 between point 3 and point 4, $g^{f(k)}_{i_{\overline{c}}}$ is performed.
4 Between point 4 and point 5, $g^{k+1}_{i^{k+1}_c} = \tilde{g}^{max}_{i_{\overline{c}}}$ is performed.
5 between point 5 and point 6, the setup $\sigma_{i_{\overline{c}}, i^{k+1}_c, i_{\overline{c}}}$ is performed.
6 between point 6 and point 7, $g^{f(k)+1}_{i_{\overline{c}}}$ is performed.

First of all point 1 cannot be positioned on the right side of point 4 (because queue i^{k+1}_c is full at the beginning of $g^{k+1}_{i_c}$ and otherwise the maximum queue length would be exceeded). Further, it holds that $g^k_{i_c} \geq g^{k+1}_{i_c}$ (because $g^k_{i_c} = \tilde{g}^{max}_{i_c}$). As a result point 2 cannot be positioned on the right of point 5. Hence, it follows that $g^{f(k)+1}_{i_{\overline{c}}} \leq g^{f(k)}_{i_{\overline{c}}}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ (because otherwise queue i_c would overflow). It holds that $g^{f(k)}_{i_{\overline{c}}} < \tilde{g}^{max}_{i_{\overline{c}}}$ (since $s^{f(k)}_{\overline{c}} = switch.3a$). Hence, $g^{f(k)+1}_{i_{\overline{c}}} \leq g^{f(k)}_{i_{\overline{c}}} < \tilde{g}^{max}_{i_{\overline{c}}}$. As a result, $s^{f(k)+1}_{\overline{c}} = switch.3b$ is not possible and $s^{f(k)+1}_{\overline{c}} = switch.2$ is not possible. ■

Lemma D.6 *The following sequences are not possible:*

1 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.2 \to switch.1a \to switch.2 \to switch.2$
2 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.2 \to switch.1a \to switch.2 \to switch.3b$
3 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3b \to switch.1a \to switch.3b \to switch.2$
4 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3b \to switch.1a \to switch.3b \to switch.3b$

Proof. Whenever $s_c^k = switch.2$ this means that at the start of $g_{i_{\overline{c}}}^{f(k)}$ the queue length of queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ satisfies $x_{i_{\overline{c}}}(t) \geq (\tilde{g}_{i_c}^{max} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$ because $(\tilde{g}_{i_c}^{max} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$ is the amount of traffic that arrived during the red period. During $g_{i_{\overline{c}}}^{f(k)}$ all queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ are emptied (because $s_{\overline{c}}^{f(k)} = switch.1a$). At the beginning of $g_{i_{\overline{c}}}^{f(k)+1}$ the queue length of a queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ is exactly $x_{i_{\overline{c}}}(t) = (\tilde{g}_{i_c}^{max} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$ (which is less than or equal to the queue level at the start of $g_{i_{\overline{c}}}^{f(k)}$). Hence, we can also empty all queues in the set $\mathcal{G}_{i_{\overline{c}}}$ during $g_{i_{\overline{c}}}^{f(k)+1}$ before we have to stop for the reason $switch.2$ or for the reason $switch.3b$. ∎

Lemma D.7 *The following sequences are not possible if* $k \geq n^{\mathcal{C}_1} + 1$

1 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.2 \to switch.3b$
2 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.3b \to switch.3b$
3 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.2 \to switch.2$
4 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.3b \to switch.2$
5 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.2 \to switch.3a$
6 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.3b \to switch.3a$

Proof. First of all we are going to proof that if the sequence 1,2,3,4,5 or 6 exists then it must hold that $s_{\overline{c}}^{f(k)-1} = switch.2$ if $s_{\overline{c}}^{f(k)} = switch.2$ and it must hold that $s_{\overline{c}}^{f(k)-1} = switch.3b$ if $s_{\overline{c}}^{f(k)} = switch.3b$.

From Lemma D.2 we know that $s_{\overline{c}}^{f(k)-1} = switch.3b$ cannot occur if $s_{\overline{c}}^{f(k)} = switch.2$ and that $s_{\overline{c}}^{f(k)-1} = switch.2$ cannot occur if $s_{\overline{c}}^{f(k)} = switch.3b$.

From Lemma D.12 we know that it is not possible that $s_{\overline{c}}^{f(k)-1} = switch.1a$ and from Lemma D.23 we know that $s_{\overline{c}}^{f(k)-1} = switch.1b$ is not possible. Further, we also look at s_c^{k-1} to proof that $s_{\overline{c}}^{f(k)-1} = switch.3a$ is not possible. In the table below we show why $s_{\overline{c}}^{f(k)-1} = switch.3a$ is not possible.

$s_c^{k-1} \to s_{\overline{c}}^{f(k)-1}$ equal to	not possible when $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1}$ is sequence 1,2,3,4,5 or 6 because
$switch.1a \to switch.3a$	Lemma D.1
$switch.1b \to switch.3a$	Lemma D.23
$switch.2 \to switch.3a$	Lemma D.3
$switch.3a \to switch.3a$	s_c^{k-1} cannot be part of a combination in the set $\mathcal{C}_1$ whenever $k \geq n^{\mathcal{C}_1} + 1$
$switch.3b \to switch.3a$	Lemma D.3

Hence, if a sequence 1,2,3,4,5 or 6 exists then it must hold that $s_{\overline{c}}^{f(k)} = switch.2$ if $s_{\overline{c}}^{f(k)+1} = switch.2$ and that $s_{\overline{c}}^{f(k)} = switch.3b$ if $s_{\overline{c}}^{f(k)+1} = switch.3b$. From Lemma D.6 we can now see that sequences 1,2,3 and 4 are not possible. From Lemma D.8 it follows that sequences 5 and 6 are not possible. ∎

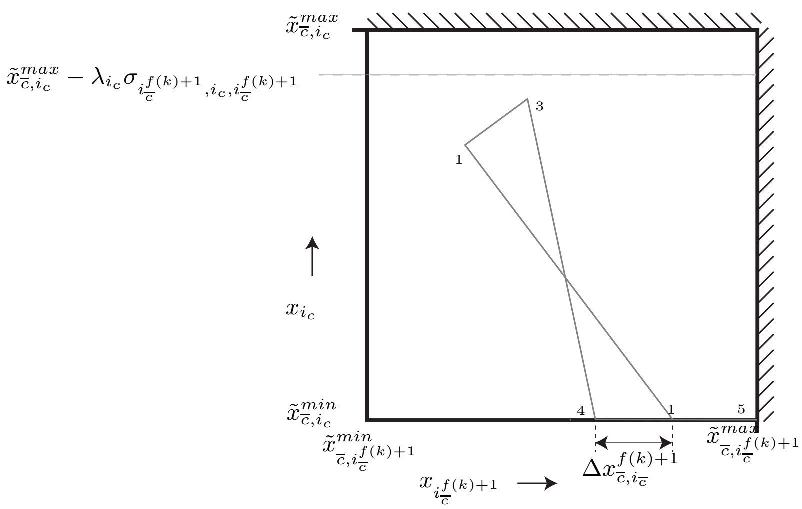
Lemma D.8 *The following sequences are not possible:*

1 $s_c^k s_c^{k-1} \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.2 \to switch.1a \to switch.2 \to switch.3a$
2 $s_c^k s_c^{k-1} \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.3b \to switch.1a \to switch.3b \to switch.3a$

Proof. Assume $s_c^{k+1} = switch.3a$. We use $i_{\overline{c}}^{f(k)+1}$ to refer to the queue that is fulll at the end of $g_{i_c}^{k+1}$ see Figure D.9a. It holds that queue $i_{\overline{c}}^{f(k)+1}$ is not emptied during $g_{i_{\overline{c}}}^{f(k)}$ (because we otherwise $s_c^{k+1} = switch.3b$). We know from inequalities (8.1h), (8.1k), (8.1n) and (8.1q) that $g_{i_{\overline{c}}^{f(k)+1}}^{f(k)} > g_{i_{\overline{c}}^{f(k)+1}}^{pbt}$. In Figure D.9a we can see that any queue i_c is emptied before queue $i_{\overline{c}}^{f(k)+1}$ is full.

Furthermore, using inequalities (8.1j), (8.1m), (8.1p) and (8.1p) and using (D.7a) and (D.7d)) and using $g_{i_{\overline{c}}^{f(k)+1}}^{\mu,f(k)} = g_{i_{\overline{c}}^{f(k)+1}}^{f(k)}$ we can derive that the minimum green times of the signals in $\mathcal{G}_c$ are satisfied before $s_c^{k+1} = switch.3a$ occurs.

Furthermore, we know that we satisfy condition 1.3 (see Section 8.3.1) before $s_c^{k+1} = switch.3a$ occurs (otherwise a maximum queue length is exceeded for the desired trajectory). Hence, we switch for the reason $s_c^{k+1} = switch.1a$ or $s_c^{k+1} = switch.1b$ before we have to switch for the reason $s_c^{k+1} = switch.3a$.



(a) Situation 1

■

Lemma D.9 *The sequence* $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.2 \to switch3b \to switch.3a$ *where* $\max_{i_{\overline{c}} \in \mathcal{G}_c} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$ *is not possible.*

Proof. Because of switch reason $s_c^k = switch.2$ and because of (8.1f) and (8.1g) it holds that:

$$g_{i_c}^k = g_{i_c}^{max}, \quad \forall i_c \in \mathcal{G}_c.$$

When $\max_{i_{\overline{c}} \in \mathcal{G}_c} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$, the queue length of every signal $i_c \in \mathcal{G}_c$ cannot be greater at the start of $g_{i_c}^{k+1}$ than it was at the start of $g_{i_c}^k$. When we performed the maximum green time $g_{i_c}^k = g_{i_c}^{max}$, no maximum queue lengths where exceeded. Hence, when performing a green time $g_{i_c}^{k+1} \leq g_{i_c}^{max}$ again no queue lengths would be exceeded and therefore s_c^{k+1} cannot be equal to $switch.3a$ (we reach the maximum green time before we have to switch for the reason $s_c^{k+1} = switch.3a$). ∎

Lemma D.10 *The following sequences are not possible:*

sequence 1 $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} = switch.3b \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$
sequence 2 $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} = switch.3b \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.2$
sequence 3 $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} = switch.2 \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$
sequence 4 $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} = switch.2 \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.2$

Proof. Lets assume sequence 1,2,3 or 4 are possible. In these cases it holds that $g_{i_c}^k = \tilde{g}_{i_c}^{max}$, $\forall i_c \in \mathcal{G}_c$. Hence, the queue length of queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ is at least $(\tilde{g}_{i_c}^{max} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}} = \tilde{r}_{i_{\overline{c}}}^{max}\lambda_{i_{\overline{c}}}$ at the beginning of $g_{i_{\overline{c}}}^{f(k)}$. The queue length at the beginning of $g_{i_{\overline{c}}}^{f(k)+1}$ is at most $(\tilde{g}_{i_{\overline{c}}}^{max} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$ (because all queues in the set $\mathcal{G}_{\overline{c}}$ where empty at the end of $g_{i_{\overline{c}}}^{f(k)}$ and $g_{i_c}^{k+1} \leq \tilde{g}_{i_c}^{max}$).

Because we could empty all queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ during $g_{i_{\overline{c}}}^{f(k)} \leq \tilde{g}_{i_{\overline{c}}}^{max}$ we are also able to empty all queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ during $g_{i_{\overline{c}}}^{f(k)+1} \leq g_{i_{\overline{c}}}^{f(k)} \leq \tilde{g}_{i_{\overline{c}}}^{max}$. However, $s_{\overline{c}}^{f(k)+1} = switch.2$ or $s_{\overline{c}}^{f(k)+1} = switch.3b$ can only occur if we are not able to empty all queues during $g_{i_{\overline{c}}}^{f(k)+1} \leq \tilde{g}_{i_{\overline{c}}}^{max}$ (see Section D.1 for the definitions of the switch reasons $switch.2$ and $switch.3b$). ∎

Lemma D.11 *Whenever,* $s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a$ *then it holds that* $\Delta g_{i_c}^{k+1} = g_{i_c}^{k+1} - g_{i_c}^k \leq \max_{l_c \in \mathcal{G}_c, l_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \frac{\rho_{l_{\overline{c}}}(1-\rho_{l_c}-\rho_{l_{\overline{c}}})}{1-\rho_{l_c}}(g_{l_{\overline{c}},l_c}^{pbt} - g_{l_{\overline{c}}}^{f(k)}) < 0$, $\forall i_c \in \mathcal{G}_c$

The sequence $s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} = switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$ *and the sequence* $s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} = switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.2$ *are not possible.*

Proof. In Figure D.9a we can see the situation where $s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a$. We use $i_{\overline{c}^*} \in \mathcal{G}_{\overline{c}}$ for the signal that satisfies $g_{i_{\overline{c}}^*}^{\mu,f(k)} = g_{i_{\overline{c}}^*}^{f(k)}$ and $i_c^* \in \mathcal{G}_c$ for the signal that satisfies $g_{i_c^*}^{\mu,k+1} = g_{i_c^*}^{k+1}$ (From Lemma D.22 we know these signals exists).

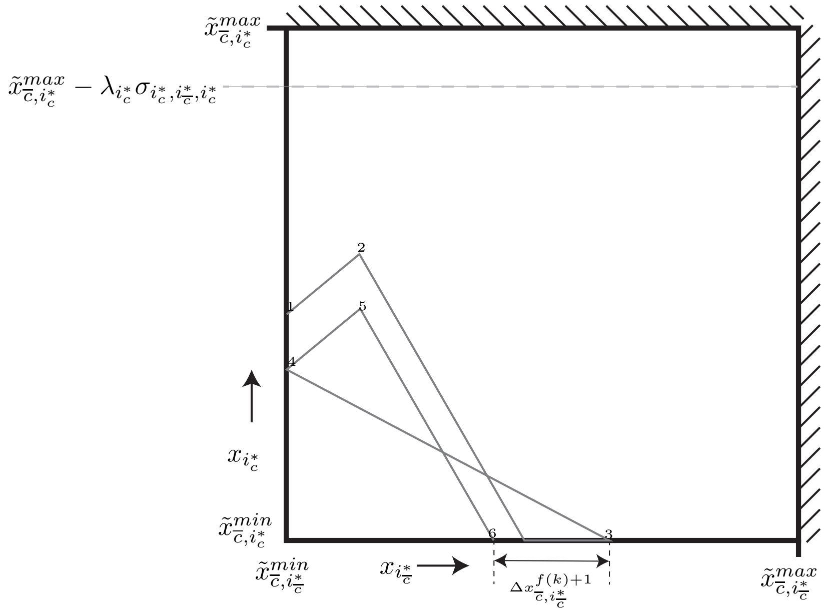
In this figure we can see the following sections:

1 Between point 1 and point 2, the setup $\sigma_{i_{\overline{c}}^*,i_c^*,i_{\overline{c}}^*}$.
2 between point 2 and point 3, $g_{i_c^*}^k$ is performed.
3 between point 3 and point 4, $g_{i_{\overline{c}}^*}^{f(k)}$ is performed.
4 Between point 4 and point 5, the setup $\sigma_{i_{\overline{c}}^*,i_c^*,i_{\overline{c}}^*}$.
5 between point 5 and point 6, $g_{i_c^*}^{k+1}$ is performed.

Using $g_{i_{\overline{c}}^*}^{\mu,f(k)} = g_{i_{\overline{c}}^*}^{f(k)}$ and $g_{i_c^*}^{\mu,k+1} = g_{i_c^*}^{k+1}$ and (D.7) we can find that:

$$\Delta x_{\overline{c},i_{\overline{c}}^*}^{f(k)+1} = \frac{\mu_{i_{\overline{c}}^*}(1-\rho_{i_c^*}-\rho_{i_{\overline{c}}^*})}{1-\rho_{i_c^*}}(g_{i_{\overline{c}}^*,i_c^*}^{pbt} - g_{i_{\overline{c}}^*}^{f(k)}).$$

From lemma D.22 we know that:



(a) Visualization of Lemma D.11

$$g^{f(k)}_{i_{\overline{c}}} > g^{pbt}_{i_{\overline{c}}}, \quad \forall i_c \in \mathcal{G}_c.$$

From (D.4) we know:

$$g^{pbt}_{i_{\overline{c}}} \geq g^{pbt}_{i_{\overline{c}},i_c}, \quad \forall i_c \in \mathcal{G}_c, \forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}.$$

Hence, it holds that:

$$\Delta x^{f(k)+1}_{\overline{c},i^*_{\overline{c}}} = \frac{\mu_{i^*_{\overline{c}}}(1-\rho_{i^*_c}-\rho_{i^*_{\overline{c}}})}{1-\rho_{i^*_c}}(g^{pbt}_{i^*_{\overline{c}},i^*_c} - g^{f(k)}_{i^*_{\overline{c}}}) < 0.$$

Note that $\Delta g^{k+1}_{i^*_c} = g^{k+1}_{i^*_c} - g^{k}_{i^*_c} = \frac{\Delta x^{f(k)+1}_{\overline{c},i^*_{\overline{c}}}}{\lambda_{i^*_{\overline{c}}}}$ and thus:

$$\Delta g^{k+1}_{i^*_c} = \frac{\rho_{i^*_{\overline{c}}}(1-\rho_{i^*_c}-\rho_{i^*_{\overline{c}}})}{1-\rho_{i^*_c}}(g^{pbt}_{i^*_{\overline{c}},i^*_c} - g^{f(k)}_{i^*_{\overline{c}}}) < 0.$$

The green times are related via (D.2). Therefore, it also holds that:

$$\Delta g^{k+1}_{i_c} = g^{k+1}_{i_c} - g^{k}_{i_c} = \frac{\rho_{i^*_{\overline{c}}}(1-\rho_{i^*_c}-\rho_{i^*_{\overline{c}}})}{1-\rho_{i^*_c}}(g^{pbt}_{i^*_{\overline{c}},i^*_c} - g^{f(k)}_{i^*_{\overline{c}}}) < 0, \quad \forall i_c \in \mathcal{G}_c.$$

As a result we can find that:

$$\Delta g^{k+1}_{i_c} = g^{k+1}_{i_c} - g^{k}_{i_c} = \frac{\rho_{i^*_{\overline{c}}}(1-\rho_{i^*_c}-\rho_{i^*_{\overline{c}}})}{1-\rho_{i^*_c}}(g^{pbt}_{i^*_{\overline{c}},i^*_c} - g^{f(k)}_{i^*_{\overline{c}}}) \leq \max_{l_c \in \mathcal{G}_c, l_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \frac{\rho_{l_{\overline{c}}}(1-\rho_{l_c}-\rho_{l_{\overline{c}}})}{1-\rho_{l_c}}(g^{pbt}_{l_{\overline{c}},l_c} - g^{f(k)}_{l_{\overline{c}}}) < 0, \quad \forall i_c \in \mathcal{G}_c. \tag{D.17}$$

$s_{\overline{c}}^{f(k)-2} \to s_c^{k-1} \to s_{\overline{c}}^{f(k)-1} \to s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1}$	is impossible because
$* \to switch.1a \to switch.1a \to switch.1a \to switch.1a \to switch.2$	Lemma D.11
$* \to switch.2 \to switch.1a \to switch.1a \to switch.1a \to switch.2$	Lemma D.13
$switch.1a \to switch.3a \to switch.1a \to switch.1a \to switch.1a \to switch.2$	Lemma D.1
$switch.2 \to switch.3a \to switch.1a \to switch.1a \to switch.1a \to switch.2$	Lemma D.3
$switch.3a \to switch.3a \to switch.1a \to switch.1a \to switch.1a \to switch.2$	s_c^k cannot be part of a combination in the set $\mathcal{C}_1$ whenever $k \geq n^{\mathcal{C}_1} + 1$
$switch.3b \to switch.3a \to switch.1a \to switch.1a \to switch.1a \to switch.2$	Lemma D.3
$* \to switch.3b \to switch.1a \to switch.1a \to switch.1a \to switch.2$	Lemma D.13
$* \to * \to switch.2 \to switch.1a \to switch.1a \to switch.2$	Lemma D.10
$* \to switch.1a \to switch.3a \to switch.1a \to switch.1a \to switch.2$	Lemma D.1
$* \to switch.2 \to switch.3a \to switch.1a \to switch.1a \to switch.2$	Lemma D.3
$* \to switch.3a \to switch.3a \to switch.1a \to switch.1a \to switch.2$	s_c^k cannot be part of a combination in the set $\mathcal{C}_1$ whenever $k \geq n^{\mathcal{C}_1} + 1$
$* \to switch.3b \to switch.3a \to switch.1a \to switch.1a \to switch.2$	Lemma D.3
$* \to * \to switch.3b \to switch.1a \to switch.1a \to switch.2$	Lemma D.10

Table D.4: In this table we show that all sequences $s_{\overline{c}}^{f(k)-2} \to s_c^{k-1} \to s_{\overline{c}}^{f(k)-1} \to s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1}$, where $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.1a \to switch.2$ are not possible. We do not consider the sequences where $s_{\overline{c}}^{f(k)-2}$, s_c^{k-1} or $s_{\overline{c}}^{f(k)-1}$ is equal to $switch.1b$. * means that this switch reason can be either one of the five switch reasons (except for $switch.1b$)

The queue length of queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ equals $(g_{i_c}^k + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$ at the beginning of $g_{i_{\overline{c}}}^{f(k)}$ and equals $(g_{i_c}^{k+1} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}} < (g_{i_c}^k + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$ at the beginning of $g_{i_{\overline{c}}}^{f(k)+1}$. For all signals $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ an amount of $(g_{i_c}^k + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$ traffic could depart during $g_{i_{\overline{c}}}^{f(k)} \leq \tilde{g}_{i_{\overline{c}}}^{max}$. Hence, an amount of $(g_{i_c}^{k+1} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}} < (g_{i_c}^k + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$ could depart during $g_{i_{\overline{c}}}^{f(k)+1} < \tilde{g}_{i_{\overline{c}}}^{max}$.

Thus, the sequence $s_{\overline{c}}^{f(k)-1} \to s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.1a \to switch.1a \to switch.1a \to switch.1a \to switch.3b$ and the sequence $s_{\overline{c}}^{f(k)-1} \to s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.1a \to switch.1a \to switch.1a \to switch.1a \to switch.2$ are not possible because during $g_{i_{\overline{c}}}^{f(k)+1}$ we switch because of the reason $switch.1a$ or $switch.1b$ before we have to switch because of the reason $switch.2$ or $switch.3b$. ■

Lemma D.12 *The sequences $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.1a \to switch.2$ and $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.1a \to switch.3b$ are not possible for $k \geq n^{\mathcal{C}_1} + 1$ (in Section D.3 we explain the definition of $n^{\mathcal{C}_1} + 1$)*

Proof. We first prove that the sequence $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.1a \to switch.2$ is not possible by also looking at $s_{\overline{c}}^{f(k)-2}$, s_c^{k-1} and $s_{\overline{c}}^{f(k)-1}$. We prove that all sequences $s_{\overline{c}}^{f(k)-2} \to s_c^{k-1} \to s_{\overline{c}}^{f(k)-1} \to s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1}$, where $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.1a \to switch.2$ are not possible if $k \geq n^{\mathcal{C}_1} + 1$.

First of all, whenever $s_{\overline{c}}^{f(k)-2}$, s_c^{k-1} or $s_{\overline{c}}^{f(k)-1}$ is equal to $switch.1b$ then $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} = switch.1a \to switch.1a \to switch.2$ is not possible because of Lemma D.23. All other possible sequences are shown in the Table D.4.

$s_{\overline{c}}^{f(k)-2} \rightarrow s_c^{k-1} \rightarrow s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1}$	is not possible because
$* \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.11
$* \rightarrow switch.2 \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.13
$switch.1a \rightarrow switch.3a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.1
$switch.2 \rightarrow switch.3a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.3
$switch.3a \rightarrow switch.3a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	s_c^k cannot be part of a combination in the set $\mathcal{C}_1$ whenever $k \geq n^{\mathcal{C}_1} + 1$
$switch.3b \rightarrow switch.3a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.3
$* \rightarrow switch.3b \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.13
$* \rightarrow * \rightarrow switch.2 \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.10
$* \rightarrow switch.1a \rightarrow switch.3a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.1
$* \rightarrow switch.2 \rightarrow switch.3a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.3
$* \rightarrow switch.3a \rightarrow switch.3a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	s_c^k cannot be part of a combination in the set $\mathcal{C}_1$ whenever $k \geq n^{\mathcal{C}_1} + 1$
$* \rightarrow switch.3b \rightarrow switch.3a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.3
$* \rightarrow * \rightarrow switch.3b \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$	Lemma D.10

Table D.5: In this table we show that all sequences $s_{\overline{c}}^{f(k)-2} \rightarrow s_c^{k-1} \rightarrow s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1}$, where $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.1a \rightarrow switch.1a \rightarrow switch.3b$ are not possible. We do not consider the sequences where $s_{\overline{c}}^{f(k)-2}$, s_c^{k-1} or $s_{\overline{c}}^{f(k)-1}$ is equal to $switch.1b$. * means that this switch reason can be either one of the five switch reasons (except for $switch.1b$)

In the same way we can prove that the sequence $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.1a \rightarrow switch.1a \rightarrow switch.3b$ is not possible for $k \geq n^{\mathcal{C}_1} + 1$. We prove that all sequences $s_{\overline{c}}^{f(k)-2} \rightarrow s_c^{k-1} \rightarrow s_{\overline{c}}^{f(k)-1} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1}$, where $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.1a \rightarrow switch.1a \rightarrow switch.3b$ are not possible if $k \geq n^{\mathcal{C}_1} + 1$.

First of all, whenever $s_{\overline{c}}^{f(k)-2}$, s_c^{k-1} or $s_{\overline{c}}^{f(k)-1}$ is equal to $switch.1b$ then $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} = switch.1a \rightarrow switch.1a \rightarrow switch.3b$ is not possible because of Lemma D.23.

All other possible sequences are shown in the Table D.5.

■

Lemma D.13 *The following sequences are not possible:*

1 $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} \rightarrow s_c^{k+2} = switch.2 \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.2$

2 $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1} \rightarrow s_c^{k+2} = switch.3b \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.1a \rightarrow switch.3b$

Proof. The queue length of a queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ at the start of $g_{i_{\overline{c}}}^{f(k)}$ is at least $(\tilde{g}_{i_c}^{max} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$. The queue length of a queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ at the start of $g_{i_{\overline{c}}}^{f(k)+1}$ is equal to $(g_{i_c}^{k+1} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}} \leq (\tilde{g}_{i_c}^{max} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_{\overline{c}}}$. Because we could empty each queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ during $g_{i_{\overline{c}}}^{f(k)}$ we are also able to empty each queue during $g_{i_{\overline{c}}}^{f(k)+1} \leq g_{i_{\overline{c}}}^{f(k)}$.

Further, the queue length at queue $i_c \in \mathcal{G}_c$ is at least $(g_{i_{\overline{c}}}^{f(k)} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_c}$ at the beginning of $g_{i_c}^{k+1}$. The queue length at queue $i_c \in \mathcal{G}_c$ is equal to $(g_{i_{\overline{c}}}^{f(k)+1} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_c} \leq (g_{i_{\overline{c}}}^{f(k)} + \sigma_{i_c,i_{\overline{c}},i_c})\lambda_{i_c}$ at the beginning of $g_{i_c}^{k+2}$. Thus at each queue $i_c \in \mathcal{G}_c$ the queue length at the beginning of $g_{i_c}^{k+2}$ is less than (or equal to) the queue length at the beginning of $g_{i_c}^{k+1}$. We could empty queue i_c during a green time $g_{i_c}^{k+1} \leq \tilde{g}_{i_c}^{max}$. Hence, we are also able to empty queue i_c during a green time $g_{i_c}^{k} \leq g_{i_c}^{k+1} \leq \tilde{g}_{i_c}^{max}$. However, $s_c^{k+2} = switch.2$ or $s_c^{k+2} = switch.3b$ can only occur if we are not able to empty all queues during $g_{i_c}^{k} \leq \tilde{g}_{i_c}^{max}$ (see D.1 for the definitions of the switch reasons $switch.2$ and $switch.3b$). ■

D.4.2 Lemmas Excluding Infinite Sequences of Switch Reasons

Lemma D.14 *The following infinite sequences are not possible:*

sequence 1 *An infinite sequence of switch reasons* $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)+1} \rightarrow \ldots,\ k > 1$ *where each stop reason is part of combination* C_8.

sequence 2 *An infinite sequence of switch reasons* $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)+1} \rightarrow \ldots,\ k > 1$ *where each stop reason is part of combination* C_{11}.

sequence 3 *An infinite sequence of switch reasons* $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^k \rightarrow s_{\overline{c}}^{f(k)+1} \rightarrow \ldots,\ k > 1$ *where each stop reason is part of combination* C_{18} *or part of combination* C_{19}.

Proof. First of all, we know that $g_{i_c}^{h} > g_{i_c}^{pbt}$, $\forall i_c \in \mathcal{G}_c$, $\forall h \geq k$ and that $g_{i_{\overline{c}}}^{f(h)} > g_{i_{\overline{c}}}^{pbt}$, $\forall h \geq k$ from (8.1h), (8.1k), (8.1n) and (8.1q).

We distinguish the following two types of combinations:

type 1 combination $s_c^h \rightarrow s_{\overline{c}}^{f(h)}$ and $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{h+1} > 0$, $h \geq k$.

type 2 combination $s_c^h \rightarrow s_{\overline{c}}^{f(h)}$ and $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{h+1} > 0$, $h \geq k$.

This proof goes as follows. If an infinite sequence of switch reasons (sequence 1, 2 or 3) is possible then we can prove that either an infinite sequence of switch reasons, where all switch reasons are part of a type 1 combination must be possible or an infinite sequence of switch reasons, where all switch reasons are part of a type 2 combination must be possible. We can prove that both are not possible. Hence, we know that sequence 1, sequence 2 and sequence 3 cannot occur.

Lets assume infinite sequences 1,2 and 3 are possible. From (D.7b) and (D.7c) we can easily see (see also Figure D.9) that whenever $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}}, h \geq k \left(\Delta x_{c,i_{\overline{c}}}^{h+1} > 0\right)$ for such an infinite sequence (infinite sequences 1,2 or 3) then it holds that the same queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ satisfies $\Delta x_{c,i_{\overline{c}}}^{h+1} = \Delta x_{c,i_{\overline{c}}}^{h+2} > 0$, $h \geq k$ because $g_{i_{\overline{c}}}^{\mu,h} = g_{i_{\overline{c}}}^{h} = g_{i_{\overline{c}}}^{\mu,h+1} = g_{i_{\overline{c}}}^{h+1}$ and $g_{i_c}^{h} = g_{i_c}^{h+1}$. Hence, when we are given an infinite sequence (either, sequence 1, sequence 2 or sequence 3) where s_c^h, $h \geq k$ and $s_{\overline{c}}^{f(h)}$ form a type 1 combination then the switch reason s_c^{h+z}, $z \in \mathcal{N}$ ($\mathcal{N}$ is the set of non-negative integers) and the switch reason $s_{\overline{c}}^{f(h)+z}$ form a type 1 combination.

In Figure D.9, i_c could refer to any signal in the set $\mathcal{G}_c$ and $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ is a signal that satisfies $\Delta x_{c,i_{\overline{c}}}^{h+1} > 0$. In Figure D.9 we can distinguish the following sections:

1 between point 1 and point 2, $g_{i_c}^{h}$ is performed.
2 between point 2 and point 3,the setup $\sigma_{i_{\overline{c}},i_c,i_{\overline{c}}}$ is performed.
3 between point 3 and point 4, $g_{i_{\overline{c}}}^{f(h)}$ is performed.
4 between point 4 and point 5, $g_{i_c}^{h+1}$ is performed.
5 between point 5 and point 6,the setup $\sigma_{i_{\overline{c}},i_c,i_{\overline{c}}}$ is performed.
6 between point 6 and point 7, $g_{i_{\overline{c}}}^{f(h)+1}$ is performed.

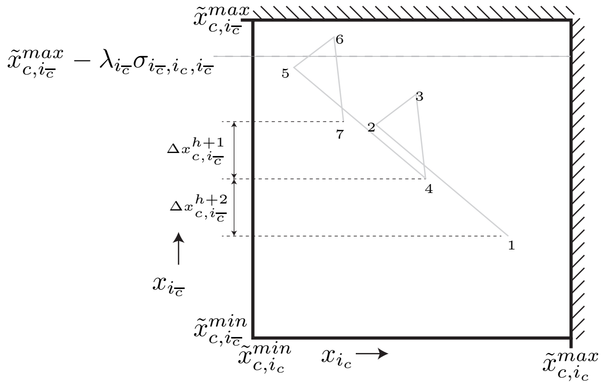


Figure D.9: Whenever $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}}, h \geq k \left(\Delta x^{h+1}_{c,i_{\overline{c}}} > 0\right)$ for infinite sequences 1,2 and 3 then it holds that the same queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ satisfies $\Delta x^{h+1}_{c,i_{\overline{c}}} = \Delta x^{h+2}_{c,i_{\overline{c}}} > 0$, $h \geq k$

Hence, whenever for infinite sequences 1,2 and 3 a switch reason is part of a type 1 combination then this infinite sequence contains an infinite sequence of switch reasons that are all part of a type 1 combination. On the other hand, whenever for infinite sequences 1,2 and 3 no switch reason is part of a type 1 combination then this infinite sequence is an infinite sequence of switch reasons that are all part of a type 2 combination.

We can show that an infinite sequence of switch reasons that are all part of a type 1 combination is not possible and that an infinite sequence of switch reasons that are all part of a type 2 combination is not possible

For an infinite sequence of switch reasons that are all part of a type 1 combination we can use Lemma D.19. Using Lemma D.19 we can see that each queue $i_c \in \mathcal{G}_c$ either empties, i.e. the queue length is equal to zero at the end of its green time, or its queue length decreases minimally $\Delta_c(g^{f(h)}_{i_{\overline{c}}}) > 0$, $h \geq k$. Note that $\Delta_c(g^{f(h)+1}_{i_{\overline{c}}}) = \Delta_c(g^{f(h)}_{i_{\overline{c}}})$, $h \geq k$ since $g^{f(h)}_{i_{\overline{c}}} = g^{f(h)+1}_{i_{\overline{c}}}$. As a result, for an infinite sequence of switch reasons that are part of a type 1 combination, the queues in the set $\mathcal{G}_c$ are eventually all empty (and we switch for the reason $switch.1a$ or $switch.1b$). Whenever we switch for the reason $switch.1a$ or $switch.1b$ this switch reason cannot be part of a combination C_8, C_{11}, C_{18} or C_{19}. Thus, an infinite sequence where each switch reason is part of type 1 combination is not possible.

Now we consider an infinite sequence of switch reasons that are all part of a type 2 combination. We can derive:

$$x^{h+1}_{c,i_c} = x^{h}_{c,i_c} - g^{\mu,h}_{i_c}(\mu_{i_c} - \lambda_{i_c}) + \tilde{r}^{max}_{i_c}\lambda_{i_c},$$

where

$$g^{\mu,h}_{i_c} = \min\{\tilde{g}^{max}_{i_c}, \frac{x^{h}_{c,i_c} - \tilde{x}^{min}_{c,i_c}}{\mu_{i_c} - \lambda_{i_c}}\}.$$

Here $g^{\mu,h}_{i_c}(\mu_{i_c} - \lambda_{i_c})$ is the net amount of traffic that is processed during the green period $g^h_{i_c}$ and $(\sigma_{i_c,i_{\overline{c}},i_c} + g^{max}_{i_{\overline{c}}})\lambda_{i_c}$ is the amount of traffic that arrives during a red period. Note that whenever $g^{\mu,h}_{i_c} = \frac{x^h_{c,i_c} - \tilde{x}^{min}_{c,i_c}}{\mu_{i_c} - \lambda_{i_c}}$ then buffer i_c is emptied during $g^h_{i_c}$.

We can distinguish two types of signals:

type 1 a signal that satisfies $i_c \in \mathcal{G}_c$ satisfies $\tilde{g}_{i_c}^{max} \geq \frac{\rho_{i_c}\tilde{r}_{i_c}^{max}}{1-\rho_{i_c}}$

type 2 a signal that satisfies $i_c \in \mathcal{G}_c$ satisfies $\tilde{g}_{i_c}^{max} < \frac{\rho_{i_c}\tilde{r}_{i_c}^{max}}{1-\rho_{i_c}}$

A signal of type 1 satisfies:

$$\Delta x_{c,i_c}^{h+1} = -g_{i_c}^{\mu,h}(\mu_{i_c} - \lambda_{i_c}) + \tilde{r}_{i_c}^{max}\lambda_{i_c} \leq -g_{i_c}^{max}(\mu_{i_c} - \lambda_{i_c}) + \tilde{r}_{i_c}^{max}\lambda_{i_c} \leq 0. \tag{D.18}$$

Hence a type 1 signal $i_c \in \mathcal{G}_c$ could never satisfy $\Delta x_{c,i_c}^{h+1} > 0$. Hence, if an infinite sequence of switch reasons that are all part of a type 2 combination, is possible then it must hold that there is a signal of type 2.

for a type 2 signal we can find:

$$\begin{aligned}
&\Delta x_{c,i_c}^{h+1} = \tilde{r}_{i_c}^{max}\lambda_{i_c} - \tilde{g}_{i_c}^{max}(\mu_{i_c} - \lambda_{i_c}) > 0 && \text{if } \tilde{g}_{i_c}^{max} \leq \frac{x_{c,i_c}^{h} - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}},\\
&\Delta x_{c,i_c}^{h+1} = -x_{c,i_c}^{h} + \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c} > 0 && \text{if } \tilde{g}_{i_c}^{max} > \frac{x_{c,i_c}^{h} - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}} \wedge x_{c,i_c}^{h} < \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c},\\
&\Delta x_{c,i_c}^{h+1} = x_{c,i_c}^{h} - \tilde{x}_{c,i_c}^{min} + (\sigma_{i_c,i_{\overline{c}},i_c} + g_{i_{\overline{c}}}^{max})\lambda_{i_c} \geq 0 && \text{if } \tilde{g}_{i_c}^{max} > \frac{x_{c,i_c}^{h} - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}} \wedge x_{c,i_c}^{h} \geq \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c}
\end{aligned}$$

Note that $\tilde{g}_{i_c}^{max} > \frac{x_{c,i_c}^{h} - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}} \wedge x_{c,i_c}^{h} \geq \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c}$ cannot be satisfied because when we fill in $x_{c,i_c}^{h} \geq \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c}$ into $\tilde{g}_{i_c}^{max} > \frac{x_{c,i_c}^{h} - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}}$ the result is $\tilde{g}_{i_c}^{max} > \frac{\rho_{i_c}\tilde{r}_{i_c}^{max}}{1-\rho_{i_c}}$. However, a signal of type 2 satisfies (by definition) $\tilde{g}_{i_c}^{max} < \frac{\rho_{i_c}\tilde{r}_{i_c}^{max}}{1-\rho_{i_c}}$.

Note that whenever a type 1 signal satisfies $x_{c,i_c}^{h} < \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c}$ then it holds that $x_{c,i_c}^{h+1} > \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c}$. Further, when a type 3 signal satisfies $x_{c,i_c}^{h+1} > \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c}$ then $\tilde{g}_{i_c}^{max} = g_{i_c}^{\mu,h}$. Thus, a (type 1) signal $i_c \in \mathcal{G}_c$ satisfies $\Delta x_{c,i_c}^{h+1} > 0, \forall h > k$ and $g_{i_c}^{\mu,h} = \tilde{g}_{i_c}^{max} = g_{i_c}^{h}, \forall h > k$.

In the beginning of the proof we showed that $g_{i_c}^{h} > g_{i_c}^{pbt}, \forall h \geq k$. Hence, we can use Lemma D.18. From Lemma D.18 it follows that each queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ either empties, i.e. the queue length is zero at the end of its green time $g_{i_{\overline{c}}}^{f(h)}$, $h > k$, or its queue length decreases minimally $\Delta_{\overline{c}}(g_{i_c}^{h})$ during the hth cycle c. Note that $g_{i_c}^{h} = g_{i_c}^{h+1} = \tilde{g}_{i_c}^{max}, \forall h \geq k$. Hence, eventually all queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ are emptied (and we stop serving the signals in $\mathcal{G}_{\overline{c}}$ for the reason *switch*.1*a* or *switch*.1*b*). Whenever we switch for the reason *switch*.1*a* or *switch*.1*b* this switch reason cannot be part of a combination C_8, C_{11}, C_{18} or C_{19}. Thus, an infinite sequence of switch reasons where each switch reason is part of type 2 combination is not possible. ■

Lemma D.15 *The following sequences are not possible:*

sequence 1 *An infinite sequence of stop reasons* $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^k \to s_{\overline{c}}^{f(k)+1} \to \dots = switch.2 \to switch.2 \to switch.2 \to switch.2 \to \dots$, *where* $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} \leq 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$ *(note that* s_c^k *and* $s_{\overline{c}}^{f(k)}$ *form combination* C_7*).*

sequence 2 *An infinite sequence of stop reasons* $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^k \to s_{\overline{c}}^{f(k)+1} \to \dots = switch.2 \to switch.3b \to switch.2 \to switch.3b \to \dots$, *where* $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$ *(note that* s_c^k *and* $s_{\overline{c}}^{f(k)}$ *form combination* C_{11}*).*

sequence 3 *An infinite sequence of stop reasons* $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^k \to s_{\overline{c}}^{f(k)+1} \to \dots = switch.3b \to switch.3b \to switch.3b \to switch.3b \to \dots$, *where* $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} = 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} = 0$ *(note that* s_c^k *and* $s_{\overline{c}}^{f(k)}$ *form combination* C_{17}*).*

Further, we know that:

1 *When* $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k+1} \rightarrow s_{\overline{c}}^{f(k)+1}$ = *switch.2* $\rightarrow$ *switch.2* $\rightarrow$ *switch.2* $\rightarrow$ *switch.2, where* $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} \leq 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$ *then it holds that* $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+2} \leq 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+2} \leq 0$.

2 *When* $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k} \rightarrow s_{\overline{c}}^{f(k)+1}$ = *switch.2* $\rightarrow$ *switch.3b* $\rightarrow$ *switch.2* $\rightarrow$ *switch.3b, where* $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$ *then it holds that* $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+2} \leq 0$

3 *When* $s_c^k \rightarrow s_{\overline{c}}^{f(k)} \rightarrow s_c^{k} \rightarrow s_{\overline{c}}^{f(k)+1}$ = *switch.3b* $\rightarrow$ *switch.3b* $\rightarrow$ *switch.3b* $\rightarrow$ *switch.3b, where* $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} = 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} = 0$ *then it holds that* $\max_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} = 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+2} = 0$.

Proof. Lets assume infinite sequence 1,2 and 3 are possible. For all these sequences it holds that:

$$g_{i_c}^h = \tilde{g}_{i_c}^{max}, \forall i_c \in \mathcal{G}_c, \forall h \geq k, \tag{D.19}$$

$$g_{i_{\overline{c}}}^{f(h)} = \tilde{g}_{i_{\overline{c}}}^{max}, \forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}, \forall h \geq k. \tag{D.20}$$

Using (D.7a) and (D.7d) we can find that the queue length x_{c,i_c}^{h+1}, $h \geq k$ can be calculated according to the following equation (when assuming infinite sequence 1, 2 or 3):

$$x_{c,i_c}^{h+1} = x_{c,i_c}^h - g_{i_c}^{\mu,h}(\mu_{i_c} - \lambda_{i_c}) + \tilde{r}_{i_c}^{max}\lambda_{i_c},$$

where

$$g_{i_c}^{\mu,h} = \min\{\tilde{g}_{i_c}^{max}, \frac{x_{c,i_c}^h - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}}\}.$$

Note that a signal $i_c \in \mathcal{G}_c$ is emptied whenever $g_{i_c}^{\mu,h} = \frac{x_{c,i_c}^h - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}}$. In the proof of Lemma D.14 we have shown that when $\tilde{g}_{i_c}^{max} < \frac{\tilde{r}_{i_c}^{max}\rho_{i_c}}{1-\rho_{i_c}}$ then $\exists h \geq k \left(\Delta x_{c,i_c}^{h+1} = \tilde{r}_{i_c}^{max}\lambda_{i_c} - \tilde{g}_{i_c}^{max}(\mu_{i_c} - \lambda_{i_c}) > 0\right)$. Hence, sequence 1 and sequence 3 can only occur if all signals $i_c \in \mathcal{G}_c$ satisfy $\tilde{g}_{i_c}^{max} \geq \frac{\tilde{r}_{i_c}^{max}\rho_{i_c}}{1-\rho_{i_c}}$.

When a signal $i_c \in \mathcal{G}_c$ satisfies $\tilde{g}_{i_c}^{max} \geq \frac{\tilde{r}_{i_c}^{max}\rho_{i_c}}{1-\rho_{i_c}}$ then we can find:

$$\begin{aligned}
&\Delta x_{c,i_c}^{h+1} = -x_{c,i_c}^h + \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c} > 0 && \text{if } \tilde{g}_{i_c}^{max} > \frac{x_{c,i_c}^h - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}} \wedge x_{c,i_c}^h < \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c}, \\
&\Delta x_{c,i_c}^{h+1} = -x_{c,i_c}^h + \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c} < 0 && \text{if } \tilde{g}_{i_c}^{max} > \frac{x_{c,i_c}^h - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}} \wedge x_{c,i_c}^h > \tilde{x}_{c,i_c}^{min} + (\sigma_{i_c,i_{\overline{c}},i_c} + g_{i_{\overline{c}}}^{max})\lambda_{i_c}, \\
&\Delta x_{c,i_c}^{h+1} = -x_{c,i_c}^h + \tilde{x}_{c,i_c}^{min} + \tilde{r}_{i_c}^{max}\lambda_{i_c} = 0 && \text{if } \tilde{g}_{i_c}^{max} > \frac{x_{c,i_c}^h - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}} \wedge x_{c,i_c}^h = \tilde{x}_{c,i_c}^{min} + (\sigma_{i_c,i_{\overline{c}},i_c} + g_{i_{\overline{c}}}^{max})\lambda_{i_c}, \\
&\Delta x_{c,i_c}^{h+1} = \tilde{r}_{i_c}^{max}\lambda_{i_c} - \tilde{g}_{i_c}^{max}(\mu_{i_c} - \lambda_{i_c}) < 0 && \text{if } \frac{\tilde{r}_{i_c}^{max}\rho_{i_c}}{1 - \rho_{i_c}} < \tilde{g}_{i_c}^{max} \leq \frac{x_{c,i_c}^h - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}}, \\
&\Delta x_{c,i_c}^{h+1} = \tilde{r}_{i_c}^{max}\lambda_{i_c} - \tilde{g}_{i_c}^{max}(\mu_{i_c} - \lambda_{i_c}) = 0 && \text{if } \frac{\tilde{r}_{i_c}^{max}\rho_{i_c}}{1 - \rho_{i_c}} = \tilde{g}_{i_c}^{max} \leq \frac{x_{c,i_c}^h - \tilde{x}_{c,i_c}^{min}}{\mu_{i_c} - \lambda_{i_c}}.
\end{aligned} \tag{D.21}$$

Note the following things:

1 Whenever $\Delta x^{h+1}_{c,i_c} \leq 0$ then $\Delta x^{h+2}_{c,i_c} \leq 0$. Hence, it also holds that whenever $\max_{i_c \in \mathcal{G}_c} \Delta x^{h+1}_{c,i_c} \leq 0$ then $\max_{i_c \in \mathcal{G}_c} \Delta x^{h+2}_{c,i_c} \leq 0$.

2 Whenever $\Delta x^{h+1}_{c,i_c} = 0$ then $\Delta x^{h+2}_{c,i_c} = 0$. Hence, it also holds that whenever $\max_{i_c \in \mathcal{G}_c} \Delta x^{h+1}_{c,i_c} = 0$ then $\max_{i_c \in \mathcal{G}_c} \Delta x^{h+2}_{c,i_c} = 0$.

3 Whenever a queue $i_c \in \mathcal{G}_c$ that satisfies $\tilde{g}^{max}_{i_c} > \frac{\tilde{r}^{max}_{i_c}\rho_{i_c}}{1-\rho_{i_c}}$ is emptied during its green time $g^h_{i_c}$ then it is emptied during al subsequent green times, i.e. when $\tilde{g}^{max}_{i_c} \geq \frac{x^h_{c,i_c} - \tilde{x}^{min}_{c,i_c}}{\mu_{i_c}-\lambda_{i_c}}$ then $\tilde{g}^{max}_{i_c} \geq \frac{x^{h+1}_{c,i_c} - \tilde{x}^{min}_{c,i_c}}{\mu_{i_c}-\lambda_{i_c}}$.

4 All queues $i_c \in \mathcal{G}_c$ that satisfy $\tilde{g}^{max}_{i_c} > \frac{\tilde{r}^{max}_{i_c}\rho_{i_c}}{1-\rho_{i_c}}$ are emptied eventually (in finite time); if $x^h_{c,i_c} > \tilde{g}^{max}_{i_c}(\mu_{i_c} - \lambda_{i_c}) - \tilde{r}^{max}_{i_c}\lambda_{i_c}$ then $\Delta x^{h+1}_{c,i_c} = \tilde{r}^{max}_{i_c}\lambda_{i_c} - \tilde{g}^{max}_{i_c}(\mu_{i_c} - \lambda_{i_c}) < 0$ and if $x^h_{c,i_c} \leq \tilde{g}^{max}_{i_c}(\mu_{i_c} - \lambda_{i_c}) - \tilde{r}^{max}_{i_c}\lambda_{i_c}$ then the queue is emptied (and it is emptied during all its subsequent green times).

5 It could be that a queue $i_c \in \mathcal{G}_c$ is never emptied if $\tilde{g}^{max}_{i_c} \geq \frac{\tilde{r}^{max}_{i_c}\rho_{i_c}}{1-\rho_{i_c}}$ (recall that for sequence 1 and sequence 3 it cannot hold that $\tilde{g}^{max}_{i_c} > \frac{\tilde{r}^{max}_{i_c}\rho_{i_c}}{1-\rho_{i_c}}$)

Using (D.7a) and (D.7d) we can find that the queue length $x^{h+1}_{c,i_{\overline{c}}}$, $h \geq k$ can be calculated according to the following equation (when assuming infinite sequence 1, 2 or 3):

$$x^{h+1}_{c,i_{\overline{c}}} = x^h_{c,i_{\overline{c}}} + \tilde{r}^{max}_{i_{\overline{c}}}\lambda_{i_{\overline{c}}} - g^{\mu,f(h)}_{i_{\overline{c}}}(\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}), \tag{D.22}$$

where

$$g^{\mu,f(h)}_{i_{\overline{c}}} = \min\{\tilde{g}^{max}_{i_{\overline{c}}}, \frac{x^{f(h)}_{\overline{c},i_{\overline{c}}} - \tilde{x}^{min}_{\overline{c},i_{\overline{c}}}}{\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}}\}.$$

When $g^{\mu,f(h)}_{i_{\overline{c}}} = \frac{x^h_{\overline{c},i_{\overline{c}}} - \tilde{x}^{min}_{c,i_{\overline{c}}}}{\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}}$ then queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ will be emptied during $g^{f(h)}_{i_{\overline{c}}}$.

Using (D.9) we can find rewrite (D.22) to:

$$x^{h+1}_{c,i_{\overline{c}}} = \max\{x^h_{c,i_{\overline{c}}} + \tilde{r}^{max}_{i_{\overline{c}}}\lambda_{i_{\overline{c}}} - \tilde{g}^{max}_{i_{\overline{c}}}(\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}), \tilde{x}^{min}_{c,i_{\overline{c}}}\}. \tag{D.23}$$

Note that when $\tilde{g}^{max}_{i_{\overline{c}}} < \frac{\tilde{r}^{max}_{i_{\overline{c}}}\rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}$ then it holds that $x^h_{c,i_{\overline{c}}} + \tilde{r}^{max}_{i_{\overline{c}}}\lambda_{i_{\overline{c}}} > \tilde{x}^{min}_{c,i_{\overline{c}}}$ because $x^h_{c,i_{\overline{c}}} \geq \tilde{x}^{min}_{c,i_{\overline{c}}}$ ($x^h_{c,i_{\overline{c}}} < \tilde{x}^{min}_{c,i_{\overline{c}}}$ implies a negative queue length) and $\tilde{r}^{max}_{i_{\overline{c}}}\lambda_{i_{\overline{c}}} - \tilde{g}^{max}_{i_{\overline{c}}}(\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}) > 0$.

As a result it follows that $\Delta x^{h+1}_{c,i_{\overline{c}}} = \tilde{r}^{max}_{i_{\overline{c}}}\lambda_{i_{\overline{c}}} - g^{\mu,h}_{i_{\overline{c}}}(\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}) = \tilde{r}^{max}_{i_{\overline{c}}}\lambda_{i_{\overline{c}}} - \tilde{g}^{max}_{i_{\overline{c}}}(\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}) > 0$. Hence, sequence 1, sequence 2 and sequence 3 can only occur if all signals $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ satisfy $\tilde{g}^{max}_{i_{\overline{c}}} \geq \frac{\tilde{r}^{max}_{i_{\overline{c}}}\rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}$. Further, we should note that $x^h_{c,i_{\overline{c}}} < \tilde{x}^{min}_{c,i_{\overline{c}}}$ is not possible since it implies that queue $i_{\overline{c}}$ had a negative queue length.

For a signal for which it holds that $\tilde{g}^{max}_{i_{\overline{c}}} \geq \frac{\tilde{r}^{max}_{i_{\overline{c}}}\rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}$, we can find that:

$$\begin{aligned}
&\Delta x^{h+1}_{c,i_{\overline{c}}} = 0 && \text{if } x^h_{c,i_{\overline{c}}} = \tilde{x}^{min}_{c,i_{\overline{c}}}, \\
&\Delta x^{h+1}_{c,i_{\overline{c}}} = 0 && \text{if } \tilde{g}^{max}_{i_{\overline{c}}} = \frac{\tilde{r}^{max}_{i_{\overline{c}}}\rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}, \\
&\Delta x^{h+1}_{c,i_{\overline{c}}} < 0 && \text{if } x^h_{c,i_{\overline{c}}} > \tilde{x}^{min}_{c,i_{\overline{c}}} \wedge \tilde{g}^{max}_{i_{\overline{c}}} > \frac{\tilde{r}^{max}_{i_{\overline{c}}}\rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}.
\end{aligned}$$

Note the following things:

1 Whenever $\Delta x^{h+1}_{c,i_{\overline{c}}} \leq 0$ then $\Delta x^{h+2}_{c,i_{\overline{c}}} \leq 0$. Hence, it also holds that whenever $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+1}_{c,i_{\overline{c}}} \leq 0$ then $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+2}_{c,i_{\overline{c}}} \leq 0$.

2 Whenever $\Delta x^{h+1}_{c,i_{\overline{c}}} = 0$ then $\Delta x^{h+2}_{c,i_{\overline{c}}} = 0$. Hence, it also holds that whenever $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+1}_{c,i_{\overline{c}}} = 0$ then $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+2}_{c,i_{\overline{c}}} = 0$.

3 Whenever queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ is emptied during its green time $g^{f(h)}_{i_{\overline{c}}}$ it will be emptied during al subsequent green times, i.e. if $x^h_{c,i_{\overline{c}}} = \tilde{x}^{min}_{c,i_{\overline{c}}}$ then $x^{h+1}_{c,i_{\overline{c}}} = \tilde{x}^{min}_{c,i_{\overline{c}}}$.

4 All queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ that satisfy $\tilde{g}^{max}_{i_{\overline{c}}} > \frac{\tilde{r}^{max}_{i_{\overline{c}}} \rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}$ will be emptied eventually (in finite time); if $x^h_{c,i_{\overline{c}}} > \tilde{g}^{max}_{i_{\overline{c}}}(\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}) - \tilde{r}^{max}_{i_{\overline{c}}} \lambda_{i_c}$ then $\Delta x^{h+1}_{c,i_{\overline{c}}} = \tilde{r}^{max}_{i_{\overline{c}}} \lambda_{i_{\overline{c}}} - \tilde{g}^{max}_{i_{\overline{c}}}(\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}) < 0$ and if $x^h_{c,i_{\overline{c}}} \leq \tilde{g}^{max}_{i_{\overline{c}}}(\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}) - \tilde{r}^{max}_{i_{\overline{c}}} \lambda_{i_{\overline{c}}}$ then the queue is emptied (and it will be emptied during all its subsequent green times).

5 It could be that whenever a queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ satisfies $\tilde{g}^{max}_{i_{\overline{c}}} = \frac{\tilde{r}^{max}_{i_{\overline{c}}} \rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}$ that this queue will never be emptied.

Now we know that:

1 Whenever $\max_{i_c \in \mathcal{G}_c} \Delta x^{h+1}_{c,i_c} \leq 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+1}_{c,i_{\overline{c}}} \leq 0$ then $\max_{i_c \in \mathcal{G}_c} \Delta x^{h+2}_{c,i_c} \leq 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+2}_{c,i_{\overline{c}}} \leq 0$. As a result, (for infinite sequence 1) each switch reason s^h_c, $h \geq k$ forms a combination C_7 with switch reason $s^{f(h)}_{\overline{c}}$.

2 Whenever $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+1}_{c,i_{\overline{c}}} \leq 0$ then $\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+2}_{c,i_{\overline{c}}} \leq 0$. As a result, (for infinite sequence 2) each switch reason s^h_c, $h \geq k$ forms a combination C_{10} with switch reason $s^{f(h)}_{\overline{c}}$.

3 Whenever $\max_{i_c \in \mathcal{G}_c} \Delta x^{h+1}_{c,i_c} = 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+1}_{c,i_{\overline{c}}} = 0$ then $\max_{i_c \in \mathcal{G}_c} \Delta x^{h+2}_{c,i_c} = 0 \wedge \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x^{h+2}_{c,i_{\overline{c}}} = 0$. As a result (for infinite sequence 3) each switch reason s^h_c, $h \geq k$ forms a combination C_{17} with switch reason $s^{f(h)}_{\overline{c}}$.

First lets consider sequence 1 and sequence 3. Recall that when $\left(\tilde{g}^{max}_{i_c} = \frac{\tilde{r}^{max}_{i_c} \rho_{i_c}}{1-\rho_{i_c}}\right)$ then it is possible that queue $i_c \in \mathcal{G}_c$ is never emptied. In the same way when $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} \left(\tilde{g}^{max}_{i_{\overline{c}}} = \frac{\tilde{r}^{max}_{i_{\overline{c}}} \rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}\right)$ then it is possible that this queue is never emptied.

However, using (D.6) we can find that $\exists i_c \in \mathcal{G}_c : \tilde{g}^{max}_{i_c} \geq \frac{\tilde{r}^{max}_{i_c} \rho_{i_c}}{1-\rho_{i_c}}$ and $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \tilde{g}^{max}_{i_{\overline{c}}} = \frac{\tilde{r}^{max}_{i_{\overline{c}}} \rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}$ can only hold whenever $\exists i_c \in \mathcal{G}_c, i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \tilde{g}^{max}_{i_c} = \frac{\sigma_{i_c,i_{\overline{c}},i_c} \rho_{i_c}}{1-\rho_{i_c}-\rho_{i_{\overline{c}}}}$. However, this does not hold because of the inequalities (8.1h), (8.1k), (8.1n) and (8.1q). Hence, either all queues in the set $\mathcal{G}_c$ empty in a finite time or all queues in the set $\mathcal{G}_{\overline{c}}$ empty in a finite time. In this case we do stop because of the reason *switch*.1*a* or *switch*.1*b*. Hence, sequence 1 and sequence 3 are not possible.

Now lets consider sequence 2. Recall that when $\exists i_c \in \mathcal{G}_c : \tilde{g}^{max}_{i_c} \geq \frac{\tilde{r}^{max}_{i_c} \rho_{i_c}}{1-\rho_{i_c}}$ then it is possible that this queue is never emptied. In the same way when $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \tilde{g}^{max}_{i_{\overline{c}}} = \frac{\tilde{r}^{max}_{i_{\overline{c}}} \rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}$ then it is possible that this queue is never emptied.

However, using (D.6) we can find that $\exists i_c \in \mathcal{G}_c : \tilde{g}^{max}_{i_c} \geq \frac{\tilde{r}^{max}_{i_c} \rho_{i_c}}{1-\rho_{i_c}}$ and $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \tilde{g}^{max}_{i_{\overline{c}}} = \frac{\tilde{r}^{max}_{i_{\overline{c}}} \rho_{i_{\overline{c}}}}{1-\rho_{i_{\overline{c}}}}$ can only hold whenever $\exists i_c \in \mathcal{G}_c, i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \tilde{g}^{max}_{i_c} \leq \frac{\sigma_{i_c,i_{\overline{c}},i_c} \rho_{i_c}}{1-\rho_{i_c}-\rho_{i_{\overline{c}}}}$. However, this does not hold because of the inequalities (8.1h), (8.1k), (8.1n) and (8.1q). Hence, either all queues in the set $\mathcal{G}_c$ empty in a finite time or all queues in the set $\mathcal{G}_{\overline{c}}$ empty in a finite time. In this case we do stop because of the reason $switch.1a$ or $switch.1b$. Hence, sequence 2 is not possible. ∎

Lemma D.16 *The following infinite sequences are not possible:*

sequence 1 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} \to \cdots = switch.1a \to switch.3b \to switch.1a \to switch.3b \to \dots$

sequence 2 $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} \to \cdots = switch.1a \to switch.2 \to switch.1a \to switch.2 \to \dots$

Proof. For sequence 1 and sequence 2 it holds that every queue $i_c \in \mathcal{G}_c$ is empty at the end of its green time $g_{i_c}^h$, $h \geq k$. The green period $g_{i_c}^{h+1}$, $i_c \in \mathcal{G}_c$ starts $\tilde{r}_{i_c}^{max}$ seconds after the end of $g_{i_c}^h$. However, the green period $g_{i_c}^{h+1}$ starts $\sigma_{i_c}^{res}$ seconds after the start of the $h+1$th cycle c. Therefore, the $h+1$th cycle c starts $\tilde{r}_{i_c}^{max} - \sigma_{i_c}^{res}$ after the end of $g_{i_c}^h$. During this time the queue length increases with rate λ_{i_c}. Hence, it holds that:

$$x_{c,i_c}^h = (\tilde{r}_{i_c}^{max} - \sigma_{i_c}^{res})\lambda_{i_c}, \quad i_c \in \mathcal{G}_c, \quad h > k. \tag{D.24}$$

Hence, it holds that:

$$\Delta x_{c,i_c}^{h+1} = 0, \quad , i_c \in \mathcal{G}_c, h > k. \tag{D.25}$$

We distinguish the following three reasons for $s_c^h = switch.1a$, $h > k$ (either $s_c^h = switch.1a$ or $s_c^h = switch.1b$):

reason 1 $s_c^h = switch.1$ and $\tau_{i_c^{r,f}}^1 = \tau_{i_c^{r,f}}^{1.1}$, i.e. we switch immediately at the moment that all queues (in the set that is served) are expected to empty.

reason 2 $s_c^h = switch.1$ and $\tau_{i_c^{r,f}}^1 = \tau_{i_c^{r,f}}^{1.2}$, i.e. we use green times are exactly large enough to satisfy all constraints on minimum green times.

Note that if $\tau_{i_c^{r,f}}^1 = \tau_{i_c^{r,f}}^{1.3}$ we switch for the reason $switch.1b$.

Note that for the second reason, there is a signal $i_c \in \mathcal{G}_c$ where $\tilde{r}_{i_c}^{max}\lambda_{i_c}$ of traffic could not depart during a green time that is exactly large enough to satisfy all constraints on minimum green times. Hence, when switch reason s_c^h, $h > k$ occurs for the second reason then s_c^h, $\forall h > k$ occurs for the second reason as well(for sequence 1 and 2). Hence, when sequence 1 or 2 is possible then either an infinite sequence where we switch s_c^h, $\forall h > k$ for the first reason must be possible or an infinite sequence where we switch s_c^h, $\forall h > k$ for the second reason. We are going to show that both are not possible must be possible.

First we consider an infinite sequence where we switch s_c^h, $\forall h > k$ for the first reason. At the beginning of this proof we have shown that $\Delta x_{c,i_c}^{h+1} = 0$, $\forall i_c \in \mathcal{G}_c$, $\forall h > k$. Further, in this case $\exists i_c \in \mathcal{G}_c : g_{i_c}^{\mu,h} = g_{i_c}^h$, $h > k$, i.e. there is a signal that we switch to red exactly at the moment that its queue is emptied. Further, it holds that $g_{i_{\overline{c}}}^{f(h)} > g_{i_{\overline{c}}}^{pbt}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$, $\forall h > k$ (because of the inequalities (8.1h), (8.1k), (8.1n) and (8.1q)).

Using (D.7a) and (D.7d), $g_{i_{\overline{c}}}^{f(h)} > g_{i_{\overline{c}}}^{pbt}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$, $\forall h > k$, $\Delta x_{c,i_c}^{h+1} = 0$, $\forall i_c \in \mathcal{G}_c$, $\forall h > k$ and $\exists i_c \in \mathcal{G}_c : g_{i_c}^{\mu,h} = g_{i_c}^h$ we can find that $\exists i_c \in \mathcal{G}_c : \Delta x_{c,i_c}^{h+1} \geq 0 \wedge g_{i_c}^{\mu,h} = g_{i_c}^h \wedge g_{i_c}^h > g_{i_c}^{pbt}$, $\forall h > k$.

Hence, we can use lemma D.18 to see that for an infinite sequence where we switch s_c^h, $\forall h > k$ for the first reason, either queue $i_{\overline{c}}$ goes empty during $g_{i_{\overline{c}}}^{f(h)} = \tilde{g}_{i_{\overline{c}}}^{max}$, $h > k$, i.e. the queue length is zero at the end of $g_{i_{\overline{c}}}^{f(h)}$ or its queue length decreases minimally $\Delta_{\overline{c}}(g_{i_c}^h) > 0$. (note that $g_{i_c}^h = g_{i_c}^{h+1}$, $\forall h > k$).

Hence, eventually all queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ are emptied (and we no longer stop serving the signals in the set $\mathcal{G}_{\overline{c}}$ because of the reason *switch*.3*b* or *switch*.2 but because of the reason *switch*.1*a* or *switch*.1*b*). Thus, an infinite sequence where we switch s_c^h, $\forall h > k$ for the first reason is not possible.

Now we consider an infinite sequence where we switch s_c^h, $\forall h > k$ for the second reason. Using the inequalities (8.1j), (8.1m), (8.1p) and (8.1p) we can see that each signal $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ can process less traffic during its (maximum possible) green time than what arrives during its (minimum possible) red time. Hence, eventually all queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ are emptied (and we no longer stop serving the signals in the set $\mathcal{G}_{\overline{c}}$ because of the reason *switch*.3*b* or *switch*.2 but because of the reason *switch*.1*a* or *switch*.1*b*). Thus, an infinite sequence where we switch s_c^h, $\forall h > k$ for the second reason is not possible.

■

D.4.3 Other Lemmas

Lemma D.17 *Whenever a queue length of signal $i_c \in \mathcal{G}_c$ goes from empty to full during $r_{i_c}^k$ then it must hold that $i_c \in \mathcal{S}_c$, $\mathcal{S}_c = \{i_c \in \mathcal{G}_c : \frac{x_{i_c}^{max}}{\lambda_{i_c}} - \sigma_{i_c,i_{\overline{c}},i_c} = \min\limits_{l_c \in \mathcal{G}_c} \frac{x_{l_c}^{max}}{\lambda_{l_c}} - \sigma_{l_c,i_{\overline{c}},l_c}\}$ and that all queue lengths of the signals in the set $l_c \in \mathcal{S}_c$ go from empty to full during their red period $r_{l_c}^k$.*

Proof. Queue $i_c \in \mathcal{G}_c$ goes from empty to the maximum queue length during $r_{i_c}^k$ whenever $r_{i_c}^k = \frac{x_{i_c}^{max}}{\lambda_{i_c}}$. Using (D.1) we can find that queue $i_c \in \mathcal{G}_c$ goes from empty to the maximum queue length during $r_{i_c}^k$ whenever:

$$g_{i_{\overline{c}}}^{f(k)} = \frac{x_{i_c}^{max}}{\lambda_{i_c}} - \sigma_{i_c,i_{\overline{c}},i_c}. \tag{D.26}$$

Hence, a green time of signal $i_{\overline{c}}$ can be at most $\min\limits_{l_c \in \mathcal{G}_c} \frac{x_{i_c}^{max}}{\lambda_{i_c}} - \sigma_{i_c,i_{\overline{c}},i_c}$ seconds, because otherwise a maximum queue length would be exceeded. Only the queue(s) $\arg\min\limits_{l_c \in \mathcal{G}_c} \frac{x_{i_c}^{max}}{\lambda_{i_c}} - \sigma_{i_c,i_{\overline{c}},i_c}$ can go from empty to full during its (their) red period(s) because the other queues need a longer green period $g_{i_{\overline{c}}}^k$ to go from empty to the maximum queue length and this is not possible.

Whenever a queue $i_c \in \mathcal{G}_c$ goes from empty to the maximum queue length during $r_{i_1}^k$, this means that $g_{i_{\overline{c}}}^k = \min\limits_{l_c \in \mathcal{G}_c} \frac{x_{i_c}^{max}}{\lambda_{i_c}} - \sigma_{i_c,i_{\overline{c}},i_c}$. This must mean that all queues $l_c \in \mathcal{S}_c$ go from empty to their maximum queue lengths during $r_{l_c}^k$ because when a queue in this set was not empty at the beginning of $r_{l_c}^k$ its maximum queue length would be exceeded when $g_{i_{\overline{c}}}^k = \min\limits_{l_c \in \mathcal{G}_c} \frac{x_{i_c}^{max}}{\lambda_{i_c}} - \sigma_{i_c,i_{\overline{c}},i_c}$ and if a queue $l_c \in \mathcal{S}_c$ was empty at the start of $r_{l_c}^k$ it goes from empty to the maximum queue length when $g_{i_{\overline{c}}}^k = \min\limits_{l_c \in \mathcal{G}_c} \frac{x_{i_c}^{max}}{\lambda_{i_c}} - \sigma_{i_c,i_{\overline{c}},i_c}$. ■

Lemma D.18 *Whenever $\exists i_c \in \mathcal{G}_c : \Delta x_{c,i_c}^{k+1} \geq 0 \wedge g_{i_c}^{\mu,k} = g_{i_c}^k \wedge g_{i_c}^k > g_{i_c}^{pbt}$ then it holds that:*

1 *All signals $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ that satisfy $x_{c,i_{\overline{c}}}^k < \Delta_{\overline{c}}(g_{i_c}^k)$ are empty at the end of the kth cycle c. It holds that $\Delta_{\overline{c}}(g_{i_c}^k) > 0$.*

2 *All queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ that are not empty at the end of the kth cycle c, have a queue length (at the end of the kth cycle c) that is at least $\Delta_{\overline{c}}(g^k_{i_c}) > 0$ lower than the queue length at the beginning of the kth cycle c.*

where $\Delta_{\overline{c}}(g^k_{i_c}) = (g^k_{i_c} - g^{pbt}_{i_c,i_{\overline{c}}})\frac{(1-\rho_{i_c}-\rho_{i_{\overline{c}}})\mu_{i_{\overline{c}}}}{1-\rho_{i_c}} > 0.$

Proof. Using $\Delta x^{k+1}_{c,i_c} \geq 0$, $g^{\mu,k}_{i_c} = g^k_{i_c}$ and (D.7a) until (D.7d) we can find that:

$$\Delta x^{k+1}_{c,i_{\overline{c}}} \leq -\Delta_{\overline{c}}(g^k_{i_c}) \text{ if } g^{\mu,m}_{i_{\overline{c}}} = g^{f(k)}_{i_{\overline{c}}}. \tag{D.27}$$

From this equation it follows that if $\exists i_c \in \mathcal{G}_c : \Delta x^{k+1}_{c,i_c} \geq 0 \wedge g^{\mu,k}_{i_c} = g^k_{i_c} \wedge g^k_{i_c} > g^{pbt}_{i_c}$ then signal $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ must have a slowmode, i.e. $g^{\mu,m}_{i_{\overline{c}}} < g^{f(k)}_{i_{\overline{c}}}$, whenever $x^k_{c,i_{\overline{c}}} < \Delta_{\overline{c}}(g^k_{i_c})$. This because otherwise it would result in an infeasible negative queue length. As a result queue $i_{\overline{c}}$ empties during $g^{f(k)}_{i_{\overline{c}}}$ whenever $x^k_{c,i_{\overline{c}}} < \Delta_{\overline{c}}(g^k_{i_c})$.

Further, note that whenever $x^{k+1}_{c,i_{\overline{c}}} > 0$, i.e. queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ is not emptied during $g^{f(k)}_{i_{\overline{c}}}$, then it holds that $g^{\mu,m}_{i_{\overline{c}}} = g^{f(k)}_{i_{\overline{c}}}$. Thus, whenever a queue is not empty at the end of the kth cycle c then the queue length is at least $\Delta_{\overline{c}}(g^k_{i_c}) > 0$ lower than at the beginning of the kth cycle c. ■

Lemma D.19 *Whenever $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \Delta x^{k+1}_{c,i_{\overline{c}}} \geq 0 \wedge g^{\mu,f(k)}_{i_{\overline{c}}} = g^{f(k)}_{i_{\overline{c}}} \wedge g^{f(k)}_{i_{\overline{c}}} > g^{pbt}_{i_{\overline{c}}}$ then it holds that:*

1 *All signals $i_c \in \mathcal{G}_c$ that satisfy $x^k_{c,i_c} < \lambda_{i_c}(g^{f(k)}_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c}) + \Delta_{\overline{c}}(g^{f(k)}_{i_c})$ are empty at the end of the kth cycle c. It holds that $\Delta_{\overline{c}}(g^{f(k)}_{i_c}) > 0$.*

2 *All queues $i_c \in \mathcal{G}_c$ that are not empty at the end of the kth cycle c, have a queue length (at the end of the kth cycle c) that is at least $\Delta_{\overline{c}}(g^{f(k)}_{i_c}) > 0$ lower than the queue length at the beginning of the kth cycle c.*

where $\Delta_{\overline{c}}(g^{f(k)}_{i_c}) = (g^{f(k)}_{i_{\overline{c}}} - g^{pbt}_{i_{\overline{c}},i_c})\frac{(1-\rho_{i_c}-\rho_{i_{\overline{c}}})\mu_{i_c}}{1-\rho_{i_{\overline{c}}}} > 0.$

Proof. First of all, note that when $\Delta x^{k+1}_{c,i_{\overline{c}}} > 0$, it holds that $g^{\mu,f(k)}_{i_{\overline{c}}} = g^{f(k)}_{i_{\overline{c}}}$, i.e. $g^{f(k)}_{i_{\overline{c}}}$ could not have a slowmode. This because queue $i_{\overline{c}}$ is not emptied during $g^{f(k)}_{i_{\overline{c}}}$.

Further, it must hold that $x^{k+1}_{c,i_c} \geq \lambda_{i_c}(g^{f(k)}_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c})$. This because, at the end of the kth cycle c, signal i_c is red for $(g^{f(k)}_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c})$.

Using (D.7a) until (D.7d), $\Delta x^{k+1}_{c,i_{\overline{c}}} \geq 0$ and $g^{\mu,f(k)}_{i_{\overline{c}}} = g^{f(k)}_{i_{\overline{c}}}$, we can find that:

$$\Delta x^{k+1}_{c,i_c} \leq -\Delta_c(g^{f(k)}_{i_{\overline{c}}}) \text{ if } g^{\mu,f(k)}_{i_c} = g^{f(k)}_{i_c}. \tag{D.28}$$

The minimum queue length during the kth cycle c (at the end of $g^k_{i_c}$)is equal to $x^k_{c,i_c} + \Delta x^{k+1}_{c,i_c} - \lambda_{i_c}(g^{f(k)}_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c})$. Hence, we can obtain from (D.28) that signal $i_c \in \mathcal{G}_c$ has a slowmode, i.e. $g^{\mu,k}_{i_c} < g^k_{i_c}$, if $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \Delta x^{k+1}_{c,i_{\overline{c}}} \geq 0 \wedge g^{\mu,f(k)}_{i_{\overline{c}}} = g^{f(k)}_{i_{\overline{c}}} \wedge g^{f(k)}_{i_{\overline{c}}} > g^{pbt}_{i_{\overline{c}}}$ and $x^k_{c,i_c} < \lambda_{i_c}(g^{f(k)}_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c}) + \Delta_c(g^{f(k)}_{i_{\overline{c}}})$. This because, if signal $i_c \in \mathcal{G}_c$ does not have a slowmode it would result in an infeasible negative queue length. This means that queue i_c empties $g^{f(k)}_{i_{\overline{c}}}$ when $x^k_{c,i_c} < \lambda_{i_c}(g^{f(k)}_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c}) + \Delta_c(g^{f(k)}_{i_{\overline{c}}})$.

Further, whenever $x^{k+1}_{c,i_c} > \lambda_{i_c}(g^{f(k)}_{i_{\overline{c}}} + \sigma_{i_c,i_{\overline{c}},i_c})$, i.e. the queue $i_c \in \mathcal{G}_c$ is not emptied during $g^k_{i_c}$, then it holds that $g^{\mu,k}_{i_c} = g^k_{i_c}$. Thus, whenever a queue is not empty at the end of the kth cycle c then the queue length is at least $\Delta_c(g^{f(k)}_{i_{\overline{c}}}) > 0$ lower than at the beginning of the kth cycle c. ■

Lemma D.20 *Whenever* $s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3b \to switch.3a \to switch.3b$ *then it will hold that* $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+2} \geq 0$.

Whenever $s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3b \to switch.3a \to switch.3b$ *then it holds that* $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : \Delta x_{c,i_{\overline{c}}}^{k+2} \geq 0 \wedge g_{i_{\overline{c}}}^{\mu,f(k)+1} = g_{i_{\overline{c}}}^{f(k)+1} \wedge g_{i_{\overline{c}}}^{f(k)+1} > g_{i_{\overline{c}}}^{pbt}$.

Proof. Lets consider the queue $i_{\overline{c}}^{f(k)+1}$ (the queue that has a maximum queue length at the start of its green time $g_{\overline{c}}^{f(k)+1}$).

Because $s_{\overline{c}}^{f(k)} = s_{\overline{c}}^{f(k)+1} = switch.3b$ it holds that $g_{i_{\overline{c}}^{f(k)+1}}^{f(k)} - g_{i_{\overline{c}}^{f(k)+1}}^{f(k)} = g_{i_{\overline{c}}^{f(k)+1}}^{f(k)+1} = \tilde{g}_{i_{\overline{c}}^{f(k)+1}}^{max}$. We are going to show that $\Delta x_{c,i_{\overline{c}}^{f(k)+1}}^{k+2} = x_{c,i_{\overline{c}}^{f(k)+1}}^{k+2} - x_{c,i_{\overline{c}}^{f(k)+1}}^{k+1} \geq 0$ and thus that it holds that $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+2} \geq 0$.

We know that queue $i_{\overline{c}}^{f(k)+1}$ was not emptied during $g_{i_{\overline{c}}^{f(k)+1}}^{f(k)}$ (because $s_c^{k+1} = switch.3a$) and thus it holds that $g_{i_{\overline{c}}^{f(k)+1}}^{\mu,f(k)} = g_{i_{\overline{c}}^{f(k)+1}}^{f(k)}$. It also holds that $g_{i_{\overline{c}}^{f(k)+1}}^{\mu,f(k)+1} = g_{i_{\overline{c}}^{f(k)+1}}^{f(k)+1}$ (because we could not empty queue $i_{\overline{c}}^{f(k)+1}$ during $g_{i_{\overline{c}}^{f(k)+1}}^{f(k)}$ and because queue $i_{\overline{c}}^{f(k)+1}$ is full at the start of $g_{i_{\overline{c}}^{f(k)+1}}^{f(k)+1}$). Further, we can see that $x_{c,i_{\overline{c}}^{f(k)+1}}^{k+1} \leq x_{i_{\overline{c}}^{f(k)+1}}^{max} - \tilde{g}_{i_{\overline{c}}^{f(k)+1}}^{max}(\mu_{i_{\overline{c}}^{f(k)+1}} - \lambda_{i_{\overline{c}}^{f(k)+1}}) + \sigma_{i_{\overline{c}}^{f(k)+1}, i_c^{g,f}} \lambda_{i_{\overline{c}}^{f(k)+1}}$ and that $x_{i_{\overline{c}}^{f(k)+1}}^{k+2} = x_{i_{\overline{c}}^{f(k)+1}}^{max} - \tilde{g}_{i_{\overline{c}}^{f(k)+1}}^{max}(\mu_{i_{\overline{c}}^{f(k)+1}} - \lambda_{i_{\overline{c}}^{f(k)+1}}) + \sigma_{i_{\overline{c}}^{f(k)+1}, i_c^{g,f}} \lambda_{i_{\overline{c}}^{f(k)+1}}$ (since queue $i_{\overline{c}}^{f(k)+1}$ is full at the start of $g_{i_{\overline{c}}^{f(k)+1}}^{f(k)+1}$). Hence, $\Delta x_{c,i_{\overline{c}}^{f(k)+1}}^{k+2} = x_{c,i_{\overline{c}}^{f(k)+1}}^{k+2} - x_{c,i_{\overline{c}}^{f(k)+1}}^{k+1} \geq 0$.

During this proof we have already shown that $\Delta x_{c,i_{\overline{c}}^{f(k)+1}}^{k+2} \geq 0 \wedge g_{i_{\overline{c}}^{f(k)+1}}^{\mu,f(k)+1} = g_{i_{\overline{c}}^{f(k)+1}}^{f(k)+1} \wedge g_{i_{\overline{c}}^{f(k)+1}}^{f(k)+1} > g_{i_{\overline{c}}^{f(k)+1}}^{pbt}$. ■

Lemma D.21 *Whenever* $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.2 \to switch.2 \to switch.2 \to switch.2$ *and* $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} \leq 0 \wedge \max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$ *then it holds that* $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{\overline{c},i_c}^{f(k)+1} \leq 0 \wedge \max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{\overline{c},i_{\overline{c}}}^{f(k)+1} \leq 0$.

Whenever $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3b \to switch.3b \to switch.3b \to switch3b$ *and* $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} = 0 \wedge \max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} = 0$ *then it holds that* $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{\overline{c},i_c}^{f(k)+1} = 0 \wedge \max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{\overline{c},i_{\overline{c}}}^{f(k)+1} = 0$.

Proof.
Note that in this case:

$$\begin{aligned}
g_{i_c}^k &= \tilde{g}_{i_c}^{max}, & \forall i_c \in \mathcal{G}_c,\\
g_{i_c}^{k+1} &= \tilde{g}_{i_c}^{max}, & \forall i_c \in \mathcal{G}_c,\\
g_{i_{\overline{c}}}^{f(k)} &= \tilde{g}_{i_{\overline{c}}}^{max}, & \forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}},\\
g_{i_{\overline{c}}}^{f(k)+1} &= \tilde{g}_{i_{\overline{c}}}^{max}, & \forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}.
\end{aligned}$$

First we prove that $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{\overline{c},i_c}^{f(k)+1} \leq 0$ if $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} \leq 0$ and that $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{\overline{c},i_c}^{f(k)+1} = 0$ if $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} = 0$ (when $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.2 \to switch.2 \to switch.2 \to switch.2$ or $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3b \to switch.3b \to switch.3b \to switch.3b$). Using (D.6) and (D.7) we can find the following expressions for $\Delta x_{c,i_c}^{k+1}$ and $\Delta x_{\overline{c},i_c}^{f(k)+1}$:

$$\Delta x_{c,i_c}^{k+1} = -g_{i_c}^{\mu,k}(\mu_{i_c} - \lambda_{i_c}) + \tilde{r}_{i_c}^{max}\lambda_{i_c}, \tag{D.29}$$

$$\Delta x_{\overline{c},i_c}^{f(k)+1} = -g_{i_c}^{\mu,k+1}(\mu_{i_c} - \lambda_{i_c}) + \tilde{r}_{i_c}^{max}\lambda_{i_c}. \tag{D.30}$$

Note that $\Delta x_{\overline{c},i_c}^{f(k)+1}$ is equal to the expression from $\Delta x_{c,i_c}^{k+2}$. In Lemma D.15 we have shown that for all signals $i_c \in \mathcal{G}_c$ it holds that:

1 $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+2} \leq 0$ if $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} \leq 0$

2 $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+2} = 0$ if $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} = 0$

Thus, it holds that $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{\overline{c},i_c}^{f(k)+1} \leq 0$ if $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} \leq 0$ and $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{\overline{c},i_c}^{f(k)+1} = 0$ if $\max\limits_{i_c \in \mathcal{G}_c} \Delta x_{c,i_c}^{k+1} = 0$

Now we prove that $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{\overline{c},i_{\overline{c}}}^{f(k)+1} \leq 0$ if $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$ and that $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{\overline{c},i_{\overline{c}}}^{f(k)+1} = 0$ if $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} = 0$ (when $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.2 \to switch.2 \to switch.2 \to switch.2$ or $s_c^k \to s_{\overline{c}}^{f(k)} \to s_c^{k+1} \to s_{\overline{c}}^{f(k)+1} = switch.3b \to switch.3b \to switch.3b \to switch.3b$). Using (D.6) , (D.7) we can find the following expressions for $\Delta x_{c,i_{\overline{c}}}^{k+1}$ and $\Delta x_{\overline{c},i_{\overline{c}}}^{f(k)+1}$:

$$\Delta x_{c,i_{\overline{c}}}^{k+1} = -g_{i_{\overline{c}}}^{\mu,f(k)}(\mu_{i_c} - \lambda_{i_c}) + \tilde{r}_{i_c}^{max}\lambda_{i_c}, \tag{D.31}$$

$$\Delta x_{\overline{c},i_{\overline{c}}}^{f(k)+1} = -g_{i_{\overline{c}}}^{\mu,f(k)}(\mu_{i_c} - \lambda_{i_c}) + \tilde{r}_{i_c}^{max}\lambda_{i_c}. \tag{D.32}$$

Note that $\Delta x_{c,i_{\overline{c}}}^{k+1} = \Delta x_{\overline{c},i_c}^{f(k)+1}$. As a result $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{\overline{c},i_{\overline{c}}}^{f(k)+1} \leq 0$ if $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} \leq 0$ and $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{\overline{c},i_{\overline{c}}}^{f(k)+1} = 0$ if $\max\limits_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \Delta x_{c,i_{\overline{c}}}^{k+1} = 0$. This concludes this proof. ■

Lemma D.22 *Whenever* $s_c^k \to s_{\overline{c}}^{f(k)} = switch.1a \to switch.1a$ *then* $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : g_{i_{\overline{c}}}^{\mu,f(k)} = g_{i_{\overline{c}}}^{f(k)}$ *and it holds that* $g_{i_{\overline{c}}}^{f(k)} > g_{i_{\overline{c}}}^{pbt}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$.

Proof. Whenever $s_{\overline{c}}^{f(k)} = switch.1a$ then it holds that $\exists i_c \in \mathcal{G}_c : x_{i_c}(t) > x_{i_c}^{\sharp}$, where $x_{i_c}(t)$ is the queue length of queue $i_c \in \mathcal{G}_c$ when signal $i_c^{r,f}$ switches to red (during $r_{i_c}^{k+1}$, $i_c \in \mathcal{G}_c$) (see Section D.1 for more information). The definition of $x_{i_c}^{\sharp}$ is shown in (8.2). Because all queues in the set $\mathcal{G}_c$ were empty at the start of $r_{i_c}^k$ (since $s_c^k = switch.1a$) this means that $\exists i_c \in \mathcal{G}_c : (r_{i_c}^{k+1} - \sigma_{i_{\overline{c}}^{r,f},i_c})\lambda_{i_c} > x_{i_c}^{\sharp}, \quad i_c \in \mathcal{G}_c$

Using (8.2) we can find that $r_{i_c}^{k+1} > r_{i_c}$ (and thus that $g_{i_{\overline{c}}}^{f(k)} > g_{i_{\overline{c}}}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$). Thus, it holds that $g_{i_{\overline{c}}}^{f(k)} > g_{i_{\overline{c}}}$. Using (7.14c), (7.14a) and (D.2) we can find that $g_{i_{\overline{c}}}^{f(k)} > g_{i_{\overline{c}}}^{pbt}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ and that $g_{i_{\overline{c}}}^{f(k)} > g_{i_{\overline{c}}}^{min}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$.

Note that (in general) when $s_{\overline{c}}^{f(k)} = switch.1a$ then either $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : g_{i_{\overline{c}}}^{f(k)} = g_{i_{\overline{c}}}^{min}$ (a signal is served for the minimum green time) or $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : g_{i_{\overline{c}}}^{\mu,f(k)} = g_{i_{\overline{c}}}^{f(k)}$ (there is a signal $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ that we switch to red exactly at the moment it is emptied). Because, when $s_c^k \to s_{\overline{c}}^{f(k)} = switch.1a \to switch.1a$ it holds that $g_{i_{\overline{c}}}^{f(k)} > g_{i_{2\overline{c}}}^{min}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ it must hold that $\exists i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : g_{i_{\overline{c}}}^{\mu,f(k)} = g_{i_{\overline{c}}}^{f(k)}$ (there is a signal $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ that we switch to red exactly at the moment that it is emptied). ■

Lemma D.23 *Whenever $s_c^k = switch.1b$ then we will follow the trajectory that we want to follow from the start of the $k+1$th cycle c. Whenever $s_c^k = switch.1b$ then $s_c^k = switch.1b$, $\forall h \geq k$ and $s_{\overline{c}}^{f(h)} = switch.1b$, $\forall h \geq k$.*

Proof. We switch because of the reason $s_c^k = switch.1b$ whenever $\tau^1_{i_c^{g,f}} \leq \tau^2_{i_c^{g,f}} \wedge \tau^1_{i_c^{g,f}} \leq \tau^3_{i_c^{g,f}}$ and $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}} : x_{i_{\overline{c}}}(t) \leq x^\sharp_{i_{\overline{c}}}$ (see Section 8.3).

We are going to show that whenever $s_c^k = switch.1b$ then it will hold that $g^k_{i_{\overline{c}}} = g_{i_{\overline{c}}}$, $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ and $s_{\overline{c}}^{f(k)} = switch.1b$.

Whenever $s_c^k = switch.1b$ we can find that:

$$\tau^{1.1}_{i_c^{g,f}} \leq \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}}\left(\frac{x^\sharp_{i_{\overline{c}}} + \sigma_{i_c^{r,f},i_{\overline{c}}}\lambda_{i_{\overline{c}}}}{\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}} + \sigma_{i_c^{r,f},i_{\overline{c}},i_c^{r,f}} - \sigma_{i_c^{r,f},i_{\overline{c}}^{r,f},i_c^{r,f}}\right) + \sigma^{res}_{i_{\overline{c}}^{r,f}} \leq g_{i_c^{r,f}} + \sigma^{res}_{i_{\overline{c}}^{r,f}}, \tag{D.33a}$$

$$\tau^{1.2}_{i_c^{g,f}} = \max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}}\left(g^{min}_{i_{\overline{c}}} + \sigma_{i_c^{r,f},i_{\overline{c}},i_c^{r,f}} - \sigma_{i_c^{r,f},i_{\overline{c}}^{r,f},i_c^{r,f}}\right) + \sigma^{res}_{i_{\overline{c}}^{r,f}} \leq g_{i_c^{r,f}} + \sigma^{res}_{i_{\overline{c}}^{r,f}}, \tag{D.33b}$$

$$\tau^{1.3}_{i_c^{g,f}} = \min_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}}\left(\frac{x^\sharp_{i_{\overline{c}}}}{\lambda_{i_{\overline{c}}}} - \sigma_{i_c^{r,f},i_{\overline{c}}^{g,f}}\right) = g_{i_c^{r,f}} + \sigma^{res}_{i_{\overline{c}}^{r,f}}, \tag{D.33c}$$

$$\tau^2_{i_c^{g,f}} = g^{max}_{i_{\overline{c}}^{r,f}} + \sigma^{res}_{i_{\overline{c}}^{r,f}} \geq g_{i_{\overline{c}}^{r,f}} + \sigma^{res}_{i_c^{r,f}}, \tag{D.33d}$$

$$\tau^3_{i_c^{g,f}} = \min_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}}\left(\frac{x^{max}_{i_{\overline{c}}}}{\lambda_{i_{\overline{c}}}} - \sigma_{i_c^{r,f},i_{\overline{c}}^{g,f}}\right) \geq g_{i_{\overline{c}}^{r,f}} + \sigma^{res}_{i_c^{r,f}}. \tag{D.33e}$$

We will explain these expressions one by one.

For the desired trajectory the queue length of queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ equals $x^\sharp_{i_{\overline{c}}} + \sigma_{i_c^{r,f},i_{\overline{c}}}\lambda_{i_{\overline{c}}}$ when the green time of signal $i_{\overline{c}}$ starts. For the desired trajectory the amount of traffic that arrives during a red period can depart during a green period. Hence, it holds that:

$$\frac{x^\sharp_{i_{\overline{c}}} + \sigma_{i_c^{r,f},i_{\overline{c}}}\lambda_{i_{\overline{c}}}}{\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}} \leq g_{i_{\overline{c}}}, \qquad \forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}.$$

Further, because the green times are related according to (D.2) we can find that for the desired trajectory it holds that:

$$\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}}\left(\frac{x^\sharp_{i_{\overline{c}}} + \sigma_{i_c^{r,f},i_{\overline{c}}}\lambda_{i_{\overline{c}}}}{\mu_{i_{\overline{c}}} - \lambda_{i_{\overline{c}}}} + \sigma_{i_c^{r,f},i_{\overline{c}},i_c^{r,f}} - \sigma_{i_c^{r,f},i_{\overline{c}}^{r,f},i_c^{r,f}}\right) \leq g_{i_{\overline{c}}^{r,f}}.$$

Further, to find the expression (D.33a) we have used the fact that signal $i_{\overline{c}}^{r,f}$ is switched to green $\sigma^{res}_{i_{\overline{c}}^{r,f}}$ seconds after signal $i_c^{g,f}$ is switched to green and the fact that each queues $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ satisfied that its queue length $x_{i_{\overline{c}}}(t)$ at the end of $g^k_{i_c^{r,f}}$ is smaller than (or equal to) $x^\sharp_{i_{\overline{c}}}$ (because $s_c^k = switch.1b$).

Because the green periods of the signals in the set $\mathcal{G}_c$ are related according to (D.2) and using (7.14c) we can find that:

$$\max_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}}\left(g^{min}_{i_{\overline{c}}} + \sigma_{i_c^{r,f},i_{\overline{c}},i_c^{r,f}} - \sigma_{i_c^{r,f},i_{\overline{c}}^{r,f},i_c^{r,f}}\right) \leq g_{i_{\overline{c}}^{r,f}}.$$

Further, to find the expression (D.33b) we have used the fact that signal $i_{\overline{c}}^{r,f}$ is switched to green $\sigma^{res}_{i_{\overline{c}}^{r,f}}$ seconds after signal $i_c^{g,f}$ is switched to green.

Queue $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ reaches a queue length of $x^{\sharp}_{i_{\overline{c}}}$ when this signal has been red for $\frac{x^{\sharp}_{i_{\overline{c}}}}{\lambda_{i_{\overline{c}}}}$. Signal $i^{g,f}_{c}$ switches to green $\sigma_{i^{r,f}_{c}, i^{g,f}_{\overline{c}}}$ seconds after signal $i^{g,f}_{c}$ switched red therefore we find that:

$$\tau^{1.3}_{i^{g,f}_{c}} = \min_{i_{\overline{c}} \in \mathcal{G}_{\overline{c}}} \left(\frac{x^{\sharp}_{i_{\overline{c}}}}{\lambda_{i_{\overline{c}}}} - \sigma_{i^{r,f}_{c}, i^{g,f}_{\overline{c}}}\right). \tag{D.34}$$

Using (8.2) and (D.2) we can find expression (D.33c).

We can find expression (D.33d) using the relation between maximum green times shown in (8.1f) and (8.1g) and using the fact that signal $i^{r,f}_{\overline{c}}$ is switched to green $\sigma^{res}_{i^{r,f}_{\overline{c}}}$ seconds after signal $i^{g,f}_{c}$ is switched to green.

Queue i_c reaches a queue length of $x^{max}_{i_{\overline{c}}}$ when this signal has been red for $\frac{x^{max}_{i_{\overline{c}}}}{\lambda_{i_{\overline{c}}}}$. Signal $i^{g,f}_{c}$ switches to green $\sigma_{i^{r,f}_{c}, i^{g,f}_{\overline{c}}}$ seconds after signal $i^{g,f}_{c}$ switched red. Using (7.14b) results in:

$$g_{i^{r,f}_{c}} \leq \min_{i_c \in \mathcal{G}_c} \left(\frac{x^{max}_{i_{\overline{c}}}}{\lambda_{i_{\overline{c}}}} - \sigma_{i^{r,f}_{c}, i^{g,f}_{\overline{c}}}\right).$$

From (D.33) we can see that $g^{f(k)}_{i^{r,f}_{\overline{c}}} = g_{i^{r,f}_{\overline{c}}}$. Because all green times in the set $\mathcal{G}_c$ are related according to (D.2) this means that $g^{f(k)}_{i_{\overline{c}}} = g_{i_{\overline{c}}}$, $\forall i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$.

From (D.33) we can easily see that:

$$\tau^{1}_{i^{g,f}_{c}} \leq \tau^{2}_{i^{g,f}_{c}} \wedge \tau^{1}_{i^{g,f}_{c}} \leq \tau^{3}_{i^{g,f}_{c}}, \forall i_c \in \mathcal{G}_c : x_{i_c}(t) \leq x^{\sharp}_{i_c}.$$

Thus, it holds that $s^{f(k)}_{\overline{c}} = switch.1b$.

Hence, when $switch.1b$ occurs from then on we always switch signals to red because of the reasons $switch.1b$ and from then on the green time of every signal is equal to the green time of that signal for the trajectory that we want to follow.

We can easily see that when $s^{k}_{c} = switch.1b$ occurs then we follow the desired trajectory from the start of the $k+1$th cycle c because it holds for all signals that $i_{\overline{c}} \in \mathcal{G}_{\overline{c}}$ that the queue length at the end of $g^{f(k)}_{i_{\overline{c}}}$ are equal to zero (just like for the desired trajectory) and $\forall i_c \in \mathcal{G}_c$ it holds that the queue length at the end of $g^{f(k)}_{i^{r,f}_{\overline{c}}}$ is equal to $x^{\sharp}_{i_c}$ (just like for the desired trajectory). ∎

Bibliography

[1] A.S. Alfa and M.F. Neuts (1995). Modelling vehicular traffic using the discrete time Markovian arrival process, *Transportation Science.* 29, 109-117.

[2] M.A.A. Boon (2011). Polling models, From theory to traffic intersections. *PhD. thesis, Technical University Eindhoven.*

[3] O.J. Boxma and W.P. Groenendijk (1987). Pseudo-conservation laws in cyclic service systems. *Journal of Applied Probability.* 24, 949-964.

[4] E. Brockfeld, R. Barlovic, A. Schadschneider and M. Schreckenberg (2001). Optimizing traffic lights in a cellular automaton model for city traffic, *Phys. Rev. E.* 64, 056132.

[5] M.S. van den Broek (2004). Traffic Signals - Optimizing and analyzing traffic control systems, *Master's thesis thesis, Technical University Eindhoven.*

[6] M.S. van den Broek, J.S.H. van Leeuwaarden and I.J.B.F. Adan (2006). Bounds and approximations for the fixed -cycle traffic-light queue, *Transportation Science.* 40 (4), 484-496.

[7] C. Daganzo (1990). Some properties os polling systems. *Queueing systems.* 6, 137-154.

[8] J.N. Darroch (1964). On the traffic-light queue, *Ann. Math. Statist.* 35, 380-388.

[9] J.N. Darroch, G.E. Newell and R.W.J. Morris (1964). Queues for a vehicle actuated traffic light. *Operations Research.* 12, 882-895.

[10] J.A.W.M. van Eekelen (2008). Modelling and control of discrete event manufacturing flow lines. *PhD. thesis, Technical University Eindhoven.*

[11] M. Fouladvand and M. Nematollahi (2001). Optimization of green-times at an isolated urban crossroads, *Eur. Phys. J. B.* 22. 395-401.

[12] R. Haijema and J. van der Wal (2007). An MDP decomposition approach for traffic control at isolated signalized intersections. *Probability in the Engineering and Informational Sciences.* 22, 587-602.

[13] D. Heidemann (1994). Queue length and delay distributions at traffic signals, *Transportation Research Part B.* 28, 377-389.

[14] TRB (2000). Highway Capacity Manual 2000, *Transportation Research Board.*

[15] S. Lämmer, D. Helbing (2008). Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment.* P04019.

[16] J.S.H. van Leeuwaarden (2006). Delay analysis for the fixed-cycle traffic-light queue, *Transportation science.* 40 (2), 189-199.

[17] V. Feoktistova, A. Matveev, E. Lefeber and J.E. Rooda (2011), On optimal switching interactive decentralized control of networked manufacturing systems, *Proceedings of the 18th IFAC World Congress, Milan, Italy,* 14048-14054.

[18] D.R. McNeill (1968). A solution to the fixed-cycle traffic-light problem for compound Poisson arrivals, *J. Appl. Probab.* 5 625-235.

[19] A.J. Miller (1963). Settings for fixed-cycle traffic signals, *Oper. Res. Quart.* 14 273-386.

[20] G.F. Newell (1965). Queues for a fixed-cycle traffic light, *Ann. Math. Statist.* 31 589-597.

[21] G.F. Newell (1969). Properties of vehicle-actuated signals: I. one-way streets. *Transportation Science.* 3, 31-52.

[22] G.F. Newell and E.E. Osuna (1969). Properties of vehicle-actuated signals: II. two-way streets. *Transportation Science.* 3, 99-125.

[23] K. Ohno (1978). Computational algorithm for a fixed-cycle traffic-light queue, *Transportation Science.* 12, 29-47.

[24] J.W. Polderman and J.C. Willems (1998). Introduction to Mathematical Systems Theory: A Behavioral Approach, *Springer.*

[25] T. Riedel and U. Brunner (1992). A control algorithm for traffic lights, *Automic Control Laboratory, Swiss Federal Institute of Technology, Switzerland.*

[26] F. Viti and H.J van Zuylen (2010). Probabilistic models for qeueus at fixed control signals, *Transportation Research Part B.* 44, 120-135.

[27] X. Wang and K. Yin. (2010) Vehicle actuated signal performance under general traffic at an isolated intersection, *Department of Civil Engineering, Texas A&M University.*

[28] F.V. Webster (1958). Traffic signal settings, *Road Res. Lab. Tech. Rep.* 39.

[29] F.V. Webster and B.M. Cobe (1966). Traffic signals, *Road Res. Lab. Tech. Rep.* 56.